



# Sar-At 12

## *Creare un sito dinamico*

Versione di settembre 2017  
Ultima revisione settembre 2021

# Sommario

<b>Sommario .....</b>	<b>2</b>
<b>Parte prima: Le sezioni .....</b>	<b>10</b>
La struttura della sezione.....	10
Nomi di campo	10
La pagina mastro .....	11
Caricamento della pagina mastro sul server .....	13
Creazione di una nuova sezione.....	13
Nome e tipo della sezione	13
Contenuti	13
Opzioni	14
Risposte	14
Pagina mastro	14
Inserimento dei dati (pagine) nella sezione .....	15
Editing al vivo	15
Pagine mastro Twig.....	18
Filtri	18
Abilitazione	19
Per saperne di più	19
Creazione di molte sezioni, layout complessivo del sito e moduli pronti	19
L'indirizzo di una pagina nel sito.....	20
id e masterId	21
Indirizzi pseudo-statici e SEO	22
Le sezioni menu .....	22
Nome menu	23
Opzioni menu	23
Richiamare una sezione menu	24
Tipo personalizzato menu	24
Pulsanti radio	25
Caselle di controllo (checkbox)	26
Raggruppamenti	26
Multiparole	27
Portale	28
Alterazione di una sezione esistente.....	28
I campi predefiniti di Sar-At.....	29
head della pagina mastro	29
Campi unici.....	29
Una grande sezione o tante piccole sezioni? .....	30
Le pagine mastro alternative .....	30
Pagine home di sezione	31

Nome amichevole delle pagine home	32
Pagine mastro localizzate	32
Pagine mastro per dispositivi mobili	33
INCLUDE e pagine mastro localizzate	33
IF e pagine mastro	33
Velocità di produzione delle pagine .....	33
La cache	33
Twig e la cache	34
Cache delle pagine mastro classiche	34
Tipo delle pagine mastro .....	34
Newsletter.....	35
Le pagine mastro e le newsletter	35
Mailmerge	35
Push	36
Informazioni sulla sezione.....	37
Nome sezione	37
Tipo sezione	37
Descrizione della sezione	37
Visibile nel menu Pagine	37
Dimensione su disco	38
Sezioni: risposte a domande frequenti.....	38
La pagina home usa una pagina mastro variante	38
Nomi delle colonne	38
Cloni	39
That's all folks .....	39
<b>Parte seconda: le viste.....</b>	<b>40</b>
Modalità semplificata .....	40
Selezione e filtro	40
Ordinamento nelle viste	41
Alcuni semplici esempi di Viste	41
Per creare un elenco alfabetico di tutti i prodotti	41
Per creare un elenco di tutti i prodotti per data di presentazione	41
Per creare un elenco dei cinque prodotti più nuovi	42
Viste complete .....	42
Condizioni AND	42
Condizioni OR	43
Priorità degli operatori	43
Viste per esperti.....	43
Viste per esperti e variabili predefinite	44
Variabili provenienti dal record corrente	44
Variabili provenienti dalla URL	45
Viste per esperti, pagine mastro classiche e cache	45

Il codice HTML nella pagina mastro.....	46
Viste nelle pagine mastro classiche	46
Elenco semplice	46
Link alle pagine interne	46
Elenco numerato programmaticamente	47
In tabella per righe a colore alternato	47
Viste nelle pagine mastro Twig	48
Elenco semplice	48
Link alle pagine interne	48
Elenco numerato programmaticamente	49
In tabella per righe a colore alternato	49
Viste navigabili .....	49
Strumenti di navigazione nelle Viste	50
Paginazione	51
Personalizzazione grafica dei link di navigazione	52
Il motore di ricerca .....	53
Il form di ricerca	53
Pagina “risultati della ricerca”	54
Quando nulla viene trovato	54
Motore di ricerca: altri esempi.....	55
Viste di tabelle correlate .....	56
Viste e portali	57
Uso avanzato dei portali	59
Ricerca a tutto testo .....	59
Viste: risposte a domande frequenti .....	61
La Vista non viene risolta nella mastro classica	61
Viste multiple	62
Viste su più sezioni	62
Vista di viste	62
Cancellare una Vista	63
<b>Parte terza: i form .....</b>	<b>64</b>
Introduzione: cosa sono i moduli .....	64
La pagina mastro per il modulo.....	65
Notifica della compilazione e pagina di ringraziamento	66
Errori di compilazione .....	67
Modulo a tappe.....	68
Autoregistrazione e moduli .....	69
Registrazione e social network .....	71
Configurazione per social network	72
Gruppi di utenti e social network	73
Presentazione al visitatore dell’opzione social	73
Anagrafica e social network	74

Modulo aggiornabile	75
Registrazione alla newsletter	75
Gestione degli errori .....	76
Radiobutton	77
Onerr	77
Selectedif	78
Checkbox	79
Upload .....	80
Nelle sezioni modulo ordinarie	80
Nelle sezioni modulo aggiornabili	80
Consultare il materiale dei visitatori	80
Mantenere riservato il materiale dei visitatori	80
Un esempio: il forum .....	81
Il blog	81
Moduli: risposte a domande frequenti.....	82
Campi nascosti	82
Campi incorporati di Sar-At e valori di default	82
Notifiche multiple	83
Trasformazione	83
<b>Parte quarta: gestione eventi.....</b>	<b>84</b>
Creazione di un evento .....	84
Tipi di evento .....	85
Reazioni possibili a un evento .....	86
Log degli eventi	87
Gruppi creati automaticamente .....	87
<b>Parte quinta: plug-in per mastro classiche .....</b>	<b>89</b>
Introduzione ai plug-in .....	89
I plug-in per pagine mastro classiche .....	89
Plug-in intestazione.....	89
Plug-in visitatore registrato.....	90
Plug-in motore di ricerca incorporato .....	90
Plug-in put_flash	96
Plug-in rectangular_view	96
Plug-in hit_counter	96
Plug-in del carrello della spesa	96
Plug-in sareIURL	97
Plug-in template	97
Plug-in motore di ricerca rapido	97
Plug-in sarat_stopgap	98
Plug-in clouder	99
Plug-in network_authentication	99
Plug-in fusi orari	100

Plug-in: risposte a domande frequenti .....	101
<b>Parte sesta: altri plug-in .....</b>	<b>102</b>
Il manager di plug-in .....	102
Plug-in per showPage .....	102
Plug-in per Twig .....	102
Tag definiti da Sar-At in Twig	103
Parametri URL in Twig	103
Plug-in per editAdd .....	103
Plug-in per validate .....	103
PDF	104
Image	104
no_http	104
Futura	104
Passata	104
Email	104
CAP	104
Codice fiscale	104
Partita IVA	104
IBAN	104
IP	104
URL	104
Password	105
Comuni italiani	105
Nations	105
Portion	105
I plug-in di inserimento .....	105
Plug-in file	106
Plug-in data	106
Plug-in ora	106
Plug-in data e ora	106
Plug-in radiobutton	106
Plug-in multiparola	106
Plug-in checkbox	106
Plug-in colore	107
Sviluppo di plug-in per showPage .....	107
Accesso ai dati dell'utente corrente	107
<b>Parte settima: funzioni avanzate .....</b>	<b>109</b>
Abilitazione dell'integrazione con Google Analytics .....	109
Google Ecommerce	111
showSitemap e il Sitemap protocol .....	112
Indirizzi pseudo-statici .....	112
Traduzione meccanica	112
URL amichevoli	112

Inclusione di frammenti html .....	112
I CSS .....	113
XML.....	114
showFile: il motore Sar-At per la distribuzione di documenti riservati ..	114
showRSS: creazione di flussi in syndication.....	115
I podcast	116
showErr e la gestione degli errori.....	117
showErr per gli amministratori	117
showErr per gli utenti	117
Codici di errore	118
Personalizzazione della pagina di errore	119
Gli alias.....	119
Attenzione ai 404	120
Verifica e riparazione alias	120
Personalizzazione di menu all'interno di Sar-At e integrazione nuovi script	120
Richiamo di funzioni interne di Sar-At a partire da script esterni	122
Messa in sicurezza degli script richiamati da Sar-At	122
Sicurezza.....	123
showCal e la generazione di appuntamenti scaricabili.....	123
Sarattizzatore .....	123
<b>Parte ottava: e-commerce .....</b>	<b>126</b>
Configurazione del commercio elettronico .....	127
Configurazione di Banca Sella	127
Configurazione di PayPal	128
Configurazione di HiPay	129
Metodi del commercio elettronico .....	129
addToCart	130
addToCartMultiple	131
addToCartMultipleByQuantity	131
changeCartQty	132
emptyCart	132
forgetBuyer	133
loginBuyer	133
removeFromCart	134
Diagnostica.....	134
<b>Parte nona: interfaccia RESTful .....</b>	<b>135</b>
Accesso alle risorse del database .....	135
Lettura di un record	135
Scrittura di un record	137
Ricerca nel database	138
Codici di errore .....	139
Possibili codici di errore	140

Validazione 140

**Appendice A. Riferimento .....141**

Campi del record corrente .....141

    Pseudo-campi predefiniti 141

INCLUDE.....141

BEGIN..END .....141

NAVIGATE...END.....141

IF per i campi della pagina.....141

    IF campo = valore 142

IF fuori dalle viste .....142

IF dentro alle viste .....143

LOCATION.....143

Costrutti condizionali sui moduli .....144

    ONERR..ENDERR 144

    CHECKBOX 144

        RADIOBUTTON 144

        SELECTEDIF 144

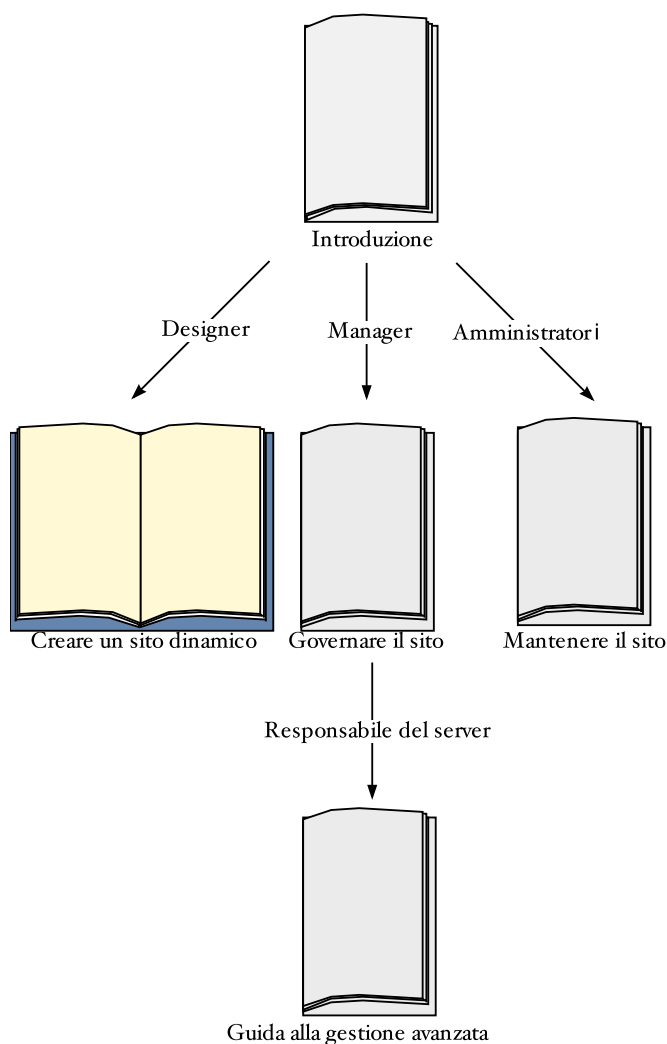
PHP .....144

Variabili predefinite per le Viste .....145



Scopo di questo manuale è spiegare con esempi concreti ai web designer come si realizza un sito dinamico usando Sar-At come sistema di gestione dei contenuti (CMS: *content management system*) e come si sviluppa un sito programmando in PHP estensioni che fanno affidamento sulla piattaforma Sar-At (*content management framework*).

Per la comprensione della componente CMS è richiesta la capacità di leggere il codice HTML, il CSS e il JavaScript. Per la sola componente framework, inoltre è richiesto il linguaggio PHP. Si suppone che il lettore abbia già consultato l'*Introduzione a Sar-At*. Lo schema sottostante presenta le dipendenze tra tutti i manuali di Sar-At.



Il libro aperto mostra il manuale che state leggendo. Le frecce mostrano le dipendenze (la lettura dei libri più in alto è essenziale per la comprensione dei testi più avanzati in basso).

Insieme a questo manuale sono forniti diversi materiali in formato digitale: pagine mastro HTML d'esempio, materiale pronto per l'importazione in Sar-At, eccetera.

Di tanto in tanto il manuale vi farà qualche suggerimento o fornirà qualche trucco d'uso. Li trovate evidenziati in giallo, come è il caso di questo stesso paragrafo.

# Parte prima: Le sezioni

Un sistema di gestione dinamico dei contenuti è utilissimo per gestire siti con moltissime pagine tutte in qualche modo simili. Immaginate di voler creare un sito web dedicato a tutti i cantanti al mondo: sarebbe impossibile per qualsiasi grafico scrivere una distinta pagina HTML per ciascuno — e mantenere aggiornato un sistema del genere sarebbe un incubo.

Usando Sar-At noi creiamo un'unica pagina HTML — una *pagina mastro* — e con quella vestiamo tutte le informazioni sui cantanti.

## La struttura della sezione

Immaginatevi le informazioni che volete mettere dentro al vostro sito. Per esempio, immaginiamo che, per cominciare, voi vogliate creare un sito che presenta i cantanti lirici. Di ogni cantante decidiamo di voler presentare: Nome e cognome; Data di nascita; Nazionalità; Timbro di voce; Curriculum vitae e una Fotografia. Ciascuna di queste informazioni viene chiamata “un campo”.

Sar-At vuole che ad ogni campo (informazione) diverso sia associato un tipo. Questo è indispensabile perché dati differenti vanno trattati in modo differente.

Facciamo un esempio pratico per capire questa esigenza. Immaginate di voler presentare i cantanti lirici per data di nascita, dal più vecchio al più giovane. I cantanti sono:  
Aristide Pallazzi, nato il 25/11/1980  
Mirella Di Panna, nata il 15/07/1984  
Se Sar-At non sapesse che le date sono, appunto, date, collocherebbe prima Di Panna e poi Pallazzi. Questo accade perché in ordine alfabetico "15/07/1984" viene prima di "25/11/1980" (infatti 1 viene prima di 2).

I tipi di campo predefiniti da Sar-At sono:

Numero intero (per esempio 41)

Numero reale (con la virgola: per esempio 3,14)

Data (per esempio primo gennaio 2020)

Ora (per esempio 07:30)

Data e ora (per esempio primo gennaio 2020 alle 7:30)

Colore

Testo breve (sino a 255 caratteri, per esempio un cognome)

Testo lungo (sino a 65.536 caratteri, per esempio una scheda prodotto)

File (un documento esterno, come per esempio una lettera creata con Microsoft Word o una fotografia)

Menu, multiparole, checkbox e radiobutton (una serie di alternative, le cui differenze vedremo in seguito).

Portale (un collegamento a una fonte di informazioni esterna. Per esempio, tutti i modelli di auto offerti dalla marca di cui si sta parlando).

## Nomi di campo

Sar-At vuole anche che ogni campo abbia un nome distinto. I nomi possono utilizzare lettere, cifre, ma anche lettere accentate, segni di interpunzione e alfabeti differenti dal latino (per esempio “città\_di\_nascita” è un nome di campo valido, e lo sono anche “io&tu'ho\_capito” o “привет”). Meglio limitare la fantasia, però: non tutti i nomi usabili dentro Sar-At possono venire riconosciuti e usati nelle pagine mastro, quindi serve un po' di attenzione o finiremo per memorizzare informazioni che non possiamo poi visualizzare sul sito. Il suggerimento è di utilizzare solo lettere non accentate sia minuscole che maiuscole, cifre e il trattino basso \_

Se assegnate a una sezione o ad un suo campo un nome che risulterebbe inaccettabile in una pagina mastro, Sar-At ve lo fa notare immediatamente.

Nel caso dei cantanti d'opera decideremo di lavorare così:

Campo	Tipo	Scopo
nome	Testo breve	Nome di battesimo
cognome	Testo breve	
date	Data	Data di nascita
nazione	Testo breve	Nazionalità di provenienza
timbro	Testo breve	Baritono o tenore o soprano o...
bio	Testo lungo	Curriculum vitae
foto	File	Fotografia

Armati di queste decisioni possiamo disegnare la nostra prima pagina mastro.

Se non avete abilità grafiche, e se lascerete che sia un grafico a progettare le pagine per voi, tenete presente che Sar-At può generare per voi una “pagina mastro sintetica”, cioè un HTML essenziale e privo di immagini ma pienamente utilizzabile.

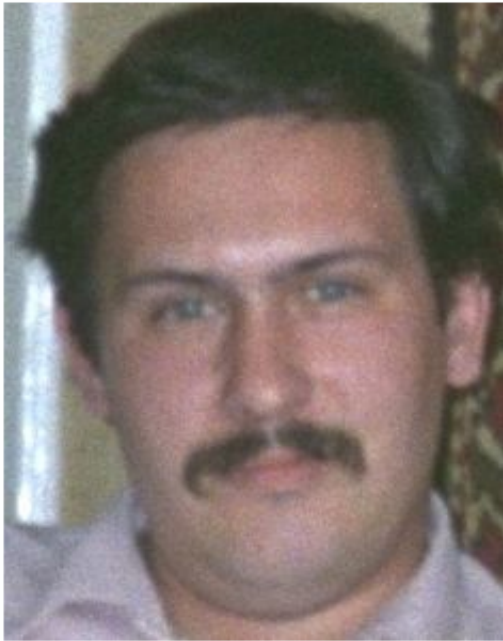
### La pagina mastro

Immaginatevi le pagine del vostro sito. Disegnatele con Dreamweaver, o lo strumento che più vi piace. Per ora metteteci dentro informazioni false. Per esempio, se stavate pensando a una pagina dedicata al grande tenore Aristide Pallazzi, disegnatela completamente e metteteci dentro la sua foto, la sua biografia, una tabella con i suoi dischi. Usate HTML puro.

Sentitevi liberi di progettare anche la struttura di navigazione del sito, anche se non è indispensabile a questo punto. Per esempio: voi avete in mente un motore di ricerca che elenca i timbri di voce, e quando ho scelto tenore mostra tutti i tenori, e cliccando il nome di Aristide Pallazzi va alla sua scheda. O magari un più tradizionale motore di ricerca dove scrivo il nome dell'artista e il sistema ne visualizza i dati.

Potrebbe venire fuori qualche cosa di simile all'immagine.

## Aristide Pallazzi



**Data di nascita:** 1/09/1980

**Nazionalità:** italiana

**Timbro:** tenore

## Carriera

Il grande Pallazzi interpreta il ruolo del Barbiere di Siviglia per la prima volta durante una recita alla scuola materna nel 1986.

Ora vogliamo generalizzare e trasformare questo *facsimile* in una vera pagina mastro. È facilissimo: ci basta entrare sulla pagina HTML e sostituire il testo d'esempio con i segnaposto.

Ci sono due modi in Sar-At versione 11.0 (e successive) di redigere le pagine mastro e fissare i segnaposto. Inizialmente vi presenteremo la versione classica, che è più semplice da comprendere e mettere all'opera, e più veloce in esecuzione. Immediatamente dopo presenteremo la versione *Twig*, ben più flessibile e potente.

Con la modalità classica delle pagine mastro in Sar-At, si tratta semplicemente di utilizzare il nome breve dell'informazione (quello elencato nella colonna di sinistra della tabella poco sopra) circondato da parentesi graffe.

**Data di nascita:** {data}

**Nazionalità:** {nazionalita}

**Timbro:** {timbro}

Per esempio, dove sta scritto "Aristide Palazzi" noi scriveremo {nome} {cognome}. Al posto di "tenore", sempre consultando la tabella che abbiamo creato al paragrafo precedente, scriveremo {timbro}. Attenzione a maiuscole e minuscole, che per Sar-At sono cose distinte: per semplicità consigliamo di usare ovunque solo le minuscole. Non bisogna neppure lasciare spazi bianchi dentro alle parentesi graffe.

Il risultato finale, naturalmente, avrà un aspetto "sbagliato" dentro Dreamweaver o semplicemente dentro al browser: assomiglierà a quello visto qui a sinistra. Non c'è di che preoccuparsi: la pagina visibile ai visitatori verrà composta dal motore di Sar-At combinando questa pagina mastro con i dati che tra poco inseriremo nel programma.

Probabilmente, arrivati a questo punto, avete qualche dubbio riguardo alla foto. Per adesso non preoccupatevi: ci penseremo tra qualche minuto.

### Caricamento della pagina mastro sul server

Una volta completata la pagina mastro, essa va caricata nel sito. Utilizzate un programma SFTP, come per esempio *WinSCP* per Windows oppure *Cyberduck* per Mac OS. Se create le pagine HTML usando un programma come Dreamweaver potete utilizzare il sistema incorporato di questi software autore.

Collegatevi al seguente indirizzo:

`sftp://nome del vostro sito`

Vi verranno richiesti nome e parola d'ordine: utilizzate quelli che il vostro amministratore di rete ha fornito a questo scopo, oppure richiedetene di nuovi.

Se la connessione non vi riesce, controllate di aver trascritto nome utente e password esattamente. Ricordate che le lettere minuscole e le maiuscole non sono interscambiabili!

Una volta collegati, vedrete la cartella che ospiterà le pagine mastro del vostro sito. Depositare una copia dei file della pagina mastro (cantanti.html, css e cartella immagini).

### Creazione di una nuova sezione

Siamo finalmente pronti per giocare con Sar-At. Colleghiamoci all'ambiente di lavoro per autori, il cosiddetto *backoffice* (il suo indirizzo è `http://nome del vostro sito/sarat/`) e digitiamo il nostro nome e la nostra password.

Vorremo creare una nuova sezione. Nel linguaggio di Sar-At, una *sezione* è lo spazio in cui inseriamo un gruppo di dati omogenei (per esempio, tutte le schede di tutti i cantanti) in modo che Sar-At realizzi una pagina web da ciascuno di essi.

Per creare una nuova sezione dovrete collegarvi al server Sar-At usando la password del proprietario del sito o la password di root. Le password dei comuni amministratori non sono sufficienti.

Nel menu "Pagine" selezioniamo l'ultima voce, "Aggiungi sezione". Nella schermata che appare possiamo definire la sezione che desideriamo. La creazione della sezione, come altre operazioni un po' complesse in Sar-At, avviene attraverso un wizard, ovvero un processo guidato e composto da più schermate successive. Nel caso della creazione della sezione, i passi sono cinque: scegliere nome e tipo della sezione, definire i contenuti, stabilire come il sistema risponde se vengono fatte aggiunte alla sezione e assegnare una pagina mastro per la visualizzazione sul sito pubblico. Lasciatevi guidare dall'aiuto a linea e vi ci troverete benissimo.

### Nome e tipo della sezione

Per il nostro esempio a proposito dell'opera lirica, come nome alla nostra sezione "cantanti" va benissimo. Si tratterà di una sezione ordinaria, dove il proprietario del sito e i suoi collaboratori e redattori inseriranno contenuti fruiti dai visitatori via web.

### Contenuti

Nella seconda schermata (e nelle successive) Sar-At ci chiede di inserire, una alla volta, una descrizione di tutti i campi (i tipi di informazione che verranno memorizzati nella sezione, cioè

Inserimento di ulteriore campo ?	
Nome campo	nome
Note	Nome di battesimo del cantante
Tipo	Breve testo

i dati dei cantanti, come per esempio le loro date di nascita). In pratica, dobbiamo solo completare il riquadro riprodotto qui di fianco, una volta per ogni riga della tabella di pagina 6.

La riga indicata "Note" è riservata a un vostro promemoria. Può anche venire lasciata bianca. Il testo scritto in questo spazio viene mostrato, a mo' di commento, quando un redattore deve compilare una nuova pagina di questa sezione.

Quando arriviamo all'ultima specifica, quella della fotografia, Sar-At vorrà anche sapere quale posizione all'interno del server vada riservato alle immagini dei cantanti. Scegliete dal menu la voce *Crea e usa una nuova cartella* e date un nome a piacere ad essa.

Dobbiamo necessariamente usare questo metodo per creare la cartella sul server che ospiterà le foto man mano che esse verranno caricate dai redattori usando Sar-At stesso. Se voi invece usaste il programma SFTP per creare quella cartella, allora la cartella apparterrebbe a voi e non a Sar-At, il nostro programma non potrebbe aggiungervi contenuti e di conseguenza la cartella non sarebbe utilizzabile per gli upload dei redattori. Nella schermata della Struttura, Sar-At mostra in grigio le caselle che non gli appartengono e non le lascia selezionare.

Sempre per quanto riguarda la foto. nello spazio "max" potete inserire una dimensione in pixel prevista per l'immagine (per esempio: 320x240). Se venisse caricata una fotografia più grande, Sar-At la ridimensionerebbe automaticamente.

### Opzioni

Qui abbiamo una serie di scelte, per la maggior parte ben comprensibili. Il suggerimento per ora è di lasciare tutte le voci al loro valore iniziale e proseguire.

### Risposte


Questa schermata si attiva solo per le sezioni di tipo modulo, che vedremo in un capitolo più avanti. Quindi viene saltata.

### Pagina mastro

Nel menu dedicato troveremo soltanto la pagina che abbiamo appena caricato, "cantanti.html".

Se il menu "Pagina mastro" non contiene la voce "cantanti.html" verificate di aver correttamente caricato i file , come spiegato nel paragrafo precedente. Potete comunque procedere permettendo a Sar-At di creare una semplice pagina mastro per voi, la "pagina mastro sintetica".

Vedremo una schermata di conferma finale (oppure, se abbiamo commesso qualche errore, un messaggio esplicativo). In questa fase, infatti, Sar-At legge la pagina HTML che abbiamo caricato e la confronta con la struttura che abbiamo definito. Se le due non sono compatibili, Sar-At ci spiega dove sta il problema e ci offre qualche suggerimento che ci aiuta a risolvere il problema.

Se qualcosa non va, nel 99% dei casi significa che abbiamo dato un nome leggermente differente a un campo nella pagina mastro rispetto a quanto abbiamo fatto nella dichiarazione di nuova sezione. Per esempio, se la pagina mastro contiene il segnaposto {titolo\_studio} e nella sezione abbiamo inserito "titoloStudio" oppure Titolo\_studio, allora Sar-At si lamenta. Per rimediare possiamo correggere la pagina mastro, e poi ricaricarla sul server, oppure specularmente possiamo cancellare il componente incongruente nella schermata Contenuti, cliccando l'icona  al suo fianco, e poi reinserirlo usando la stessa nomenclatura della pagina mastro.

Complimenti! Avete appena creato il vostro primo sito dinamico! Tra un attimo vedremo di inserirci qualche contenuto e di navigarlo. Per ora premete il pulsante *Popola la nuova sezione*

## Inserimento dei dati (pagine) nella sezione

Il manuale *Mantenere il sito* copre i dettagli dell'inserimento di nuove pagine. In sintesi, comunque, ecco una spiegazione.

Abbiamo creato un sito dinamico, pronto per raccogliere e pubblicare le schede personali di migliaia di cantanti. Per ora, tuttavia, non abbiamo inserito nessun nome al suo interno. Sar-At, amichevolmente, ci lascia davanti a una schermata che ci invita a riempire di contenuti il sito (avremmo potuto raggiungerla noi stessi cliccando il menu *Pagine* in alto, ma Sar-At lo ha fatto per noi).

Clicchiamo il pulsante *Aggiungi pagina*: ci troviamo davanti a una scheda che possiamo compilare indicando i dati di un cantante (uno fittizio andrà benissimo).

<b>Accesso dei visitatori</b>	italiano <input type="checkbox"/> <b>Visibile</b> Non riservata, leggibile da chiunque <input type="checkbox"/> <input type="checkbox"/> <b>Conta le visite (ma rallenta il server)</b> ?
<b>title</b> Nome e cognome cantante	Aristide Pallazzi ?
<b>date</b> Data di nascita	1980-09-01 <input type="text"/>
<b>nazionalita</b> Nazione di nascita	italiana
<b>timbro</b> Basso, baritono o tenore	tenore ?
	<div style="border: 1px solid #ccc; padding: 5px;"> <p style="text-align: center;">Il grande Pallazzi interpreta il ruolo del <i>Barbiere di Siviglia</i> per la prima volta durante una recita</p> </div>

Inserite il nome e cognome del cantante nello spazio "title", ignorate i campi successivi (se ne parla in *Mantenere il sito*) e compilate tutte le righe da "nome" in giù tranne (per ora) "foto", poi cliccate il pulsante "Invia modifiche".

Sar-At conferma l'inserimento e vi chiede se volete creare ulteriori pagine o se preferite provare la nuova pagina. Sentitevi liberi di impraticarvi e giocare con il sistema inserendo alcune schede su cantanti veri o immaginari. Poi premete il pulsante "Editing al vivo".

### Editing al vivo

Dovreste vedere una schermata simile a quella qui sotto.

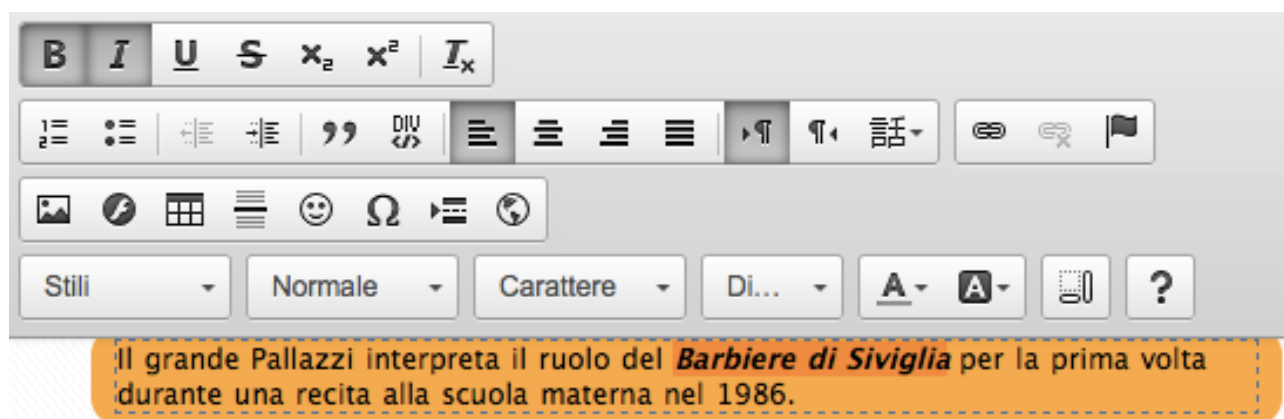
The screenshot shows the Sar-At administration interface. At the top left is the Sar-At logo with the version number 11.0.062. To the right, it says "Amministratore root" with a question mark icon and a "Logout" button. Below this is a navigation bar with links for "Sar-At", "Server", "Sito", "Pagine", "Utilizzatori", and "E-Commerce". The main content area displays the profile for "Aristide Pallazzi". On the left is a photo of a man with a mustache. To the right, in a green box, are the following details: "Data di nascita: 09/01/1980", "Nazionalità: italiana", and "Timbro: tenore" with a pencil icon. Below the photo is a section titled "Carriera" with a text box containing: "Il grande Pallazzi interpreta il ruolo del *Barbiere di Siviglia* per la prima volta durante una recita alla scuola materna nel 1986." At the bottom of the interface is a control bar with three buttons: "Editing al vivo", "Modulo", and "Anteprima".

La schermata di editing al vivo, conosciuta anche come "testPage", è un ambiente ibrido in cui in alto trovate la barra degli strumenti Sar-At mentre in basso vedere una anteprima della pagina web come apparirà ai vostri visitatori. Se volete tornare alla visione modulare della pagina premete la freccia verso sinistra che vedete in alto a sinistra nella pagina. Se volete vedere la pagina web come la vedrebbe un utente finale del vostro sito, senza la barra degli strumenti Sar-At, premete il pulsante *Anteprima* in basso.

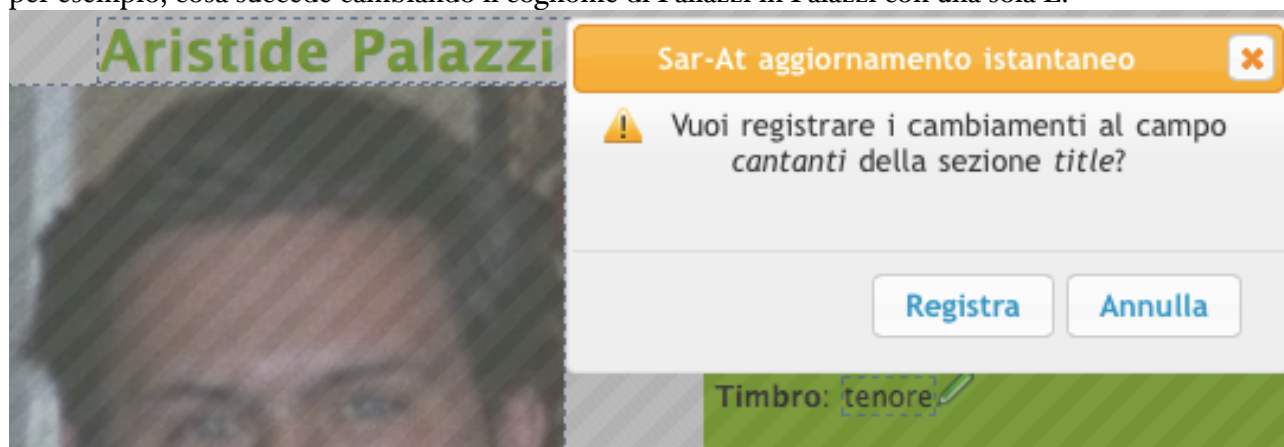
La caratteristica più interessante di testPage, però, è la sua capacità di lasciarvi ritoccare i contenuti del sito "al vivo".

Cliccate un testo qualsiasi tra quelli contenuti nel database e scoprirete che potete editarne il valore scrivendo direttamente nella pagina.

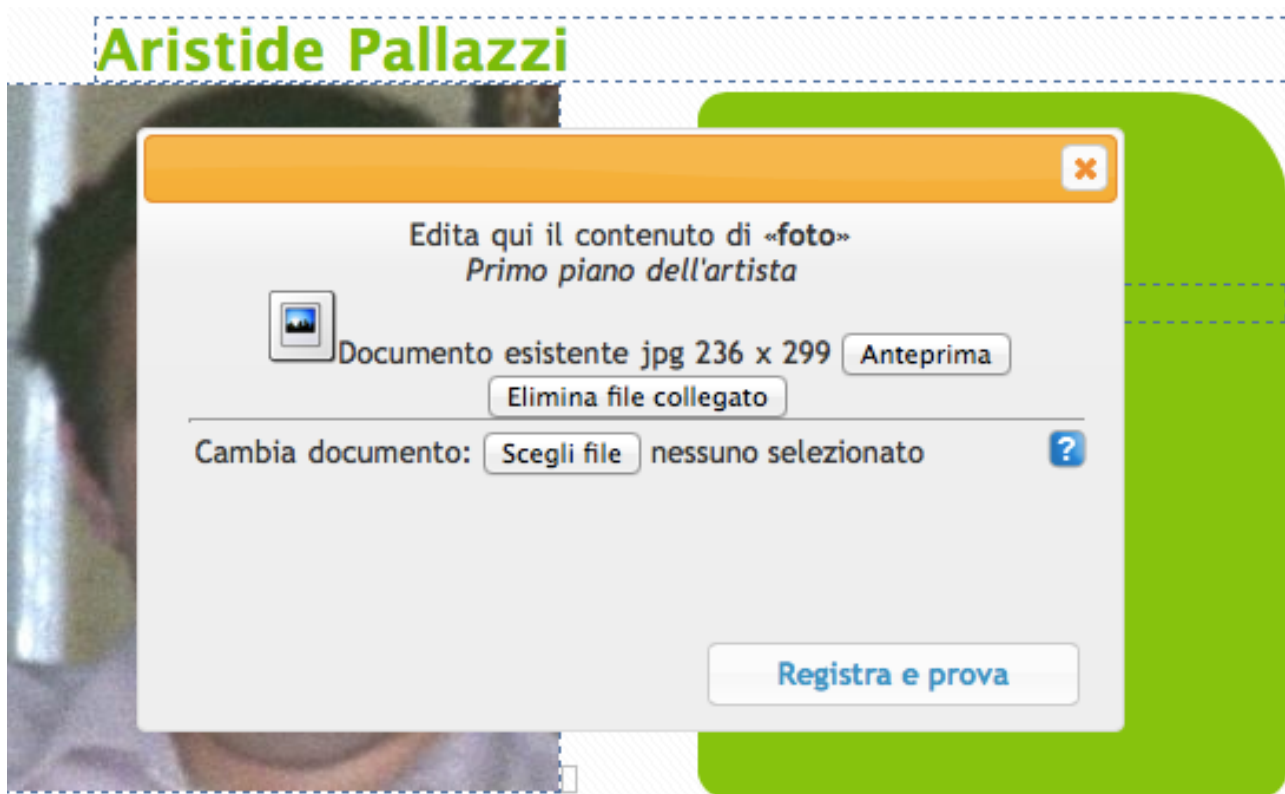




Tutti i cambiamenti vengono immediatamente applicati e sono in linea non appena confermate. Ecco, per esempio, cosa succede cambiando il cognome di Pallazzi in Palazzi con una sola L.



E per i campi che non contengono un testo ma qualcos'altro, come la fotografia, o la data di nascita? In questo caso non è possibile semplicemente scrivere una cosa qualsiasi alla tastiera e di conseguenza si usa un metodo differente; vedrete lì a fianco l'icona di una matitina. Facendo clic appare in sovrapposizione una finestrella galleggiante che consente le modifiche.



Potete spostare la finestrella cliccando sulla barra del titolo e trascinando. Potete ridimensionare la finestrella se lo desiderate cliccando sull'angolo in basso a destra.

Lo *editing al vivo* funziona solo con pagine mastro classiche, ma non con quelle in Twig che stiamo per presentare.

## Pagine mastro Twig

Sar-At consente la creazione di pagine mastro anche usando il sistema *Twig*. Si tratta di un componente *open source* il cui sito istituzionale si trova all'indirizzo [twig.sensiolabs.org](http://twig.sensiolabs.org). Rispetto a quanto spiegato nelle pagine precedenti, la differenza è minimale. Se volete provare Twig, duplicate la pagina mastro `cantanti.html` che avete creato poco fa e date alla copia il nome `cantanti.twig.html`.

Nella copia, cambiate leggermente i segnaposto. In Twig si usano due parentesi graffe, non una sola come nel sistema classico, e tradizionalmente viene lasciato uno spazio bianco tutto attorno al nome del campo. Quindi la riga HTML che in precedenza aveva questa forma:

```
<p><strong>Data di nascita</strong>: {date}</p>
```

Nella versione Twig diventa:

```
<p><strong>Data di nascita</strong>: {{ date }}</p>
```

### Filtri

Sin qui, vi verrà spontaneo pensare, se non è zuppa, è pan bagnato. Twig però offre una serie di opzioni ulteriori che il metodo classico non permette.

Per esempio, potremmo desiderare di visualizzare la data di nascita di Pallazzi mettendo il mese con una parola e non con un numero. Per soprammercato vogliamo usare il nome italiano, "settembre". Ecco qui:

```
<p><strong>Data di nascita</strong>:{{ date | localizeddate('%d %B %Y', 'none', 'it_IT') }}</p>
```

Il trattino verticale in Twig indica un filtro, cioè un sistema per prendere il dato grezzo proveniente dalla nostra base dati e trasformarlo in una forma più digeribile dal visitatore. Potreste chiedervi da dove arrivi il filtro `localizeddate`, e la risposta è che Twig ne offre una gran quantità, tutti documentati sul suo sito.

Un altro esempio. Immaginate di aver impostato nel vostro sito un commercio elettronico. Il prezzo di un prodotto, mostrato con `{price}` nel sistema tradizionale delle pagine mastro, viene prodotto da Sar-At così come si trova nel database. Milleduecento virgola cinquanta euro appariranno come `1200.5`

In Twig, possiamo usare il filtro `number_format`, e scrivendo `{{ price | number_format(2, ',', '.') }}` avremo a video un ben più leggibile `1.200,50`

### Abilitazione

Per vedere la pagina mastro Twig all'opera, andiamo nel menu Pagine, selezioniamo la nostra sezione (cantanti, nel caso dell'esempio), facciamo clic sulla linguetta *Pagine mastro* e nel menu selezioniamo `cantanti.twig.html`. Poi facciamo clic sul pulsante *Aggiungi*.

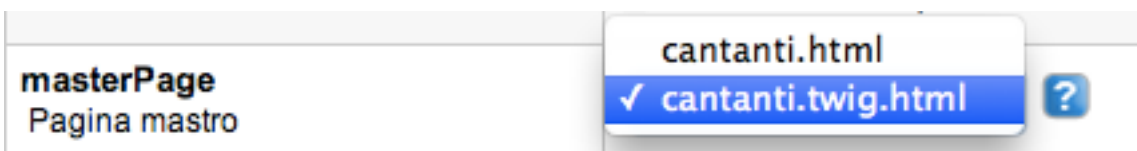
Sar-At distingue automaticamente le mastro classiche dalle mastro Twig che ci sia senza bisogno di indicarlo. Anche la presenza di "twig" nel nome è solo una convenzione, non una necessità.

Noterete che in questo momento la sezione cantanti ha due pagine mastro. Possiamo scegliere di utilizzare, per default, la versione Twig facendo clic sul pulsante radio nella sua riga e poi cliccando sottostante pulsante *Rendi default cantanti.twig.html*

Pagine mastro abilitate per la sezione <i>cantanti</i> ?				
Pagina Mastro HTML	Tipo	Default	Controllo Applicabilità	Azioni
<code>cantanti.html</code>	Foglia	<input type="radio"/>	21/08/2014 14:14:20	
<code>cantanti.twig.html</code>	Twig	<input checked="" type="radio"/>	21/08/2014 14:14:20	 Elimina

[Rendi default cantanti.twig.html](#)

Sar-At, comunque, consente anche di indicare la mastro individualmente, pagina per pagina. Se dalla linguetta *Dati* rientriamo nella scheda di Pallazzi possiamo usare la riga *masterPage* del Modulo per scegliere se questa pagina venga visualizzata attraverso questa o quella pagina mastro. Il che in questo caso non fa nessuna differenza, ma è utile nel nostro ipotetico sito dedicato ai cantanti lirici se volessimo, per esempio, una certa impaginazione per i tenori e un'altra per i baritoni.



Come si vede, nulla impedisce di usare per una medesima sezione alcune pagine mastro classiche e alcune Twig, come più è comodo per il progettista del sito.

### Per saperne di più

Twig ha una nutrita comunità di sviluppatori a cui ci possiamo rivolgere per cercare soluzioni e consigli, che saranno tutti automaticamente spendibili in Sar-At. Ecco un assaggio:

<http://stackoverflow.com/questions/tagged/twig?sort=votes&pageSize=15>

### Creazione di molte sezioni, layout complessivo del sito e moduli pronti

Durante la fase iniziale dello sviluppo di un sito vengono create molte sezioni, consecutivamente. Possono comunemente crearsi due situazioni -- che Sar-At vi aiuta a gestire.

*È necessario cominciare a fare data entry (creazione dei dati che popolano le pagine) prima che la grafica sia disponibile.*

In questo caso, potete creare la sezione indicando che desiderate usare una “pagina mastro a fil di ferro”. Sar-At genera un semplicissimo HTML che è sufficiente per una anteprima minimalista dei dati e lo usa come pagina mastro.

Le pagine mastro a fil di ferro sono particolarmente preziose quando viene generata una sezione modulo (di cui parliamo nel terzo capitolo di questo manuale). Lo HTML generato da Sar-At per i moduli (form) è perfettamente funzionante e gestisce i casi di errore più comuni (come la possibilità che un visitatore del sito lasci in bianco un campo obbligatorio, o tenti di iscriversi per una seconda volta con la medesima username). È molto comodo generare la mastro a fil di ferro e passarla all'impaginatore che crea la grafica attorno al codice funzionante.

*Tutte le sezioni da creare hanno un aspetto grafico molto simile tra di loro*

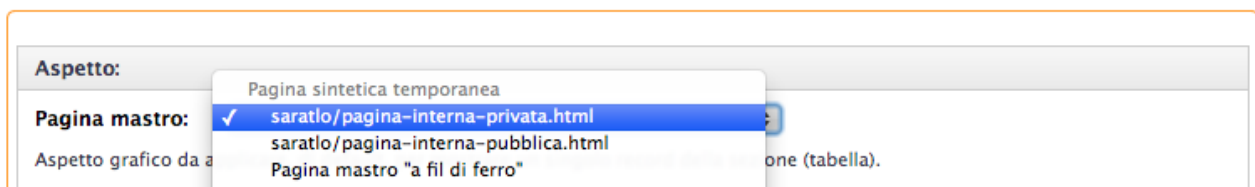
In questo caso è consigliabile creare una “super-pagina mastro” che venga adottata da tutte le sezioni. Sar-At si baserà su questa meta-pagina mastro per sintetizzare pagine mastro classiche individuali, una per ciascuna sezione. È perfettamente possibile avere più meta-pagine mastro. Per esempio, il grafico potrebbe creare un layout che va usato per mostrare elenchi (tutte le ricette, tutti gli eventi, tutti i prodotti in catalogo, tutti i più recenti post del blog...) e un'altra meta-pagina mastro con un layout per pagine di dettaglio (una scheda prodotto, una ricetta, un singolo post del blog con tutti i commenti...)

Chiedete al grafico di progettare la pagina, o le pagine, del layout indicando con il testo {SAR-AT-METAMASTER-CONTENT} il punto preciso in cui Sar-At dovrà inserire il contenuto specifico della sezione. Un modulo di iscrizione è un buon esempio di contenuto specifico di sezione.

Verificate che il layout generico così prodotto appaia correttamente caricandolo sul server insieme a tutte le immagini, script eccetera e digitando il suo indirizzo nel vostro browser (qualcosa come <https://www.nomesito.it/layout.html>).

Una volta finalizzato il layout nella pagina meta-mastro, dovrete spostare il solo file HTML generico (nel nostro esempio, il file “layout.html”) sul server in una sottocartella chiamata saratlo (contrazione di “layout per Sar-At”).

Ciò fatto, vi renderete conto che quando in Sar-At viene generata una nuova sezione, il programma si offre di applicarle automaticamente il layout generico. Potrete scegliere uno dei vostri layout generici da menu e Sar-At creerà una pagina mastro per voi che il grafico potrà comunque in seguito scaricare e ritoccare.



### L'indirizzo di una pagina nel sito

Le pagine web generate da Sar-At per i visitatori del vostro sito hanno un indirizzo che comincia con showPage.php se la pagina mastro è classica oppure showTwig.php se la pagina mastro è di tipo Twig.

ShowPage e showTwig sono due componenti gemelle di Sar-At che sintetizzano lo HTML partendo dalle pagine mastro HTML prodotte dal web designer; dalla progettazione (le Viste, descritte nel seguito di questo manuale) e dal contenuto inserito dai redattori attraverso il programma Sar-At stesso.

ShowPage e showTwig, naturalmente, hanno bisogno di qualche informazione: bisogna fare in modo che l'indirizzo di una pagina contenga il nome della sezione che si vuole visitare e

l'identificativo della singola pagina all'interno di quella sezione. Questo si ottiene passando parametri. Per esempio, la pagina di Aristide Pallazzi che abbiamo creato nell'esempio poco sopra è la prima della sezione cantanti e quindi il suo indirizzo sarà `showTwig.php?template=cantanti&id=1`

Vedremo tra pochissimo come sia possibile mascherare indirizzi funzionali come questo con altri, più amichevoli e memorabili. Per ora, però, è utile comprendere come funziona Sar-At

Alcuni parametri sono standard; uno solo, "template", è obbligatorio. Ecco i parametri standard elencati in tabella

Nome	Uso	Cosa accade se manca
template	Nome sezione corrente	Errore. Nessuna pagina viene generata.
id	Numero pagina corrente	Viene scelta la pagina più recente tra tutte quelle della sezione
masterPage	Pagina mastro (contesto grafico)	Viene scelta quella di default (indicata in Sar-At nella linguetta "Pagine mastro")
search	Valore da cercare nei motori di ricerca interni al sito	Viene cercato il valore ""

La gestione di *id* merita una riflessione. Il comportamento è comodo per chi ha una sezione aggiornata di frequente e vuole mettere in vista le più recenti modifiche. Per esempio, sul sito [sar.at.it](http://www.sar-at.it) abbiamo una sezione carnet che mostra i siti più prestigiosi realizzati con la nostra piattaforma. Chi visita l'indirizzo

`http://www.sar-at.it/showPage.php?template=carnet`

vede sempre la più recente aggiunta a quella sezione.

Nei nomi dei parametri, così come nel nome di `showPage`, maiuscole e minuscole fanno differenza. Fate quindi attenzione a non scrivere *masterpage* invece di *masterPage*, *Search* invece di *search* o altri refusi del genere. Possono essere piuttosto difficili da notare e correggere.

Se altri parametri vengono passati nell'indirizzo, Sar-At ne tiene conto e li memorizza come contesto della pagina. Questi parametri possono venire utilizzati nelle Viste: ne parleremo nel capitolo dedicato ad esse.

### id e masterId

In alternativa a *id*, `showPage` e `showTwig` accettano un altro parametro chiamato `masterId`. Questo è occasionalmente utile con i siti multilingue.

Immaginate di voler offrire il sito dedicato ai cantanti lirici in versione italiana, inglese e russa. Naturalmente sarà necessario che un traduttore metta mano alle schede ed esprima nelle tre lingue le biografie di ciascun cantante, ma il vostro lavoro di progettista di sito viene prima. Lavorando con la voce *Lingue* del menu *Sito* abiliterete tutte le lingue che il sito deve avere. Forse sarà necessario declinare le pagine mastro nelle altre lingue (questo è spiegato meglio più avanti in questo stesso manuale, ma comunque, potete facilmente capire che un sito in declinazione araba o israeliana dovrebbe venire ridisegnato in modo da avere i menu che scorrono da destra verso sinistra visto che quelle lingue si scrivono in quel modo).

Quando questa predisposizione è stata effettuata, un traduttore lavorando nel modo spiegato nel manuale *Mantenere il sito* predispose le traduzioni. Se la scheda di Pallazzi viene redatta in tre lingue, le tre traduzioni avranno tre *id* differenti ma tre identici `masterId`. Il `masterId` in sostanza è il sistema che Sar-At usa per capire che non si tratta di tre cantanti differenti, ma di tre traduzioni.

cantanti					
Titolo	Data Di Modifica	Lingua	MasterId	Id	Distruggi
Аристиде Паллазци	2014-08-21 14:42:00	русский	1	3	
Aristide Pallazzi	2014-08-21 14:06:00	italiano	1	1	
Aristide Pallazzi	2014-08-21 14:06:00	english	1	2	

La figura sottostante mostra la situazione in modo abbastanza chiaro.

L'utilità di chiamare showPage o showTwig passando il parametro masterId anziché id a questo punto comincerà a farsi più chiara. Sempre usando Pallazzi come esempio:

Questo indirizzo...	significa...
showPage.php?template=cantanti&id=1	La pagina di Pallazzi in italiano
showPage.php?template=cantanti&id=2	La pagina di Pallazzi in inglese
showPage.php?template=cantanti&id=3	La pagina di Pallazzi in russo
showPage.php?template=cantanti&masterId=1	La pagina di Pallazzi nella lingua corrente

Diciamolo con altre parole in modo che sia ancor più chiaro. Ogni volta che una pagina viene visualizzata da un visitatore del sito, anche un visitatore anonimo, Sar-At prende nota di quale lingua è stata appena letta. Se la pagina richiesta subito dopo ha un indirizzo con masterId e senza id, Sar-At sceglie automaticamente la variante in quella lingua.

### Indirizzi pseudo-statici e SEO

Sar-At supporta gli indirizzi pseudo statici, cioè che non appaiono prodotti da un sistema di gestione contenuti, di tipo amichevole come opera/aristide-pallazzi\_000001.html

Guardando una URL amichevole come questa non c'è modo di capire se la pagina HTML è stata generata via showPage o via showTwig.

Le URL amichevoli vanno impostate in fase di creazione del sito. Si seleziona un nome per la pseudo-cartella (nel caso del nostro esempio di poco fa, "opera") e Sar-At sa che tutte le URL che gli arrivano e che cominciano con quel nome vanno risolte chiamando appropriatamente showPage o showTwig. Il nostro programma mantiene internamente una mappa interna che lo informa di quale nome amichevole corrisponde a quale combinazione di programma (showPage o showTwig), sezione, id e pagina mastro.

### Le sezioni menu

Se avete provato a introdurre un po' di schede di cantanti, vi sarete forse resi conto di un potenziale problema. Nel campo "timbro" è possibile scrivere qualsiasi cosa. Non soltanto "baritono" o "tenore" o "soprano", ma anche (magari per errore) "barittono". Questo può essere un problema in un sito che è destinato a contenere molte centinaia di pagine.

Per rimediare, proviamo a creare un nuovo tipo di dati. Cliccate la linguetta *Sezioni* della nostra sezione cantanti fate clic ed editate il campo "timbro". Nel menu *Tipo* selezionate la voce *Nuovo gruppo di opzioni...* Sar-At fa partire un apposito wizard che, in pochi passi, vi permette di definire una mini-

sezione (chiamata nel gergo degli esperti di Sar-At una “sezione menu”) che conterrà tutte e sole le opzioni accettabili per il timbro dei cantanti.

### Nome menu

Nella prima schermata del wizard, *Nome menu*, in alto scegliete un nome per la sezione menu. In termini HTML, state definendo il *name* della vostra *select*.

Consigliamo di pescare un nome che comincia con “menu\_”, ma questa non è una regola, solo una convenzione.

Poco più sotto vedrete una casella di spunta chiamata *Viene aggiornato*. Se la spuntate, la sezione menu che state creando appare nel menu pagine e i redattori del sito possono aggiungere e togliere voci. Se non la spuntate la sezione è invisibile nel menu pagine e soltanto voi, che ne conoscete il nome, potete apportare modifiche all’elenco che essa definisce.

Scegli un’etichetta per il nuovo menu. La utilizzeremo in seguito quando vorremo includere il menu in un questionario.

**Nome menu:**

Il nome del menu può contenere solo lettere e cifre. Gli spazi bianchi non sono ammessi ma puoi usare il carattere \_.

---

**Viene aggiornato**

Se le voci del menu verranno periodicamente aggiornate, spunta questa casella.  
Se lo fai, troverai questo menu in Sar-At sotto alla voce "Pagine".

### Opzioni menu

Nella seconda e ultima schermata del wizard andate a inserire le voci del menu. In termini HTML ogni riga che inserite qui diventa una *option* e per ciascuna di esse potete scegliere sia il testo che il *value*.

Se il vostro sito è monolingua, Sar-At vi chiede di digitare soltanto il testo visibile ai redattori delle opzioni. In un sito di cantanti lirici monolingua, questa predisposizione genererà un HTML simile a questo

```
<select name='menu_timbri'>
  <option value='1'>Baritono</option>
  <option value='2'>Basso</option>
  <option value='3'>Tenore</option>
</select>
```

Se il vostro sito è multilingua, Sar-At vi chiederà tutte le traduzioni in tutte le lingue delle voci di menu, più un ulteriore testo da usare per la stringa del *value*. Per continuare con il nostro esempio del sito dedicato all’opera, potreste fare come nella figura sottostante:

#### Opzioni menu:

Italiano	English	Русский	Stringa	Elimina
Baritono	Baritone	Баритон	baritono	
Basso	Bass	Бас	basso	
Tenore	Tenor	Тенор	tenore	
<input type="text" value="italiano"/>	<input type="text" value="english"/>	<input type="text" value="русский"/>	<input type="text"/>	

#### Aggiungi voce

Risultato, per l’inglese questa predisposizione genererà un menu tipo questo:

```
<select name='menu_timbri'>
```

```

<option value='baritono'>Baritone</option>
<option value='basso'>Bass</option>
<option value='tenore'>Tenor</option>
</select>

```

Ma per l'inserimento dati in russo il menu che verrà generato da Sar-At è:

```

<select name='menu_timbri'>
  <option value='baritono'>Баритон</option>
  <option value='basso'>Бас</option>
  <option value='tenore'>Тенор</option>
</select>



```

La *stringa* deve, comprensibilmente, essere unica. Sar-At non vi consentirà di introdurre più voci con la medesima *stringa*.

Quando avete finito di inserire le alternative, fate clic su *Crea menu*.

### Richiamare una sezione menu

Da questo momento, il Sar-At riconosce e ci permette di utilizzare il menu\_timbri come sorgente di valori per creare un menu popup (una select, in termini HTML) o una serie di radiobutton, e i redattori del sito si vedranno proprio quel tipo di input quando compilano la scheda di un cantante. Noi impostiamo nella linguetta *Struttura* della sezione cantanti per il campo “timbro”:

Nome campo	timbro
Note	Basso, baritono o tenore
Tipo	Pulsanti radio (radiobutton)
<input type="text" value="menu_timbri"/>  	

I redattori e traduttori che usano la linguetta *Dati* e inseriscono le schede di Aristide Pallazzi e compagnia cantante si vedono davanti a questo:

<p><b>timbro</b></p> <p>Basso, baritono o tenore</p>	<p><input type="radio"/> baritono <input type="radio"/> basso <input checked="" type="radio"/> tenore</p>
--	---

Notate che i valori possibili vengono mostrati al redattore ordinati in ordine alfabetico.

### Tipo personalizzato menu


Riassumendo questo visto nel paragrafo precedente: voi potete definire una speciale sezione in Sar-At che non corrisponde a una vera e propria sezione sul sito pubblico. Questa sezione, chiamata *sezione menu*, è una sorgente di tutte le alternative valide per i valori di un campo di una sezione “vera”.

Per esempio, immaginate di aver creato una sezione menu chiamata “nazioni” e strutturata così:


Title	lingua	masterId	stringa
Stati Uniti d'America	italiano	1	US
Corea del Nord	italiano	2	KN
USA	english	1	US
North Korea	english	2	KN

Poi create una sezione chiamata “distributori” dove fate uso della sezione menu, in modo che di ogni distributore possa venire indicata la nazione di appartenenza scegliendola da menu. Con Sar-At automaticamente vengono visualizzati i testi espliciti (per esempio, “Corea del Nord” per gli utenti italiani) e saranno memorizzate le stringhe (come KN).



Modifica struttura della sezione <i>cantanti</i> ?	
Nome campo	timbro
Note	Basso, baritono o tenore
Tipo	Menu popup (select) 
<input type="text" value="menu_timbri"/>  	

Per un esempio concreto, torniamo ai nostri cantanti lirici e al campo “timbro” della loro sezione. Faccio clic sulla linguetta *Struttura* e un secondo clic sul campo “timbro”, poi vado a modificare il *Tipo* di

<b>timbro</b> Basso, baritono o tenore	<input type="radio"/> baritono <input checked="" type="radio"/> basso <input type="radio"/> tenore	
---	--	---

campo scegliendo il *menu popup (select)*.

Ora quando un redattore edita la pagina web dedicata a un cantante vedrà un menu di opzioni.

Notate che i valori possibili vengono mostrati al redattore ordinati in ordine alfabetico.

Se il numero delle opzioni alternative è notevole, Sar-At non mostra un menu (che sarebbe scomodo da srotolare, lungo com'è) ma invece uno spazio ad autocompletamento (meglio descritto in *Mantenere il sito*). Inizialmente Sar-At si predispone per mostrare al più 30 voci per ogni menu, ma questo valore è personalizzabile usando il menu *Sar-At*, voce *Preferenze*.

### Pulsanti radio

Quando le opzioni sono poche, come nel caso dei timbri di voce dei cantanti lirici, la massima comodità per il redattore si ha visualizzandole come una serie di alternative in pulsanti radio.

I pulsanti radio sono una tra le più semplici e comprensibili visualizzazioni per i valori accettabili di un campo. Il timbro dei cantanti lirici può venire comodamente rappresentato in questo modo:

<b>timbro</b> Basso, baritono o tenore	<input type="radio"/> baritono <input type="radio"/> basso <input checked="" type="radio"/> tenore
---	--

A tale scopo, quando lavorate nella linguetta *Struttura* dovete selezionare sotto *Tipo* la voce *Pulsanti radio (radiobutton)*. Immediatamente al di sotto appare il menu da cui dovete scegliere la sezione menu che funziona come sorgente di valori accettabili.

Tutto questo ha effetto solo per i redattori. Noterete che il pubblico, il quale guarda le schede dei cantanti sul sito, non vede né menu né pulsanti radio, ma solo il timbro del singolo interprete d'opera.

Nel database finirà registrata la *stringa* dell'opzione selezionata.

Un uso interessante. Immaginate di creare la sezione menu “sesso” inserendovi due voci: “title: femmina; stringa: r” e “title: maschio; stringa: o”. A questo punto Sar-At userà il valore zero per i maschi e uno per le femmine. In una pagina mastro classica voi potrete testare con grande semplicità il contenuto del campo usando il comando IF, che vedremo in dettaglio nel capitolo sei. Per esempio: {IF sesso}<img src='fiocco\_rosa.gif' />{ELSE sesso}<img src='fiocco\_azzurro.gif' />{ENDIF sesso}



### Caselle di controllo (checkbox)

Le caselle di controllo (o, in inglese, *checkbox*) sono il metodo più semplice per far scegliere zero, una o più opzioni tra una serie di alternative. Immaginiamo di voler aggiungere alle schede dei cantanti un campo “incisioni”, dove appariranno tutte le opere liriche che il personaggio ha interpretato e che sono finite su un disco. Vogliamo, in sostanza, che i redattori del sito possano compilare una interfaccia di questo tipo:

<b>incisioni</b> Quali opere ha interpretato	<input type="checkbox"/> <b>Barbiere di Siviglia</b> <input type="checkbox"/> <b>Flauto magico</b> <input type="checkbox"/> <b>Rigoletto</b> <input type="checkbox"/> <b>Traviata</b>
---	---

Notate che i valori possibili vengono mostrati al redattore ordinati in ordine alfabetico.

A tale scopo, quando definite il campo dovete selezionare sotto *Tipo* l'opzione *Caselle di controllo (checkbox)*. Immediatamente al di sotto appare un secondo menu da cui dovete scegliere la sezione menu che funziona come sorgente di valori accettabili.

<b>Nome campo</b>	incisioni
<b>Note</b>	Quali opere ha interpretato
<b>Tipo</b>	Caselle di controllo (checkbox) ▾
<input type="text" value="menu_opere"/>  	
Può essere nullo, zero o vuoto <input checked="" type="checkbox"/>	

menu_opere		
Titolo	Compositore	Distrucci
Aida	Verdi	<input type="checkbox"/>
Barbiere di Siviglia	Rossini	<input type="checkbox"/>
Flauto magico	Mozart	<input type="checkbox"/>
La cenerentola	Rossini	<input type="checkbox"/>
Le nozze di Figaro	Mozart	<input type="checkbox"/>
Rigoletto	Verdi	<input type="checkbox"/>
Traviata	Verdi	<input type="checkbox"/>

Se mostrate il contenuto del campo “incisioni” in una pagina mastro, i valori appariranno come una lista separata da virgole. Per esempio, se nella pagina mastro cantanti.html scrivete {incisioni} potrebbe apparire la scritta “Rigoletto, Traviata”.

### Raggruppamenti

Se la sezione menu\_opere finisse per venire popolata con tutte le più grandi opere liriche al mondo sarebbe davvero immensa e, di conseguenza, risulterebbe difficile per amministratori e redattori usarla per la compilazione. Sar-At, se la cosa vi è utile, può raggruppa le opzioni. Usando il campo *description* per farlo.

Andiamo nel menu *Pagine*, voce *Menu | Menu\_opere*. Usando la linguetta *Struttura* facciamo clic sul campo “description”. Sotto *Note* cancelliamo la scritta “Sintesi della pagina” e

rimpiazziamo con “Compositore”. Spuntiamo la casella *Editabile dagli amministratori* e togliamo la spunta a *Può essere nullo zero o vuoto*.

Ora torniamo alla linguetta *Dati*. Portiamo il mouse sulla testata della tabella dove ci sono le colonne Titolo...Distruggi. Appare un triangolo con la punta verso il basso e facciamo clic. Nel menu che appare e che mostra tutti i campi della sezione menu “menu\_opere” deseleggiamo i campi che non ci interessa visualizzare in tabella: autore, data di creazione, visibile, riservata. Selezioniamo invece il campo “description”

Ora passiamo alla linguetta *Display*. Nella riga “description”, sulla sinistra cancelliamo la scritta “Description” e la rimpiazziamo con “Compositore”, poi facciamo clic su *OK*.

Da questo momento la tabella che appare quando si fa clic su *Dati* ha l'aspetto estremamente compatto e utile che vediamo nell'illustrazione. Facendo clic sul titolo di ciascuna opera possiamo completare la descrizione compilando il nome del compositore. Quando viene aggiunta un'opera ulteriore, Sar-At richiede che il compositore venga specificato e non permette eccezioni.

Inoltre, da questo momento in poi quando amministratori e redattori del sito editeranno la scheda di un cantante, il campo “incisioni” avrà questa forma:



The image shows a form field labeled "incisioni" with the subtext "Quali opere ha interpretato". The field contains three grouped sections, each with a header bar and a list of radio button options:

- Mozart**
  - Flauto magico
  - Le nozze di Figaro
- Rossini**
  - Barbiere di Siviglia
  - La cenerentola
- Verdi**
  - Aida
  - Rigoletto
  - Traviata

## Multiparole

E se compositori ed opere fossero un elenco sterminato, con centinaia di opzioni? Il rischio è che lo spazio occupato a schermo da esse sia grande quanto un lenzuolo e anche i raggruppamenti che abbiamo appena visto servirebbero a poco. Per risolvere questi problemi Sar-At offre nella linguetta *Struttura* il tipo di campo “multiparole”.

Abbiamo visto che i pulsanti radio e i menu sono due modi diversi per visualizzare una serie di alternative e farne selezionare una sola. Analogamente, le caselle di controllo e i multiparola sono metodi alternativi per visualizzare una serie di alternative e farne scegliere nessuna, una o più d'una. In altre parole, esso consente di scegliere un numero a piacere di opzioni all'interno di un elenco.

Il multiparola appare mostrando tutte le opzioni raggruppate, secondo la medesima logica espressa nel paragrafo precedente. Scegliendo una opzione essa viene aggiunta nel campo di testo. Noi raccomandiamo l'uso del multiparola quando le opzioni alternative sono più di una decina e destreggiarsi con le caselle di controllo è scomodo.

Per vederlo all'opera proviamo a modificare una volta di più la tabella “cantanti” che abbiamo creato. Nella linguetta *Struttura* facciamo clic su “incisioni” e alla riga *Tipo* selezioniamo la voce *Multiparole*. Ora quando un amministratore di sito edita la scheda di un cantante vede questa interfaccia utente:

The screenshot shows a web interface for selecting opera titles. At the top, there is a search bar with the text "Digita cosa cercare" and two buttons: "Seleziona tutto" (checked) and "Deseleziona tutto" (unchecked). Below the search bar, the titles are grouped by composer:

- Mozart:**
  - Flauto magico
  - Le nozze di Figaro
- Rossini:**
  - Barbiere di Siviglia** (highlighted)
  - La cenerentola
- Verdi:**
  - Aida
  - Rigoletto
  - Traviata

Il multipare offre due gran comodità: primo, quando non è in uso viene collassato su una sola riga nel modulo compilabile della scheda cantante, quindi non intralcia il redattore. Secondo, offre un comodo motore di ricerca interno che rende la compilazione molto più spiccia quando le scelte sono davvero tante.

Anche in questo caso, le vostre scelte hanno influenza esclusivamente sul lavoro di redazione e il risultato finale non cambia. Se l'amministratore che esegue l'inserimento dati è diligente non importa se il campo "incisioni" è definito in *Struttura* come *Breve testo*, come *Caselle di controllo (checkbox)* o come *Multiparole*, alla fin fine nella base dati finisce un elenco di opere separato da virgole. Nel primo caso però è immensamente più probabile che il redattore commetta un errore umano.

Se lo desiderate, anche il campo incorporato *keywords* di Sar-At può venire visualizzato come un campo multiparola. Questo è comodo se volete costringere i compilatori a scegliere le parole chiave tra una lista predefinita e non liberamente. Per ottenere questo effetto dovete semplicemente definire una sezione menu chiamata *keyword* oppure *keywords*. Inseriteci tutte le parole chiave che preferite. Sar-At ne riconoscerà automaticamente l'esistenza e mostrerà il campo incorporato *keywords* come un campo multiparola.

## Portale

I campi di tipo portale sono presentati nel capitolo sulle Viste di questo manuale, perché la loro creazione richiede l'esistenza di una vista relazionale. L'uso dei campi di tipo portale da parte degli amministratori e redattori è presentato nel manuale *Mantenere il sito*.

## Alterazione di una sezione esistente

Come abbiamo già visto negli esempi precedenti, le sezioni esistenti possono venire modificate semplicemente facendo uso della linguetta *Struttura*. Possiamo aggiungere nuovi campi che si rendono necessari, o togliere campi esistenti che non servono più. Possiamo anche modificare la definizione dei campi esistenti.

Quando cliccate il pulsante "Aggiorna", Sar-At vi avvisa se la modifica è potenzialmente distruttiva.

Per esempio, cancellare un campo è sempre una modifica distruttiva perché tutte le informazioni di quel tipo in tutte le pagine esistenti vanno perse. Modificare il tipo di un campo è potenzialmente distruttivo solo in alcuni casi. Se abbiamo creato la scheda di un cantante scrivendo "falsetto" quando il campo accettava qualsiasi cosa, questa informazione viene persa se trasformiamo il campo in modo che accetti soltanto i valori di basso, baritono, tenore.

### I campi predefiniti di Sar-At

Quando voi create una sezione, Sar-At non trova spazio soltanto per i campi scelti e specificati da voi: automaticamente e inevitabilmente il programma inserisce in ogni sezione i campi predefiniti, che vengono presentati per la compilazione subito prima dei vostri nel modulo di creazione nuova pagina (menu *Pagine*, qualsiasi voce, pulsante *Aggiungi pagina...*)

Un elenco completo dei campi predefiniti si trova nel manuale *Mantenere il sito*. L'idea è che quei campi vengono compilati dagli amministratori, quindi è nel loro manuale che vengono dettagliatamente presentati.

I campi predefiniti sono ritoccabili (ma non eliminabili) da voi. Per esempio, potete editarne i valori di default.

Sia i campi predefiniti sia quelli definiti da voi possono venire resi invisibili agli amministratori, sempre usando la linguetta *Struttura*. Nel caso dei campi predefiniti, è utile nascondere quelli che non servono. Per esempio in una sezione menu il campo *keywords* non serve, mentre invece nella sezione "cantanti" ha abbastanza senso che i redattori vengano invitati a compilarla, per amor di Google.

### head della pagina mastro

Valorizzando i campi incorporati di Sar-At, nella pagina mastro di default di una sezione, come per esempio "cantanti", ha molto senso che la <head> ne faccia uso. Una pagina mastro di tipo classico, per esempio, ha molto senso che cominci pressappoco così:

```
<!DOCTYPE html>
<html lang='it'>
<head>
  <meta charset='utf-8' />
  <title>{title}</title>
  <meta name="description" content="{description}" />
  <meta name="keywords" content="{keywords}" />
</head>
```

### Campi unici

Sar-At vi permette di definire un campo come "unico". Se spuntate questa opzione, Sar-At verifica che ogni record (ogni pagina) di questa tabella (sezione) abbia un valore individuale in questo campo. In un catalogo prodotti, il codice prodotto è normalmente un campo unico. In una anagrafica dei cittadini, il campo "codice fiscale" è tipicamente un campo unico.

<b>Opzionale</b>	<b>Può essere nullo, zero o vuoto</b> <input type="checkbox"/>
	<b>Deve essere unico</b> <input checked="" type="checkbox"/>

"Id", il campo incorporato di Sar-At che assegna un numero intero a ogni pagina sul sito, è sempre e automaticamente un campo unico. Questo attributo non può venire rimosso al campo "id".

Le sezioni modulo, che descriveremo nel prossimo capitolo, sono spesso associate all'anagrafica dei visitatori -- ovvero, un visitatore che compila un modulo lasciandoci i suoi dati anagrafici si registra sul

sito. Vale la pena di notare che i campi unici sono tutti e soli i campi che l'utente registrato di ritorno potrà usare per farsi identificare. Per esempio, in una sezione in cui chiedete ai visitatori il nome, cognome e data di nascita (campi non unici); il codice fiscale, il codice cliente e l'indirizzo email (campi unici) questi ultimi tre potranno venire usati per fare log-in nel vostro sito. I dettagli sono nella parte settima di questo manuale, sotto "loginBuyer".

### Una grande sezione o tante piccole sezioni?

Per Sar-At, tutte le pagine dinamiche di un sito appartengono a una qualche sezione. Quando arriva il momento di creare le sezioni, molti responsabili si pongono un dilemma: accorpare o dividere? Per esempio, se stiamo preparando il sito di un mobilificio che produce tavoli, sedie e armadi, ci conviene creare una singola sezione chiamata "mobili" oppure tre sezioni, una per ciascuna tipologia di prodotto? Ciascuna delle due alternative offre qualche vantaggio.

I motori di ricerca, esperienza che tutti gli utilizzatori di Internet hanno ben presente in testa, favoriscono l'accorpamento. Da questo punto di vista è più naturale lavorare accorpando: se un visitatore sa che abbiamo in catalogo un oggetto chiamato "Alessandro Manzoni" può scriverne il nome in un motore di ricerca sul sito e trovarne la scheda prodotto, anche se il visitatore non si ricorda che il prodotto è una scrivania.

Viceversa, le diverse tipologie di prodotto hanno caratteristiche differenti che il visitatore potrebbe considerare discriminanti. Per esempio, quando un architetto sta cercando sedie di legno di ciliegio per il salotto, probabilmente non vuole fare una ricerca di "ciliegio" sul campo "materiale" e trovare gli armadi mischiati alle sedie.

La discriminante, alla fin fine, è proprio questa. Se vogliamo fare ricerche su una delle tipologie di prodotto alla volta, lavorando su una caratteristica peculiare della tipologia (per esempio: trova tutti i tavoli a tre gambe), allora conviene creare sezioni distinte. Se invece le ricerche lavoreranno su attributi comuni a tutti i prodotti (per esempio le dimensioni) allora è davvero consigliabile accorpare.

Nel prossimo capitolo di questo manuale (nel paragrafo "Viste su più sezioni") mostreremo una tecnica che permette di combinare i risultati di ricerche parallele su più sezioni differenti, all'interno di una singola pagina web. Tale tecnica consente di simulare una ricerca su più sezioni e rende più semplice il lavoro di chi ha scelto di creare sezioni distinte.

Chi invece sceglie di accorpare le sezioni troverà grandi benefici dall'uso delle pagine mastro alternative, o "pagine varianti", che presentiamo nel prossimo paragrafo.

### Le pagine mastro alternative

Abbiamo già introdotto nelle pagine precedenti di questo manuale il concetto di pagina mastro e la possibilità di avere più di una pagina mastro in una sezione. Per esempio potremmo desiderare una certa impaginazione per le schede dei cantanti lirici in attività e un'altra per quelli che non partecipano più alle rappresentazioni.

Un altro caso in cui servono pagine mastro differenti è quando stiamo progettando la navigazione del sito. Immaginate di voler creare una pagina home della sezione "cantanti" con un elenco di tutti gli interpreti lirici in attività. Sia l'elenco che la scheda personale sono pagine mastro della sezione "cantanti", ma ovviamente sono molto differenti. E se poi consentiamo al visitatore di cercare e trovare tutti gli esecutori di nazionalità scelta a piacere, anche in questo caso avremo bisogno di una distinta pagina mastro per la sezione "cantanti", il cui testo comincerà con "Abbiamo trovato i seguenti cantanti nati nella nazione da te scelta".

Tutto sommato, pagina elenco e pagina risultati della ricerca sono abbastanza simili e Sar-At ha la medesima (irrisoria) difficoltà nel prepararle. Cambia soltanto il metodo con cui viene scelto il criterio di ricerca e di ordinamento, nel primo caso vengono stabiliti da voi in quanto progettista del sito a tavolino, nel secondo caso li sceglie il visitatore con una interfaccia utente che voi avete realizzato.

Le pagine mastro “foglia”, quelle usate per mostrare un singolo elemento della sezione (per esempio un cantante) invece vanno di solito messe tutte a disposizione del redattore, che sceglie con criteri suoi quale applicare a ciascun elemento della sezione. Magari vogliamo avere un “interprete lirico del mese” la cui scheda appare in evidenza: dobbiamo quindi permettere a un amministratore del sito di preselezionare un cantante.

La pagina mastro di default di ciascuna sezione deve essere una pagina foglia. Questo è necessario perché i motori di ricerca di Sar-At usano sempre la pagina mastro di default quando devono visualizzare i risultati di una ricerca. La pagina home di sezione, la pagina risultati della ricerca nella sezione, eccetera, saran o pagina mastro varianti non di default.

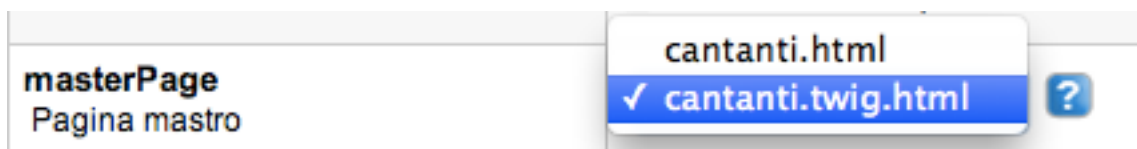
Voi preparate una pagina mastro alternativa in HTML e la caricate sul server web (tipicamente, usando SFTP). Entrate nel Sar-At e selezionate la sezione che ci interessa nel menu *Pagine*. Nelle linguette, cliccate *Pagine mastro*.

Pagine mastro abilitate per la sezione <i>cantanti</i> ?				
Pagina Mastro HTML	Tipo	Default	Controllo Applicabilità	Azioni
<b>cantanti.html</b>	Foglia	<input checked="" type="radio"/>	21/08/2014 14:14:20	
<b>cantanti.twig.html</b>		<input type="radio"/>	 <b>Verifica ora</b>	 <b>Elimina</b>
<b>cantanti_home.html</b>	Home	<input type="radio"/>	22/08/2014 11:20:51	 <b>Elimina</b>

Appare una schermata che elenca tutte le pagine mastro abilitate per questa sezione (la prima volta che visitate questa schermata, l'elenco mostrerà solo la pagina mastro iniziale). Selezionate dall'elenco *Aggiunta pagina mastro alternativa* il file HTML che avete appena creato e cliccate il pulsante *Aggiungi*. Sar-At esegue un controllo di applicabilità (di coerenza), analogo a quello che era stato fatto per la pagina mastro principale quando è stata creata la sezione.

Se (come si vede nella seconda riga della illustrazione qui sopra) appaiono una icona viola e il pulsante “Verifica ora”, avete modificato il documento HTML della pagina mastro di recente. Dovete premere il pulsante “Verifica ora” in modo che Sar-At possa rifare il controllo di applicabilità, altrimenti la produzione di pagine da parte di Sar-At non funzionerà. Se più di una pagina va verificata, appare anche un pulsante che consente di verificarle tutte simultaneamente.

Da questo istante, quando un amministratore di sezione modifica o aggiunge una nuova pagina in questa sezione, ha l'opzione di indicare che quella pagina di aprirà nel contesto grafico variante.



### Pagine home di sezione

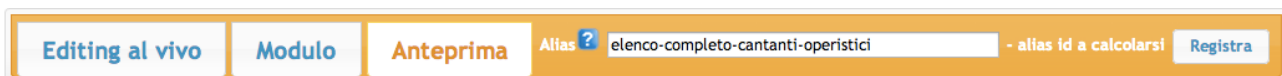
Passiamo all'altro caso: stiamo creando una pagina mastro alternativa che verrà utilizzata come parte integrante della navigazione, per esempio per mostrare i risultati di una ricerca, oppure un elenco delle altre pagine della sezione. Di solito, queste pagine mastro alternative contengono Viste, un argomento che affronteremo nel prossimo capitolo. La creazione della pagina variante avviene come nel caso precedente, e così anche il suo caricamento via SFTP. Ne regoliamo l'apparizione lavorando sulla URL, usando il parametro aggiuntivo *masterPage*.

Immaginate di aver preparato una pagina mastro chiamata `cantanti_home.html` che mostra i nomi di tutti gli interpreti lirici. Per leggerla scriviamo nel browser l'indirizzo `showPage.php?template=cantanti&masterPage=cantanti_home.html`

### Nome amichevole delle pagine home

Possiamo assegnare a una pagina home un indirizzo amichevole come `/opera/elenco-completo-cantanti-operistici.html` usando l'Anteprima.

1. Digitiamo la URL `showPage` (o `showTwig`) della pagina home, che avrà una forma simile a `showPage.php?template=cantanti&masterPage=cantanti_home.html`
2. Controlliamo di NON aver specificato un id. Questo sarebbe un grave errore perché se la pagina home della sezione "cantanti" resta associata all'id personale di Aristide Pallazzi, allora se Pallazzi viene cancellato dal sito svanisce anche la home page, lasciando il sito inutilizzabile.
3. Nella barra colorata in basso, sulla destra, digitiamo il nome amichevole desiderato. NON va indicato il nome della cartella (nel caso del nostro esempio "opera") né il suffisso `.html`. Quindi potremmo scrivere qualcosa come `elenco-completo-cantanti-operistici`



Se il riquadro "alias" non appare, il vostro sito non è predisposto per le URL amichevoli.

Fate clic sul pulsante **Registra**. Sar-At calcola un "alias id", ovvero un numero di sette cifre unico sul vostro sito, e lo assegna alla pagina. Poi ricarica la pagina usando il nuovo nome amichevole. Lo alias id non cambierà quando i redattori modificheranno il contenuto della base dati.

Da questo momento potete usare l'indirizzo amichevole nel vostro codice HTML per riferirvi alla pagina home.

### Pagine mastro localizzate

Se il vostro sito web offre più lingue, potrebbe essere necessario creare delle declinazioni delle pagine mastro, per tenere testa ai diversi requisiti delle diverse lingue. Per esempio, in media le parole tedesche sono molto lunghe e in inglese molto corte, e di conseguenza potreste trovarvi obbligati a lasciare più spazio ai testi nella versione tedesca del sito. Un esempio più estremo: in arabo e israeliano il testo è scritto da destra verso sinistra, quindi è necessario creare una variante dello HTML in cui la grafica è presentata in modo speculare. L'immagine mostra come si comporta Google quando vengono cercate immagini di gattini in italiano (a sinistra) e in arabo (a destra).



Sar-At gestisce automaticamente i layout varianti per lingua. Per ciascuna lingua dovete creare un documento HTML con le varianti del caso. Per esempio, se state creando lavorando sul sito dei



cantanti lirici, avrete una pagina mastro cantanti.html.it per mostrare la scheda di Aristide Pallazzi e colleghi ai visitatori italiani e una pagina mastro cantanti.html.ru per i visitatori russi.

Lo stesso semplice metodo si applica per le pagine mastro varianti. Per esempio potremmo avere una pagina risultati-ricerca.html.en per l'inglese e risultati-ricerca.html.ar per l'arabo. Sar-At utilizzerà automaticamente quella appropriata a seconda del contenuto che deve piazzarvi.

Nelle schermate di Sar-At in cui si abilitano e verificano le pagine mastro apparirà una sola voce, risultati-ricerca.html, che rappresenta entrambe le varianti.

#### Pagine mastro per dispositivi mobili

Se desiderate una declinazione del sito ottimizzato per smartphone, potete fare affidamento su Sar-At per utilizzare automaticamente una pagina mastro studiata in HTML per schermi particolarmente piccoli. Continuando l'esempio del paragrafo precedente, dovrete sviluppare una pagina mastro chiamata risultati-ricerca.html.en-mobi ed un'altra con il nome risultati-ricerca.html.ar-mobi

#### INCLUDE e pagine mastro localizzate

Le pagine mastro classiche di Sar-At (ma non le pagine Twig) consentono di ritagliare gli elementi comuni di tutte le pagine in documenti esterni. Per esempio, se la testata e la navigazione di tutto il vostro sito è identica in ogni pagina, potreste creare un file testata.ihtml e riutilizzarlo ovunque scrivendo

```
{INCLUDE testata.ihtml}
```

INCLUDE è descritto compiutamente nel paragrafo "Inclusione di frammenti HTML" del capitolo 6, "Funzioni avanzate".

Se decidete di usare il meccanismo INCLUDE, tenete presente che anche in questi file è possibile utilizzare i suffissi di lingua e il suffisso -mobi per le pagine ottimizzate per schermo piccolo. Quindi potremo avere testata.ihtml.it e testata.ihtml.en. Dentro alla pagina mastro potete scrivere semplicemente

```
{INCLUDE testata.ihtml}
```

e lo showPage di Sar-At penserà automaticamente a includere la variante linguisticamente corretta. Il medesimo principio vale per le inclusioni operate dai motore di ricerca incorporati, search\_site7 e search\_engine

#### IF e pagine mastro

Nelle pagine mastro è possibile condizionare l'apparizione di un blocco HTML al contenuto di un campo della base dati. Per esempio, mostrare alcune informazioni su un cantante lirico solo se il visitatore si è registrato sul sito.

Per Twig, la sintassi da usare è presentata sul suo sito. Per le pagine mastro classiche, una spiegazione estesa è nell'appendice A di questo stesso manuale.

## Velocità di produzione delle pagine

Man mano che i visitatori affollano il sito Sar-At comincia a generare le pagine web richieste da ciascuno. Combinare la pagina mastro e i contenuti del database è un lavoro che Sar-At svolge particolarmente in fretta. Se il vostro sito conta poche migliaia di pagine, potete ignorare queste righe e passare oltre, ma se il vostro sito conta dalle diecimila pagine in su vi consigliamo di passare qualche momento a riflettere su come sia possibile velocizzarlo il più possibile.

#### La cache

Ogni pagina una volta generata viene salvata, se possibile, nella memoria cache e da quel momento viene inviata a ogni successivo visitatore che la richiede senza che sia necessario ricalcolarla, a una velocità elevatissima.

Abbiamo scritto “se possibile”. Non tutte le pagine possono finire in una cache, però. Per esempio, una scheda prodotto è un buon candidato alla cache, perché i prodotti cambiano di rado, ma un carrello della spesa in un commercio elettronico non va mai messo in cache, perché è diverso per ogni singolo visitatore.

Le pagine mastro classiche e le pagine mastro Twig sono lavorate da Sar-At in modi molto diversi tra di loro e quindi la loro capacità di venire messe in cache e di venire prodotte in fretta sono dissimili.

### Twig e la cache

Le pagine mastro scritte in Twig vengono tradotte in linguaggio PHP la prima volta che un visitatore ne richiede la visione. Il PHP risultato viene messo in cache e nuovamente eseguito per ogni visitatore successivo. Quindi tutte le pagine Twig sono compatibili con questa forma di cache e noi diremo che hanno una velocità media di produzione.

Qualcuno sulla base di questa considerazione vorrà riscrivere qualche pagina mastro delle due da classica a Twig, ed è una buona idea. Attenzione, però. In Twig maiuscole e minuscole fanno differenza, mentre nelle pagine mastro vecchio stile di Sar-At questo non è vero. Quindi e per esempio, una pagina mastro vecchio stile che contiene {ID} funzionerebbe, ma la sua trasformazione pedissequa a Twig produrrebbe {{ ID }} che non verrebbe riconosciuto né accettato.

### Cache delle pagine mastro classiche

Il programma, la prima volta che vede una pagina mastro classica (cioè quando voi effettuate l'aggiunta o la validazione nella linguetta *Pagine mastro*) calcola se lo HTML che verrà prodotto rappresentando la pagina possa venire messo interamente in cache e poi spedito a ogni successivo visitatore senza ulteriori elaborazioni PHP.

Questo calcolo è particolarmente complesso. Basti comunque dire al momento della validazione e inoltre quando se voi visitate la linguetta *Pagine mastro* e fare clic sul nome di una pagina mastro classica, Sar-At vi informa se essa sia compatibile o meno con la cache.

Pagina Mastro HTML	Compatibilità	Azioni
cantanti.twig.html	Compatibile con la cache veloce di Sar-At	
cantanti.html	52	Elimina
cantanti_home.twig.html	52	Elimina

Una pagina mastro classica compatibile con la cache è estremamente veloce in produzione. Una pagina mastro classica incompatibile con la cache è il caso meno performante possibile.

### Tipo delle pagine mastro

Nella schermata delle pagine mastro Sar-At vi riporta anche un'altra valutazione: la natura della pagina. Per Sar-At tutte le pagine mastro appartengono a uno di sei tipi diversi.

Tipo	Possibile per	Spiegazione
<b>Foglia</b>	Classiche	È una pagina mastro che prende dal database molte informazioni da un singolo record. Per esempio, la pagina con la scheda di un cantante lirico. Una pagina foglia può anche contenere una o più Viste

Tipo	Possibile per	Spiegazione
<b>Home</b>	Classiche	È una pagina mastro che mostra poche informazioni prese da molti record. Per esempio, una pagina che mostra tutti i nomi di tutti i cantanti lirici e cliccandone uno permette di andare alla relativa scheda
<b>Ricerca</b>	Classiche	Pagina mastro destinata a contenere i risultati di una ricerca predisposta da un visitatore.
<b>Dubbia</b>	Classiche	Apparentemente destinata a mostrare contemporaneamente sia il risultato di una ricerca che informazioni da una singola scheda. Il che è probabilmente sbagliato.
<b>Modulo</b>	Classiche	Apparentemente destinata a contenere un modulo che il visitatore del sito può compilare. Si veda il capitolo dedicato alle sezioni modulo, più avanti.
<b>Twig</b>	Twig	Twig non permette a Sar-At di guardare “dentro” a una pagina e quindi il programma non può dire nulla sull’uso e appropriatezza della mastro.

## Newsletter

Sar-At permette di creare e inviare newsletter impaginate in HTML a un numero anche molto grande di destinatari. Le newsletter richiedono pagine mastro realizzare con qualche accorgimento in più.

### Le pagine mastro e le newsletter

Per Sar-At, una newsletter è una qualsiasi pagina del sito (collegata o non collegata ai normali percorsi di navigazione) che noi scegliamo di spedire. Tuttavia, come web designer dovete essere coscienti del fatto che non tutte le pagine HTML sono adatte a venir spedite per email. Quando producez una pagina mastro per newsletter ci sono due regole che dovete rispettare e alcuni consigli da seguire.

Regola 1: tutti i riferimenti a immagini, CSS, animazioni Flash e altri file separati dalla pagina HTML devono venire riportati con un indirizzo esplicito e assoluto. Per esempio, dovrete usare

```
<img src='http://www.peresempio.com/img/fotografia.jpg' />
```

e non sarà invece valido

```
<img src='img/fotografia.jpg' />
```

Le immagini vanno dunque caricate sul server web e lì restano, mentre lo HTML con CSS in linea viene separatamente spedito per posta elettronica.

Regola 2: il Javascript non viene eseguito all’interno di una email in HTML. Quindi, evitatelo. Se desiderate un effetto *rollover*, per esempio, dovete fare uso di un CSS.

I consigli: la newsletter verrà mostrata dentro a un programma email, quindi non potrà occupare un grande spazio a video. Contenetene le dimensioni.

Per un utente PC che usa Explorer e Outlook, la pagina verrà mostrata dentro Outlook usando il motore di Explorer. Per un utente Mac che usa Safari e Apple Mail avverrà un processo analogo. In tutte le altre combinazioni, la rappresentazione dello HTML verrà effettuata da un motore molto inferiore per qualità, potenza, modernità. Non contate sul fatto che qualsiasi effetto HTML appaia nelle email bene tanto quanto appare in un programma di navigazione del web. Progettate newsletter il più semplici possibili.

### Mailmerge

Le newsletter spedite da Sar-At possono venire personalizzate con informazioni anagrafiche dei destinatari. Un esempio usando la sintassi classica: “caro {nome} ti inviamo le nuove iniziative del mese di agosto, il prodotto del mese è {prodotto}”.

In altre parole, la pagina mastro può piazzare nella pagina mastro simultaneamente sia informazioni provenienti dalla sezione newsletter sia informazioni provenienti dalla anagrafica clienti (lo fa in quest'ordine, quindi se voi per caso avete un campo con il medesimo nome in questa e quella sezione, vedrete apparire il contenuto preso dalla sezione newsletter e non quello della sezione anagrafica.

Aggiunta mastro alternativa ?

Aggiungi

standard, per il sito  newsletter con mailmerge da

Aggiungi

Quando create una pagina mastro da usarsi per una newsletter, Sar-At vi chiede se desiderate integrare informazioni anagrafiche dei visitatori in mailmerge e, se sì, da quale sezione.

### Push

Sar-At supporta un tipo particolare di newsletter, super-personalizzate, in cui i destinatari non ricevono tutte le comunicazioni che voi erogate, ma solo quelli attinenti agli specifici prodotti/argomenti di loro interesse.

Nottetempo, il Sar-At esaminerà tutte le novità pubblicate dai redattori in giornata, le raggrupperà per argomento e spedirà una singola mail a ogni visitatore registrato del sito che sceglie di avvalersi del servizio push.

Per attivare il sistema push si procede come segue.

1. Si crea una sezione chiamata keywords la quale contiene tutte le tipologie di argomenti di cui ci si occupa sul sito. Questo elenco, che può essere localizzato (cioè i nomi delle categorie possono venire tradotti nelle varie lingue in cui il sito si articola) servirà ai visitatori del sito per indicare quali sono i loro interessi. Nulla vi impedisce, ad attività iniziate, di aggiungere o togliere voci. Mettete i nomi delle categorie nel campo "title". Le categorie possono venire raggruppate per affinità: indicate il gruppo di appartenenza nel campo "description".
2. Nella sezione anagrafica che accoglie i dati dei visitatori registrati, verificate che il campo "keyword" sia visibile ai redattori (usando la linguetta Struttura). In questo spazio verranno memorizzate le scelte dei visitatori
3. Nella sezione anagrafica che accoglie i dati dei visitatori registrati, usando la linguetta Modulo, nel box chiamato "push dei contenuti" indicate quali sono le sezioni oggetto del push. Tutte e sole le sezioni del sito indicate qui verranno segnalate ai visitatori (vedi immagine).

**Push dei contenuti**  
Sezioni di cui notificare le aggiunte

HTML per le mail:

Crea un nuovo file chiamato *push-potenziali.html* se necessario

Usa file *push-potenziali.html* esistente

Sovrascrivi file *push-potenziali.html* esistente

4. Al completamento del punto precedente, Sar-at genera automaticamente per voi una speciale pagina mastro in HTML che verrà utilizzata per contenere i dati del push e che verrà spedita ai visitatori, il cui nome è sempre push-{nome del gruppo di visitatori}.html. Potete scaricarla dal server e

Info	
Caratteristica	Valore Corrente
Nome sezione	<input type="text" value="acquirenti"/> <input type="button" value="Rinomina"/>
Tipo sezione	<p><b>Modulo sicuro.</b> Compilata dai visitatori del sito; le password sono protette e non ricostruibili neppure da chi ha pieno accesso alla base dati del sito. I visitatori possono aggiornare i dati che hanno inserito in moduli compilati da loro stessi. I visitatori che compilano compiutamente il modulo, che è un modulo di iscrizione, vengono iscritti nel gruppo di utenti registrati <i>potenziali</i>. Si tratta dei clienti del commercio elettronico.</p>

- impaginarla come meglio credete, sempre tenendo presente che essa verrà spedita per email e quindi valgono le raccomandazioni e le peculiarità descritte in questo stesso manuale alla pagina precedente.
- Verificate che in tutte le sezioni indicate al punto 3 il campo “keyword” sia visibile ai redattori (usando la linguetta Struttura in ciascuna di esse).
  - Quando inserite contenuti in una delle sezioni indicate al punto 3, il redattore dovrà usare il campo “keyword” per indicare a quali argomenti è attinente il contenuto della pagina.
  - Nella pagina in cui il visitatore si iscrive al sito, o aggiorna la sua anagrafica, dovrà essere presente ed editabile il campo keyword, e pubblicata una spiegazione che presenta ad essi il meccanismo.

## Informazioni sulla sezione

La linguetta *Proprietà* raggruppa in una singola schermata le caratteristiche modificabili di una sezione.

Il pulsante *Verifica ora* è documentato nel manuale *Governare il sito*.

In alto, il pulsante “Informazioni” (i bianca dentro a un cerchio blu) permette di accedere alle informazioni e ad alcune opzioni raramente necessarie.

### Nome sezione

Qui è possibile cambiare nome alla sezione, ma l’opzione non va usata alla leggera, specialmente se il sito è già in linea da un pezzo. Infatti, se il vostro sito contiene link in formato tradizionale alle pagine di questa sezione, ovvero

`showPage.php?id=numero&template=nomesezione`

allora sarà necessario editare manualmente tutti i link esistenti. Sar-At automaticamente aggiornerà tutte le Viste semplici e complete che fanno riferimento alla sezione, ma non potrà intervenire su quelle per esperti, né sulle statistiche scritte a mano in SQL dai programmatori. Anche gli eventi SQL che fanno riferimento alla sezione andranno controllati a mano da un esperto.

### Tipo sezione

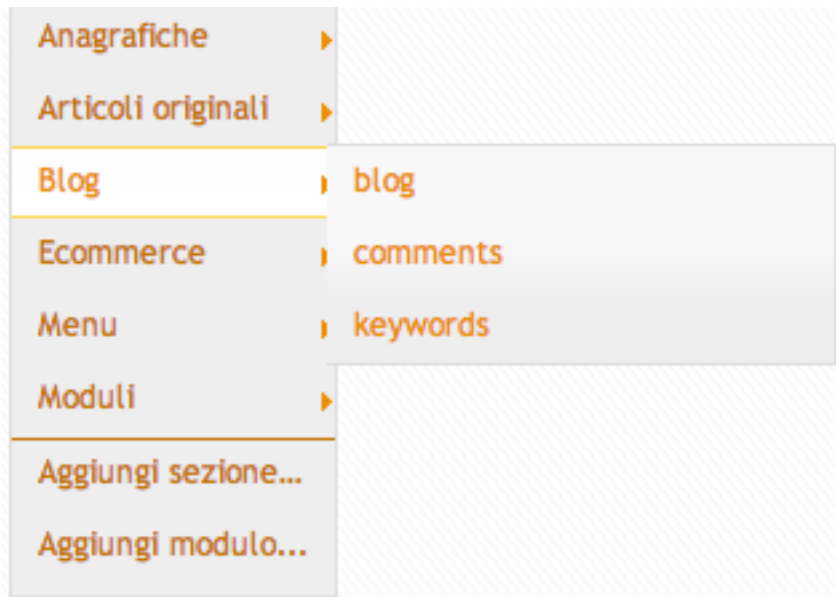
Una descrizione a parole della tipologia della sezione, in italiano semplice e facilmente comprensibile.

### Descrizione della sezione

Il testo inserito qui apparirà come aiuto contestuale quando un redattore seleziona il nome della sezione nel menu Pagine.

### Visibile nel menu Pagine

Intervenendo in questo box delle Proprietà della sezione possiamo rendere una sezione invisibile nel menu Pagine. Questo è utile nei casi in cui i redattori non debbano mettere le mani nei contenuti di una sezione. Per esempio, in un sito che offre video digitali, potrebbe essere opportuno creare una sezione di tipo menu che contiene due sole righe, “quattro terzi” e “sedici noni”. In questo caso è certo che non ci sarà bisogno di aggiungere o togliere nulla dalla sezione, la quale potrà venire nascosta. Questo tra l’altro rende il menu *Pagine* più corto e dunque più facilmente maneggiabile.



In caso di necessità, tutte le sezioni invisibili possono venire rese nuovamente visibili, per tutto il tempo necessario, usando il menu *Sar-At*, voce *Preferenze*. Va spuntata la casella *Mostra le sezioni di servizio nel menu Pagine*.

Le sezioni possono anche venire raggruppate, di modo che il menu Pagine divenga gerarchico, come mostrato nella figura.

Per ottenere questo effetto è sufficiente selezionare nel menu “visibile sotto...” il gruppo in cui si desidera trovare la sezione corrente.

Per creare un nuovo gruppo, scegliere nel menu *Visibile sotto...* la voce *Altro gruppo*, poi digitare il nome desiderato.

#### Dimensione su disco

Indica lo spazio occupato sul server dalla base dati. Il pulsante *Distruggi la sezione*, qui posizionato, permette di eliminare completamente la sezione -- compresi tutti i dati in essa contenuti, gli eventi e le Viste. Sar-At non elimina le pagine mastro e le statistiche.

#### Sezioni: risposte a domande frequenti

##### La pagina home usa una pagina mastro variante

*La pagina home del mio sito, una rivista on-line, mostra i tre articoli aggiunti più di recente al sito. Tecnicamente, è una pagina mastro alternativa della sezione “articoli”. La vedo perfettamente se uso questo indirizzo nel browser:*

`http://www.ilsito.it/showPage.php?template=articoli&masterPage=home.html`

*Però io voglio che questa pagina venga presentata ai visitatori che accedono per la prima volta al sito, scrivendo:*

`http://www.ilsito.it/`

*Come mi comporto?*

Crea un file di testo chiamato `index.php` e incolla al suo interno questa riga di codice:

```
<?php
header("Location: showPage.php?template=articoli&masterPage=home.htm");
?>
```

Carica il file sul tuo server.

Quando un sito viene attivato per la prima volta in Sar-At potete specificare l'indirizzo della pagina home e il file `index.php` appropriato verrà automaticamente creato per voi. Le istruzioni qui sopra sono dunque utili quando cambiate qualcosa nella pagina home a cose fatte.

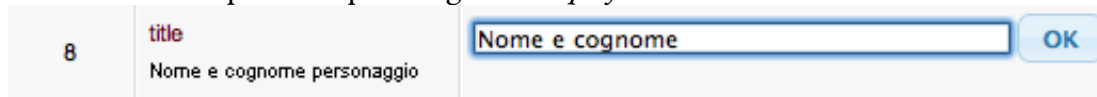
#### Nomi delle colonne

*Quando i redattori della mia rivista fanno clic nel menu Pagine, trovano anche la sezione Personaggi che io ho appena creato. Nella sezione Personaggi il nome e cognome viene registrato nel campo incorporato “title” di Sar-At. Forse i miei redattori sono tonti, fatto sta che mi scrivono “ing” o “dott” nella colonna titolo e poi si lamentano che manca lo spazio per nome e cognome. Come me la cavo?*

Il campo `title`, come tutti i campi incorporati standard di Sar-At, non può venire rinominato. Tuttavia il tuo problema ha una soluzione facile.

Per quanto riguarda l'inserimento dati: clicca la linguetta *Struttura*, poi clicca il campo “title”, e inserisci una spiegazione nel campo *Note*. Per esempio: “titolo pagina, nome e cognome del personaggio”.

Per quanto riguarda la tabella con l'elenco di tutte le pagine esistenti, che appare quando si fa clic sul menu *Pagine* e si sceglie il nome di una sezione, puoi cambiare a piacimento i nomi che appaiono in testa alle colonne. Usa a questo scopo la linguetta *Display*.



The screenshot shows a table with a header row. The first column contains the number '8'. The second column is labeled 'title' and contains the text 'Nome e cognome personaggio'. To the right of the table is a text input field with the placeholder text 'Nome e cognome' and an 'OK' button.

## Cloni

*Nella mia sezione Prodotti è necessario allegare una foto dell'oggetto. Mi è chiaro che posso farlo usando un campo di tipo File. Però il mio sito è in cinque lingue, dunque ogni prodotto ha cinque schede, una per ogni lingua. Non mi piace l'idea che ogni singola foto sia registrata sul server cinque volte, duplicandola inutilmente. Ma c'è di peggio: nella pagina del prodotto io devo offrire un PDF scaricabile, e quel PDF al contrario della foto deve essere localizzato, dunque tradotto, dunque differente in ogni caso (in ogni lingua). Come me la cavo?*

Te la cavi bene. Innanzitutto controlla di aver abilitato il Localization manager nella sezione prodotti, spuntando "Verifica che ogni pagina della sezione venga localizzata in ogni lingua" nella linguetta Proprietà.

Ciò fatto, nella linguetta Struttura del campo PDF spunta l'opzione "Deve essere unico".

Risultato: quando si inserisce un nuovo prodotto se ne carica la foto. Per creare la versione tradotta, si usa il pulsante + verde che appare di fianco alla bandierina (i dettagli sono nel manuale "Mantenere il sito"). Sar-At crea un nuovo record, ci mette dentro un riferimento alla medesima foto, duplica il PDF (quest'ultimo comportamento è una novità della versione 9). A questo punto al redattore basta sovrascrivere il testo originale con la traduzione e sostituire il PDF con la versione localizzata.

## That's all folks

Questo conclude la nostra prima passeggiata esplorativa dentro Sar-At. Prima di passare al secondo capitolo, dove vedremo come realizzare i motori di ricerca e come strutturare le pagine indice, vi suggeriamo di passare un po' di tempo a riguardare le funzionalità Sar-At che abbiamo presentato qui. Provate a creare una nuova sezione e a popolarla. Modificate il contenuto sia nella modalità a moduli sia nella modalità al vivo.

# Parte seconda: le viste

Creando una *vista*, noi possiamo estrarre un elenco di dati depositati dentro al nostro sistema. Se preferite, potete considerarla come un punto di vista (da cui il nome), o una selezione di dati. Se abbiamo un catalogo di cinquemila prodotti, per esempio, possiamo creare una vista sui prodotti che costano meno di dieci Euro, e inserirla in una pagina web. Anche quando l'assortimento di prodotti a catalogo o il prezzo di listino cambiano noi non dobbiamo preoccuparci: quella pagina mostrerà sempre e soltanto i prodotti che in un dato istante costano meno di dieci Euro.

Il criterio di inclusione in una vista può essere semplicissimo, come nell'esempio poco sopra, o più complicato. Possiamo chiedere di inserire in una vista tutte le pagine modificate nell'ultimo mese, oppure le schede del personale il cui cognome comincia con le lettere A-L, oppure gli articoli il cui titolo contiene la parola "novità".

Se il nostro catalogo a stampa contiene una sezione "ultime notizie", possiamo creare una vista che le include tutte e ci siamo così procurati un efficiente indice dell'intera sezione.

Una vista è una regola di selezione, che noi definiamo e a cui diamo un nome. Per esempio, la vista «prodotti che costano meno di dieci Euro» ci cui parlavamo sopra. Sar-At registra la vista e la utilizza (esegue la selezione) quando la trova riferita dentro una pagina mastro.

Sar-At consente la creazione delle Viste in tre modalità; semplificata, completa e per esperti.

## Modalità semplificata

Per creare una vista in modalità semplificata noi clicchiamo la linguetta "Viste" in alto, premiamo il pulsante "Aggiungi vista" e compiliamo il modulo che appare.

In alto dobbiamo specificare il nome della vista (possiamo scrivere qualsiasi cosa – si tratta in effetti di un commento, un promemoria, che Sar-At registrerà a nostro vantaggio ma che non utilizzerà). Di seguito scriviamo la etichetta della vista. In questo caso, invece, si tratta di una singola parola (deve contenere soltanto lettere, numeri e il carattere =\_").

Nome ed etichetta ?		*Campi obbligatori
Nome della Vista (un semplice promemoria)	<input type="text"/>	
Etichetta della Vista (da usare nelle pagine mastro)	<input type="text"/>	

La etichetta della Vista verrà usata per chiedere al Sar-At di eseguire la selezione. Ogni volta che Sar-At trova l'etichetta dentro una pagina mastro del nostro sito web, la selezione viene eseguita e il risultato appare nella pagina, formattato in HTML.

## Selezione e filtro

Nel modulo di definizione delle viste semplificate troviamo le regole di selezione e filtro. Questo è il riquadro più importante da compilare.

Il caso più semplice e più comune è il primo: la Vista non è filtrata. In questo caso, Sar-At restituisce tutti gli elementi della sezione. Per esempio, se vogliamo creare un indice dei nostri prodotti allora creiamo una vista non filtrata che restituisce un elenco di tutti i record (tutte le pagine) della sezione "prodotti".

La voce "verranno presentate contemporaneamente informazioni da non più di ... record" in questo caso va lasciata in bianco oppure (è lo stesso) va scritto il numero zero, che Sar-At in questo caso prende nel senso di "nessun limite".



**Selezione e filtro ?**

La Vista conterrà pagine in

La Vista non è filtrata e mostra tutti i record della sezione *bibliografia*.

Filtra, e mostra i soli record in cui

il valore   
  della pagina corrente ?  
 un valore scelto dal visitatore, e la pagina HTML invocherà questa Vista passando il parametro *search* ?

Effettua una ricerca a tutto testo ?

La ricerca a tutto testo non è disponibile perché non sono stati selezionati i campi multipli su cui può operare.  
[Seleziona i campi per la ricerca a tutto testo.](#)

Verranno presentate contemporaneamente informazioni da non più di  record.

Immaginiamo di voler trovare solo i prodotti che costano meno di dieci Euro. In questo caso selezioneremo il pulsante *Filtra e mostra solo i record in cui il prezzo è inferiore a*. Subito sotto, selezioneremo il pulsante *il valore* e digiteremo “10”

Le opzioni *numero caratteristico del record corrente* e *un valore scelto dal visitatore* si usano in casi speciali, che presentiamo nel seguito di questo capitolo: le relazioni e i motori di ricerca, rispettivamente.

### Ordinamento nelle viste

Le viste restituiscono i loro risultati in un ordine specificato da noi.

**Ordinamento ?**

Mostra i record ordinati per

Presenta i risultati in ordine

ascendente (A-Z, gennaio-dicembre)

discendente (Z-A, dicembre-gennaio)

Questo riquadro è piuttosto intuitivo e non richiede particolari spiegazioni. Vediamo piuttosto qualche esempio pratico di Viste reali.

### Alcuni semplici esempi di Viste

In questo paragrafo immaginiamo di aver creato una sezione “catalogo prodotti” e di voler preparare un indice, per la home page, usando le Viste di Sar-At. Presentiamo qui tre casi molto frequenti: si veda oltre il paragrafo “le viste: uso avanzato” per altri esempi.

Per creare un elenco alfabetico di tutti i prodotti

*La vista non è filtrata e mostra tutti i record della sezione “prodotti”. Verranno presentate contemporaneamente informazioni da non più di 0 record. Mostra i record ordinati per title. Presenta i risultati in ordine ascendente A-Z*

La opzione *verranno presentate contemporaneamente informazioni da...* in questo caso va lasciata in bianco oppure (è lo stesso) va scritto il numero zero, che Sar-At prende nel senso di “nessun limite”.

Per creare un elenco di tutti i prodotti per data di presentazione

*La vista non è filtrata e mostra tutti i record della sezione “prodotti”. Verranno presentate contemporaneamente informazioni da non più di 0 record. Mostra i record ordinati per creation. Presenta i risultati in ordine discendente dic-gen*

Un caso analogo al precedente, ma qui usiamo l'ordinamento discendente (da dicembre a gennaio, cioè i più recenti per primi). Ordinando per data di creazione mostriamo per primi i record (le pagine) inserite più di recente. Se invece ordinassimo per data di aggiornamento allora apparirebbero per primi i record (le pagine) che abbiamo ritoccato poco tempo fa.

**Per creare un elenco dei cinque prodotti più nuovi**

*La vista non è filtrata e mostra tutti i record della sezione “prodotti”. Verranno presentate contemporaneamente informazioni da non più di 5 record. Mostra i record ordinati per creation. Presenta i risultati in ordine discendente dic-gen*

Un caso identico al precedente, ma qui mettiamo all'opera l'opzione “verranno presentate contemporaneamente informazioni da...”

### Viste complete

Sar-At offre poi una seconda modalità per creare le viste, molto più potente della modalità semplificata che abbiamo usato sin qui. È la *modalità completa*. Potete sempre trasformare una Vista creata in modalità semplificata in una Vista completa e continuare ad editarla, mentre il contrario non è possibile.

Per creare una nuova Vista in modalità completa, o per convertire una Vista semplificata a questa modalità, selezionate *Completa* dal selettore a tre stati in alto.

## Creazione nuova Vista

Genera una Vista sulle informazioni contenute in bibliografia

in modalità semplificata completa per esperti

Rispetto a quanto abbiamo visto in precedenza la a cambiare “selezione e filtro” potete ora inserire una lunga serie di condizioni. Per esempio, potete selezionare all'interno del vostro catalogo di libri tutti quelli ancora in catalogo, dal prezzo superiore a dieci euro, scritti in italiano e il cui editore è Apogeo.

**Selezione e filtro** ?

---

**Verranno selezionati nella tabella *libri* solo i record (per ora, 7) per cui:**

- 'visible' = ✓

- AND 'prezzo' > «10»

- AND 'language' = 🇮🇹 italiano

AND editore è uguale a masterId ? ✓ ∅

OR Apogeo

Usare il pulsante + per aggiungere una condizione. Quando l'avete aggiunta, usate il pulsante ✓ per confermarla e aggiungerla alla lista delle condizioni in uso, oppure il pulsante ∅ se ci avete ripensato e preferite non aggiungerla. Se avete già aggiunto una condizione all'elenco e volete eliminarla, cliccate il pulsante - alla sua sinistra.

### Condizioni AND

Le condizioni vengono normalmente concatenate con la modalità chiamata AND, ovvero un record deve rispettare tutte le condizioni per essere selezionato. Per esempio, se io compilo una Vista come nella figura sopra, allora verranno selezionati e mostrati sul vostro sito solamente i libri di Mondadori sopra i dieci euro. Un libro Mondadori da 9 euro e un libro Einaudi da 15 euro non appariranno.

### Condizioni OR

Cliccando il pulsante radio “OR” mentre digito una condizione la aggiungerò come alternativa alle condizioni precedenti. Per esempio potrò selezionare una attrice bionda *oppure* rossa di capelli.

### Priorità degli operatori

AND viene eseguito prima di OR. Per esempio, una Vista che dice:

Verranno selezionati dalla tabella cantanti solo i record per cui

Timbro è Tenore

AND Nazione è Italia

OR Timbro è Soprano

AND Nazione è Spagna

Selezionerà sia i tenori italiani che le soprano spagnole, ma non selezionerà i tenori spagnoli e le soprano italiane. Se l'ordine degli operatori da voi desiderato è differente, convertite la Vista in modalità per esperti e aggiungete le parentesi. Per esempio,

Timbro è Tenore

OR Timbro è Soprano

AND Nazione è Spagna

Trova tutte le soprano spagnole e anche tutti i tenori di qualsiasi nazionalità. Se vi interessa trovare tenori e soprano spagnoli, convertite la Vista in modalità per esperti e vedrete apparire

Timbro = 'Tenore' OR Timbro='Soprano' AND nazione='Spagna'

Modificate così;

(Timbro = 'Tenore' OR Timbro='Soprano') AND nazione='Spagna'

### Viste per esperti

Sar-At consente la creazione di Viste ad hoc in modalità avanzata. In altre parole, un programmatore che conosca il linguaggio standard per le basi dati, SQL, può scrivere a mano la istruzione (o “query”) che Sar-At eseguirà per selezionare i record oggetto della Vista. Per attivare la modalità avanzata, nella schermata delle Viste si seleziona l'opzione “avanzata” nel menu a scomparsa.

**Questa Vista è stata creata manualmente e quindi ogni modifica va effettuata a mano da un programmatore esperto in SQL. Se si desidera rimpiazzarla con una Vista standard è necessario cancellarla e ricrearla da zero.**

Nome ed etichetta ?	*Campi obbligatori
<b>Nome della Vista (un semplice promemoria)</b>	
<input type="text" value="akko_quantefaq"/>	
<b>Etichetta della Vista (da usare nelle pagine mastro)</b>	
<input type="text" value="akko_quantefaq"/>	
<b>Query ?</b>	
<b>Testa della query</b>	
<input type="text" value="SELECT count(*) AS quantefaq FROM faq WHERE visible='Y'"/>	

Sar-At non consente editing “al vivo” delle viste avanzate. In altre parole, non apparirà la matitina che consente di editare il contenuto dei campi direttamente all’interno della pagina web.

Sar-At effettua una verifica sommaria della sintassi di una Vista avanzata e non vi permette di registrare una Vista non funzionale.

### Viste per esperti e variabili predefinite

Sar-At definisce alcune variabili di ambiente in linguaggio SQL che sono utilizzabili da tutte le viste ad hoc. Sono documentate nella tabella sottostante. Gli esempi che faremo da qui in poi suppongono che la Vista venga invocata all’interno di una pagina richiamata come

`showPage.php?template=turismo&id=101&masterPage=mastro.html&search=Madrid&anno=1980`

immaginiamo anche che il record 101 della tabella esempio corrisponda a una pagina di lingua spagnola, intitolata “España”. La lingua spagnola, codifica ISO, in Sar-At è codificata con il numero 1004 (l’elenco completo dei codici di lingua si trova nel manuale *Mantenere il sito*).

Variabile	Valore
<b>@id</b>	Lo id del record corrente. Nel caso dell’esempio la variabile varrà 101. Per i <i>form</i> , quindi quando il programma corrente è showForm, non ha alcun valore.
<b>@sarelLanguage</b>	Il codice numerico della lingua della pagina. È un numero intero che nel caso dell’esempio vale 1004. Per i moduli, quindi quando il programma corrente è showForm, è il codice dell’ultima lingua usata nell’ultima pagina vista con showPage in precedenza.
<b>@saratLanguage</b>	Il codice ISO della lingua della pagina, composto da due lettere. Nel caso dell’esempio vale “es”. Per i moduli, quindi quando il programma corrente è showForm, è il codice dell’ultima lingua usata nell’ultima pagina vista con showPage in precedenza
<b>@masterPage</b>	Il nome della pagina mastro chiamata a vestire di grafica questi dati. Nel caso dell’esempio, vale “mastro.html”
<b>@template</b>	Il nome della sezione corrente. Nel caso dell’esempio vale “turismo”
<b>@saratPremium</b>	La pagina è riservata ai visitatori registrati o di libero accesso anche ai visitatori anonimi? Nel secondo caso questa variabile, che è sempre un numero intero, vale zero. Per i moduli vale normalmente zero
<b>@visible</b>	Un singolo carattere, che vale Y o N. Indica se la pagina è visibile (Y) o nascosta e irraggiungibile (N) sul web
<b>@sarel_userlogged</b>	Se il visitatore ha inserito una username e una password, vale 1.
<b>@sarel_username</b>	Se il visitatore ha inserito una username e una password, vale come la sua username. Altrimenti è una stringa univoca di circa trenta caratteri che lo identifica a livello cookie.
<b>@sarat_adminlogged</b>	Se la pagina viene visualizzata da un amministratore Sar-At che ha inserito username e password, vale 1.
<b>@sarat_adminname</b>	Se la pagina viene visualizzata da un amministratore Sar-At che ha inserito username e password, vale come la sua username. Altrimenti è vuota
<b>@saratMobile</b>	Se la pagina appare su dispositivo a piccolo schermo vale 1, altrimenti vale 0

Variabili provenienti dal record corrente

Tutti i campi del record corrente – dia quelli che avere definito voi sia quelli incorporati - sono variabili predefinite. Per esempio, immaginate di aver realizzato il sito dei cantanti lirici, come mostrato a pagina 7 del manuale. Immaginate anche di aver richiamato nel browser la scheda del tenore Amilcare Pallazzi, digitando l'indirizzo

```
showPage.php?template=cantanti&id=1
```

In questo caso, nelle vostre viste SQL risultano automaticamente definite e inizializzate le seguenti variabili: *@timbro* con valore Tenore, *@nazione* con valore Italia, *@masterId* con valore 1. Eccetera.

Vi è dunque molto semplice inserire nella pagina “Amilcare Pallazzi” un elenco “altri tenori” o un elenco “altri cantanti lirici provenienti dall'Italia”. Per trovare gli altri cantanti italiani infatti vi basta una Vista come questa:

```
SELECT * FROM cantanti WHERE timbro=@timbro
```

E per gli altri italiani:

```
SELECT * FROM cantanti WHERE nazione=@nazione
```

L'uso delle variabili predefinite è compatibile con la cache che Sar-At gestisce per le pagine mastro classiche.

### Variabili provenienti dalla URL

Tutti i parametri della URL corrente sono variabili predefinite. Nel caso dell'esempio risultano automaticamente definite e inizializzate le seguenti variabili: *@anno* con valore 1980, *@search* con valore Madrid.

Poiché aggiungere parametri alla URL è facilissimo (e si può anche spedirne da un modulo HTML), le variabili passate via URL vi permettono di costruire con grande semplicità motori di ricerca arbitrariamente complicati. Per esempio, dentro la pagina indicata potreste mostrare un elenco di tutti i cantanti nati nel 1980.

```
SELECT * FROM cantanti WHERE YEAR(data_nascita)=@anno
```

Attenzione! L'uso delle variabili URL automaticamente e inevitabilmente rende impossibile il meccanismo della cache di Sar-At per pagine mastro classiche. La produzione della pagina dunque risulta dieci volte più lenta di quanto accadrebbe per una pagina in cui la cache è attiva.

### Viste per esperti, pagine mastro classiche e cache

Una pagina che contiene una vista standard e viene usata in una pagina mastro classica può sempre finire nella cache di Sar-At, e quindi venire servita ai visitatori del sito in tutta velocità. Le viste ad hoc, cioè quelle scritte a mano in linguaggio SQL, potrebbero risultare incompatibili con la cache delle pagine mastro classiche. Facciamo un esempio per capire dove sta il problema. Avete definito una sezione “banner” e l'avete riempita di inserzioni pubblicitarie che debbono apparire a rotazione in tutte le pagine del vostro sito. Avete poi scritto una Vista ad hoc che ha lo scopo di pescare un banner a caso tra tutti quelli disponibili. Se Sar-At mettesse nella sua cache le vostre pagine web, succedrebbe che quando il primo visitatore naviga su una pagina allora la vostra Vista viene eseguita e pesca casualmente il banner dell'inserzionista Pinco Pallino. La pagina così prodotta viene poi salvata nella memoria cache e inviata in velocità a tutti i visitatori susseguenti. Questo comportamento però a voi non sta bene, perché voi desiderate che tutti gli inserzionisti vengano ugualmente rappresentati e non appaia il solo Pinco Pallino. Per questo motivo, quando una pagina mastro classica contiene una vista ad hoc, Sar-At prudenzialmente la esclude dal sistema cache e quindi la rigenera ogni volta che un visitatore ci capita sopra.

Per una descrizione della cache di Sar-At si veda il manuale *Governare il sito*, capitolo primo, sotto a Prestazioni.

Quando si definisce una vista ad hoc, Sar-At permette al programmatore SQL di specificare se la Vista va ritenuta compatibile con la cache o meno. Vanno definite non compatibili tutte le viste che non estraggono sistematicamente alcuni record identificati per numero caratteristico (id). In sostanza, quando Sar-At genera una pagina web si segna quali record concorrono a formarla. Per esempio, in un sito di ricette, la pagina con la ricetta della torta di mele potrebbe dipendere solo dal record numero 1 della sezione ricette (che contiene appunto le istruzioni per cucinare una torta di mele). Se noi

impaginiamo su un lato anche un indice di tutte le ricette dei dolci, Sar-At si appunta tutti gli id di tutte le ricette: in questo modo se noi cambiamo il nome del “tiramisù” in “tiramisu” (senza l’accento) Sar-At può riconoscere la necessità di rigenerare la pagina della torta di mele.

Le Viste ad hoc possono risultare non compatibili con la cache quando sfuggono a questa logica. Per esempio: è incompatibile una vista che non seleziona il campo id (SELECT campo1, campo2 FROM sezione non va bene); è incompatibile una vista che seleziona informazioni da due o più sezioni contemporaneamente (JOIN); è incompatibile una vista ad hoc che seleziona sulla base di informazioni che cambiano in continuazione, come l’ora corrente (SELECT NOW() AS adesso).

### Il codice HTML nella pagina mastro

Come dicevamo, Sar-At registra le nostre impostazioni nella sua base dati e le esegue tutte le volte che trova l’etichetta della Vista in una pagina mastro. La definizione della Vista decide quali e quanti record appariranno nel contesto della pagina mastro. Ma come decidiamo cosa deve apparire in grassetto e cosa in corsivo? E come stabiliamo se, di ogni record, deve apparire solo il titolo oppure tutte le informazioni (i campi) che abbiamo a disposizione in base dati? Tutto questo avviene durante la compilazione della pagina mastro, ma ci sono differenze importanti a seconda che quest’ultima si stia scritta in linguaggio Twig o in modalità classica.

Per i nostri esempi da qui in poi immagineremo di avere a disposizione una tabella chiamata “prodotti” che contiene tre oggetti:

prodotti			
Titolo	Price	Lingua	Distruggi
<b>Biscotti al cioccolato</b>	10	italiano	
<b>Torta al cioccolato</b>	25	italiano	
<b>Yogurt al cioccolato</b>	1.75	italiano	

### Viste nelle pagine mastro classiche

Il richiamo di una vista in una pagina mastro classica si fa usando la sintassi {BEGIN etichetta}...{END etichetta}. Tutto ciò che si trova tra BEGIN ed END viene ripetuto una volta per ogni record trovato.

#### Elenco semplice

Per avere un elenco di tutti i prodotti nella mia pagina home io scrivo:

```
{BEGIN elenco_prodotti}
{title}<br />
{END elenco_prodotti}
```

Il risultato HTML (che viene inviato al visitatore) sarà:

```
Biscotti al cioccolato<br />
Torta al cioccolato<br />
Yogurt al cioccolato<br />
```

Il risultato a video sarà semplicemente:

```
Biscotti al cioccolato
Torta al cioccolato
Yogurt al cioccolato
```

#### Link alle pagine interne

Immaginiamo che di ciascun prodotto esiste una ampia descrizione. Vogliamo allora che, quando il visitatore clicca il nome del prodotto, venga mandato alla scheda del prodotto. Se abbiamo impostato correttamente le cose, il nostro sito avrà una “cartella magica” SEO usata per le URL amichevole (si veda la spiegazione nel capitolo precedente) e la scheda del prodotto “Yogurt al cioccolato” avrà un indirizzo registrato nel campo automatico saratUrl quindi sarà accessibile a un indirizzo del tipo [www.nomesito.it/nome-cartella-magica/yogurt-cioccolato-1234567.html](http://www.nomesito.it/nome-cartella-magica/yogurt-cioccolato-1234567.html)

Per avere un elenco linkato nella mia pagina home io scrivo:

```
{BEGIN elenco_prodotti}
    <a href='/nome-cartella-magica/{saratUrl}'>{title}</a>
{END elenco_prodotti}
```

Il risultato HTML (che viene inviato al visitatore) sarà simile a questo:

```
<a href='/nome-cartella-magica/yogurt-cioccolato-1234567'>
    Yogurt al cioccolato
</a>
<a href='/nome-cartella-magica/yogurt-cioccolato-1234568'>
    Biscotti al cioccolato
</a> <a href='/nome-cartella-magica/yogurt-cioccolato-1234569'>
    Torta al cioccolato
</a>
```

Se le URL amichevoli non sono state accese sul vostro sito bisogna ricorrere all'indirizzo diretto showPage, quindi useremo

```
{BEGIN elenco_prodotti}
    <a href='showPage.php?template=prodotti&id={id}'>{title}</a>
{END elenco_prodotti}
```

#### Elenco numerato programmaticamente

Sar-At offre all'interno di tutte le Viste create nelle pagine mastro classiche il tag saratviews\_serial, che possiamo usare per creare un elenco numerato senza fare affidamento all'HTML (per esempio, quando bisogna produrre uno XML, oppure quando vogliamo creare i numeri in grafica, oppure dentro le viste navigabili).

Il seguente esempio mostra una vista navigabile, usata per presentare i prodotti a dieci per volta, ciascuna con il numero alla sinistra. Quando il visitatore clicca per vedere la seconda sottopagina (prodotti da 11 a 20), la numerazione è corretta

```
{NAVIGATE dieci_prodotti_alla_volta}
    <div class="numerato">
        {saratviews_serial}
    </div>
    <div class="collegamento">
        <a href="/nome-cartella-magica/{saratUrl}">{title}</a>
    </div>
{END dieci_prodotti_alla_volta}
```

#### In tabella per righe a colore alternato

Sar-At riconosce la necessità di creare tabelle visivamente aggraziate e leggibili usando il meccanismo delle Viste. Per questo motivo Sar-At definisce automaticamente in tutte le pagine mastro classiche uno pseudo-campo chiamato sarelvviews\_row e gli assegna alternativamente il valore "odd" e "even" (rispettivamente nelle righe dispari e pari). In combinazione con un semplice foglio stile, questo permette di ottenere righe a sfondo alternato.

```
<style type="text/css">
    td.odd { background-color: #CCFFFF; }
    td.even { background-color: #0099FF; }
</style>
<table border='1'>
{BEGIN elenco_prodotti}
    <tr>
        <td class='{sarelvviews_row}'>
            {title}
        </td>
        <td class='{sarelvviews_row}'>
            {prezzo}
    </tr>
```

```

        </td>
    </tr>
{END elenco_prodotti}
</table>

```

Il risultato HTML (che viene inviato al browser) sarà:

```

<table border='1'>
  <tr>
    <td class='odd'>Biscotti al cioccolato</td>
    <td class='odd'>10</td>
  </tr>
  <tr>
    <td class='even'>Torta al cioccolato</td>
    <td class='even'>25</td>
  </tr>
  <tr>
    <td class='odd'>Yogurt al cioccolato</td>
    <td class='odd'>1.75</td>
  </tr>
</table>

```

Il risultato a video sarà:

Biscotti al cioccolato	10
Torta al cioccolato	25
Yogurt al cioccolato	1.75

### Viste nelle pagine mastro Twig

Per Twig ogni Vista inclusa nella pagina diventa un vettore, compilato da Sar-At con tutti i record trovati nella base dati eseguendo la Vista.

#### Elenco semplice

Per avere un elenco di tutti i prodotti nella mia pagina home io scrivo:

```

{% for unProdotto in elenco_prodotti %}
  {{ unProdotto.title }}<br />
{% endfor %}

```

Il risultato HTML (che viene inviato al visitatore) sarà:

```

Biscotti al cioccolato<br />
Torta al cioccolato<br />
Yogurt al cioccolato<br />

```

Il risultato a video sarà semplicemente:

```

Biscotti al cioccolato
Torta al cioccolato
Yogurt al cioccolato

```

#### Link alle pagine interne

Immaginiamo che di ciascun prodotto esiste una ampia descrizione. Vogliamo allora che, quando il visitatore clicca il nome del prodotto, venga mandato alla scheda del prodotto. Se abbiamo impostato correttamente le cose, il nostro sito avrà una “cartella magica” SEO usata per le URL amichevole (si veda la spiegazione nel capitolo precedente) e la scheda del prodotto “Yogurt al cioccolato” avrà un indirizzo registrato nel campo automatico saratUrl quindi sarà accessibile a un indirizzo del tipo [www.nomesito.it/nome-cartella-magica/yogurt-cioccolato-1234567.html](http://www.nomesito.it/nome-cartella-magica/yogurt-cioccolato-1234567.html)

Per avere un elenco linkato nella mia pagina home io scrivo:

```

{% for unProdotto in elenco_prodotti %}
  <a href='/nome-cartella-magica/{{ unProdotto.saratUrl }}'>

```



```

        {{ unProdotto.title }}
    </a>
{% endfor %}

```

Il risultato HTML (che viene inviato al visitatore) sarà simile a questo:

```

<a href='/nome-cartella-magica/yogurt-cioccolato-1234567'>
    Yogurt al cioccolato
</a>
<a href='/nome-cartella-magica/yogurt-cioccolato-1234568'>
    Biscotti al cioccolato
</a> <a href='/nome-cartella-magica/yogurt-cioccolato-1234569'>
    Torta al cioccolato
</a>

```

Se le URL amichevoli non sono state accese sul vostro sito bisogna ricorrere all'indirizzo diretto showTwig, quindi useremo

```

{% for unProdotto in elenco_prodotti %}
    <a href='showTwig.php?template=prodotti&id={{ unProdotto.id }}'>
        {{ unProdotto.title }}
    </a>
{% endfor %}

```

#### Elenco numerato programmaticamente

Il tag saratviews\_serial presentato in precedenza non esiste sotto Twig, né ce n'è alcun bisogno. Il numero di prodotti trovati dalla vista del nostro esempio è facilmente disponibile come `{{ elenco_prodotti|length }}`.

Se vogliamo presentare solo i prodotti da 11 a 20 scriveremo:

```

{% for numero, unProdotto in elenco_prodotti %}
    {%if numero > 10 and numero < 21 %}
        {{ numero }}. {{ unProdotto.title }}<br />
    {% endif %}
{% endfor %}

```

#### In tabella per righe a colore alternato

Il tag sareviews\_row presentato in precedenza non esiste sotto Twig, né ce n'è alcun bisogno. Per stilizzare una tabella con righe di colore alternato possiamo usare una programmazione di questo tipo:

```

<style>
    td.odd { background-color: #CCFFFF; }
    td.even { background-color: #0099FF; }
</style>

<table>
{% for numeroRiga, unProdotto in tutti_prodotti %}
    <tr>
        {%if numeroRiga is odd %}
            <td class='odd'>
        {% else %}
            <td class='even'>
        {% endif %}
        {{ unProdotto.title }}
    </td>
    </tr>
{% endfor %}
</table>

```

## Viste navigabili

Nella pagina mastro classiche, Sar-At offre l'uso delle viste navigabili, cioè Viste che non mostrano tutti i loro risultati al visitatore in una pagina sola ma invece li dividono in sottopagine, stile risultati della ricerca in Google. Anche in questo caso, Twig non dispone della funzionalità perché non ne ha alcun bisogno.

Per esempio, se il vostro database contiene 500 squadre di calcio e voi desiderate mostrare dieci pagine web, ciascuna con una tabella che riporta nome, città e allenatore per un gruppo di cinquanta squadre, allora vi serve una Vista navigabile. Voi disegnerete una singola pagina mastro che verrà usata, nel caso del nostro esempio, dieci volte.

Come autore del sito potete decidere quanti risultati appariranno in ogni sottopagina, come impaginare i risultati, e costruire strumenti di navigazione per andare avanti e indietro di una pagina, saltare a una sottopagine scelta per numero, mostrare quante sottopagine sono disponibili, tornare al primo o saltare all'ultimo risultato, mostrare uno sfogliatore sulle sottopagine.

L'uso delle viste navigabile è particolarmente semplice. Al posto dell'ormai consueto {BEGIN nomevista}...{END nomevista} adatterete {NAVIGATE nomevista}...{END nomevista}

Qualche ragionamento in più serve per capire come vengano realizzati gli strumenti di navigazione

Il pulsante, o link, "passa alla sottopagina successiva" si troverà al di fuori del blocco {NAVIGATE nomevista}...{END nomevista}. Se così non fosse, apparirebbe molte volte nella pagina. Proseguendo col nostro esempio di 500 squadre di calcio mostrate a gruppi di cinquanta, se il link alla sottopagina 2 si trovasse dentro al blocco {NAVIGATE nomevista}...{END nomevista} di sottopagina 1, allora lo vedremmo cinquanta volte. Quindi questo link starà al di fuori del blocco.

Nulla ci impedisce di usare due viste navigabili nella stessa pagina mastro. Per questo motivo, l'istruzione (tag) Sar-At che genera il link "vai alla sottopagina successiva" dovrà avere due nomi diversi. Per esempio, nella pagina che mostra 50 squadre di calcio, potremmo mostrare anche 25 squadre di basket. Facendo clic su "avanti alla sottopagina successiva del calcio" vedremo le squadre di calcio da 51 a 100 e le squadre di basket da 1 a 25. Facendo invece clic su "avanti alla sottopagina successiva del basket" vedremo le squadre di calcio da 1 a 50 e le squadre di basket da 26 a 50.

### Strumenti di navigazione nelle Viste

Per ogni Vista navigabile presente in una vostra pagina mastro, Sar-At automaticamente definisce sette tag aggiuntivi che possono venire usati ovunque nella pagina (ma che non ha molto senso usare dentro al blocco {NAVIGATE nomevista}...{END nomevista}), come dicevamo). Non è necessario usare tutti i tag: potete sfruttare solo quelli che vi paiono più appropriati.

Il loro nome è composto con un prefisso, mostrato nella tabella sottostante, e un suffisso che è sempre il nome della Vista navigabile di riferimento.

Tag	Scopo	Valore
<i>sar-at-qnext-nomevista</i>	Avanti alla prossima sottopagina	Un link
<i>sar-at-qprev-nomevista</i>	Indietro alla precedente sottopagina	Un link
<i>sar-at-qfirst-nomevista</i>	Torna alla prima sottopagina	Un link
<i>sar-at-qlast-nomevista</i>	Vai all'ultima sottopagina	Un link
<i>sar-at-qcount-nomevista</i>	Numero di sottopagine esistenti	Un numero intero
<i>sar-at-qhere-nomevista</i>	Numero della sottopagine corrente	Un numero intero
<i>sar-at-qnav-nomevista</i>	Navigatore, stile Google, sulle sottopagine disponibili	Serie di link
<i>sar-at-qnavnum-nomevista</i>	Numero di record da restituire per ogni pagina	Un numero intero
<i>sar-at-qfound-nomevista</i>	Numero di record selezionati in tutto	Un numero intero
<i>sar-at-qlinknext-nomevista</i>	Soddisfare Google	Un <link> HTML

Tag	Scopo	Valore
<code>sar-at-qlinkprev-<i>nomevista</i></code>	Soddisfare Google	Un <code>&lt;link&gt;</code> HTML
<code>sar-at-qsd-<i>nomevista</i></code>	Ordinare i risultati in modo ascendente o discendente	ASC o DESC

Vediamo un esempio pratico. Nel nostro ipotetico sito sportivo che tratta di calcio e mostra 500 squadre a gruppi di 50, il visitatore sta guardando la quarta sottopagina (squadre da 151 a 200) in ordine alfabetico.

In questo caso avremo:

`sar-at-qnext-nomevista`: link alla quinta sottopagina (squadre da 201 a 250)

`sar-at-qprev-nomevista`: link alla terza sottopagina (squadre da 101 a 150)

`sar-at-qfirst-nomevista`: link alla prima sottopagina (squadre da 1 a 50)

`sar-at-qlast-nomevista`: link alla ultima sottopagina (squadre da 451 a 500)

`sar-at-qcount-nomevista`: numero di sottopagine, quindi 10

`sar-at-qnavnum-nomevista`: numero di righe per pagina, quindi 50

`sar-at-qhere-nomevista`: numero di questa sottopagina, quindi 4

`sar-at-qnav-nomevista`: i numeri da 1 a 10, collegati alle relative sottopagine (tranne il numero 4 che non è collegato ad alcunché visto che stiamo già guardando la sottopagina 4).

`sar-at-qsd-nomevista`: vale ASC perché l'ordinamento è alfabetico ascendente.

### Paginazione

Come sappiamo, una Vista viene definita sulla base di un filtro (per esempio: mostra solo i prodotti dal costo unitario sopra i 5 euro, oppure mostra solo le voci del blog più recenti di un mese) e di un numero di risultati da mostrare. Le cose non cambiano nel caso delle Viste navigabili. Semplicemente, il secondo numero rappresenta il numero di risultati per ogni sottopagina.

Nel caso delle Viste realizzate in modalità "per esperti", appare un box compilabile con l'aspetto che mostriamo nella figura sottostante.

Nel caso di una Vista avanzata usata in modo paginabile con il costrutto NAVIGATE, dunque, abbiamo che alla prima esecuzione la Vista viene eseguita esattamente come l'avete digitata. Quando il visitatore naviga, per esempio cliccando il link `sar-at-qnext` per andare alla sottopagina successiva, Sar-At esegue normalmente la testa della vostra query e vi concatena una cosa che genera lui stesso. Facciamo un esempio osservando la immagine sottostante.

Al primo giro, la query SQL eseguita sarà esattamente quella digitata:

```
SELECT *, YEAR(date) AS anno FROM inlinea WHERE visible='Y' ORDER BY date DESC LIMIT 3
```

Quando il visitatore clicca un link `sar-at-qnext` per andare alla sottopagina successiva, Sar-At esamina la configurazione del box "Navigazione". Trova l'indicazione di navigare per date discendenti e genera

dunque la clausola ORDER BY date DESC LIMIT 3,3. Produce la query finale concatenando la testa così come l'ha trovata con la coda generata. Abbiamo dunque:

```
SELECT *, YEAR(date) AS anno FROM inlinea WHERE visible='Y' ORDER BY date DESC
LIMIT 3,3
```

### Personalizzazione grafica dei link di navigazione

I link di navigazione “avanti di una pagina”, “indietro di una pagina”, “torna all’inizio” e “vai alla fine” generati da Sar-At sono normalmente rappresentati con semplici caratteri, e cioè rispettivamente:

Tag	Testo
<i>sar-at-qnext-nomevista</i> Avanti alla prossima sottopagina	>
<i>sar-at-qprev-nomevista</i> Indietro alla precedente sottopagina	<
<i>sar-at-qfirst-nomevista</i> Torna alla prima sottopagina	<<
<i>sar-at-qlast-nomevista</i> Vai all'ultima sottopagina	>>

Potreste desiderare di usare immagine grafiche (icone) o testi localizzati nella lingua del vostro sito. Nel primo caso, è sufficiente usare un CSS, così:

```
<style type='text/css' title='text/css' media='screen'>
#sar-at-qlast-nomevista span { display: none; }
#sar-at-qlast-nomevista
{
    display: inline-block;
    background: url(navigate/2rightarrow.png) 0 0 no-repeat;
    width: 16px;
    height: 16px;
}
#sar-at-qnext-nomevista span { display: none; }
#sar-at-qnext-nomevista
{
    display: inline-block;
    background: url(navigate/1rightarrow.png) 0 0 no-repeat;
    width: 16px;
    height: 16px;
}
#sar-at-qprev-nomevista span { display: none; }
#sar-at-qprev-nomevista
{
    display: inline-block;
    background: url(navigate/1leftarrow.png) 0 0 no-repeat;
    width: 16px;
    height: 16px;
}
#sar-at-qfirst-nomevista span { display: none; }
#sar-at-qfirst-nomevista
{
    display: inline-block;
    background: url(navigate/2leftarrow.png) 0 0 no-repeat;
    width: 16px;
    height: 16px;
}
</style>
```

Per sostituire i testi con altri, usate un sistema JavaScript come jQuery. La sintassi sarà tipo questa:  
`$('#sar-at-qfirst-nomevista span').html('quel che vuoi tu');`

## Il motore di ricerca

Il meccanismo delle Viste si può utilizzare anche per creare semplici motori di ricerca interni al sito - usabili indifferentemente da pagine mastro Twig o classiche. (Per un motore di ricerca avanzato si veda “plug in motore di ricerca incorporato” nell’ultimo capitolo del manuale Sar-At).

Le viste utilizzate come motore di ricerca sono quelle in cui specifichiamo “mostra solo i record in cui (regola di ricerca) — è uguale a — valore scelto dal visitatore”.

Facciamo un esempio pratico. Creiamo una nuova Vista. Le diamo un nome a piacere e per etichetta scegliamo “motore\_ricerca”. Impostiamo la Vista perché selezioni i soli record il cui titolo contiene un valore scelto dal visitatore, che la pagina HTML passerà nel parametro *search*.

Perché il motore di ricerca abbia senso dovremo preparare due pagine HTML. La prima pagina (form di ricerca) permette al visitatore di scrivere il testo che vuole cercare e può avere un aspetto di questo tipo:

La seconda pagina mostra i risultati della ricerca, impaginati a nostro piacere, usando la Vista.

### Il form di ricerca

Il form di ricerca è un comune form (o “modulo”) HTML che chiama Sar-At a eseguire la ricerca. La modalità di invio è indifferentemente POST o GET (Sar-At supporta entrambe). Il modulo deve necessariamente passare a Sar-At le seguenti informazioni :

Nome	Obbligatorio?	Usi e note
<b>template</b>	Sì	Come sempre quando chiamiamo Sar-At dobbiamo specificare la sezione che fornisce il contesto. Se nella pagina “risultati della ricerca” mostriamo esclusivamente i risultati della ricerca, questo parametro può avere un valore qualsiasi, ma il valore deve essere fornito e deve essere valido (il nome di una sezione esistente). Se nella pagina “risultati della ricerca” mostriamo anche altro (per esempio, la scheda del nostro prodotto di punta, il numero 250, “crema al cioccolato”) allora passeremo la sezione che ci interessa (per esempio, “prodotti”).

Nome	Obbligatorio?	Usi e note
<b>id</b>	No	Se nella pagina “risultati della ricerca” mostriamo esclusivamente i risultati della ricerca, questo parametro può venire omissis. Se nella pagina “risultati della ricerca” mostriamo anche altro (per esempio, la scheda del prodotto numero 250, “crema al cioccolato”) allora passeremo il numero caratteristico che ci interessa (per esempio, 250).
<b>masterPage</b>	Sì	Nome della pagina “risultati della ricerca”.
<b>search</b>	Sì	È il valore da cercare

Vediamo un esempio di form di ricerca in HTML:

```
<form action="showPage.php" method="post">
  <input type="hidden" name="template" value="prodotti" />
  <input type="hidden" name="masterPage" value="trova.html" />
  <input type="text" name="search" size="30" />
</form>
```

### Pagina “risultati della ricerca”

La pagina con il risultato si impagina esattamente come una qualsiasi pagina mastro che contiene una Vista. (Vedere il paragrafo “Viste e pagine mastro: alcuni semplici esempi” sopra).

Quando create una nuova Vista, Sar-At genera automaticamente il codice di un modulo analogo a quello appena visto e anche il codice HTML per una pagina “risultati della ricerca” che utilizza quella vista. Potete copiare quel codice e reimpaginarlo come più vi aggrada.

### Quando nulla viene trovato

In una pagina “risultati della ricerca” molto semplice, come quella proposta qui sopra, quando il visitatore cerca un testo che non esiste, allora appare una pagina bianca. Una soluzione più elegante si ottiene utilizzando il meccanismo “se...allora..altrimenti”. Ecco un esempio scritto per una pagina mastro classica:

```
{IF motore_ricerca}
  <p>Elenco dei prodotti il cui nome contiene la parola <em>{search}</em>:</p>
  {BEGIN motore_ricerca}
    <p>{title}</p>
  {END motore_ricerca}
{ELSE motore_ricerca}
  <p>Spiacente, nessun prodotto contiene la parola <em>{search}</em> nel nome.</p>
{ENDIF motore_ricerca}
```

Ecco l'equivalente Twig:

```
{% if motore_ricerca %}
  {% for unProdotto in motore_ricerca %}
    <p>{{ unProdotto.title }}</p>
  {% endfor %}
{% else %}
  <p>Spiacente, nessun prodotto contiene la parola <em>{{ _GET.search }}</em> nel nome.</p>
{% endif %}
```

Se il visitatore cerca “biscotti” allora il risultato HTML sarà:

```
<p>Elenco dei prodotti il cui nome contiene la parola <em>biscotti</em>:</p>
<p>Biscotti al cioccolato</p>
```

Se il visitatore cerca “vaniglia” allora il risultato HTML sarà:

```
<p>Spiacente, nessun prodotto contiene la parola <em>vaniglia</em> nel nome.</p>
```

Una trattazione completa della sintassi se...allora...altrimenti si trova nell'ultimo capitolo di questo manuale, la quale contiene anche la documentazione dell'istruzione LOCATION — spesso utile per gestire le ricerche che non hanno avuto successo.

### Motore di ricerca: altri esempi

Un motore di ricerca può prendere anche altre forme: per esempio, quello di un menu popup. In HTML, un menu è semplicemente una istruzione `<select>` che contiene una serie di `<option>`, una per ogni voce. Possiamo creare una pagina mastro tale per cui Sar-At crea la `<select>` per noi.

Proseguiamo con l'esempio visto nelle pagine precedenti: immaginiamo dunque di aver creato una vista etichettata “elenco\_prodotti” che trova tutti i tre membri della sezione “prodotti” e li elenca in ordine alfabetico. Immaginiamo anche di aver definito una vista “motore\_ricerca”, la quale seleziona i soli record il cui titolo contiene un valore scelto dal visitatore.

Possiamo allora creare il form di ricerca in questo modo. Versione classica:

```
<form action="showPage.php" method="post">
  <select name="id">
    <option value='' disabled selected>
      (Scegli il prodotto che ti interessa)
    </option>
    {BEGIN elenco_prodotti}
    <option value="{id}">
      {title}
    </option>
    {END elenco_prodotti}
  </select>
  <input type="hidden" name="template" value="prodotti">
  <input type="submit" value="Vai">
</form>
```

Versione Twig:

```
<form action="showTwig.php" method="post">
  <select name="id">
    <option value='' disabled selected>
      (Scegli il prodotto che ti interessa)
    </option>
    {% for unProdotto in elenco_prodotti %}
    <option value="{{ unProdotto.id }}">
      {{ unProdotto.title }}
    </option>
    {% endfor %}
  </select>
  <input type="hidden" name="template" value="prodotti">
  <input type="submit" value="Vai">
</form>
```

Ecco il risultato HTML:

```
<form action="showPage.php" method="post">
```

```

<select name="id">
  <option value='' disabled selected>
    (Scegli il prodotto che ti interessa)
  </option>

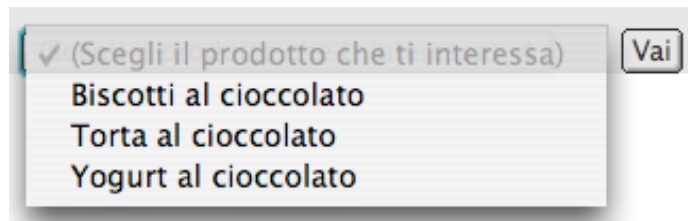
  <option value="103">
    Biscotti al cioccolato
  </option>

  <option value="101">
    Torta al cioccolato
  </option>

  <option value="102">
    Yogurt al cioccolato
  </option>
</select>
<input type="hidden" name="template" value="prodotti">
<input type="submit" value="Vai">
</form>

```

A video apparirà così:



La pagina “risultati della ricerca” non va affatto cambiata rispetto all’esempio visto sopra.

### Viste di tabelle correlate

In alcuni casi può rendersi necessario il collegamento tra loro delle informazioni contenute in due o più sezioni di un sito Sar-At.

Immaginate di voler creare un sito web che presenta marche e modelli di automobili. Avremo una sezione chiamata “marche” che contiene, tra le altre, queste informazioni:

Title	Nazione	Id
Volkswagen	Germania	1
Ford	USA	2
Honda	Giappone	3
Nissan	Giappone	4

Avremo anche una sezione “modelli” che contiene, tra le altre, queste informazioni:

Title	Id	Id_marca
Civic	100	3
Insight	101	3



Title	Id	Id_marca
Lupo	102	1

È evidente che ci sono molte cose da dire sulle case automobilistiche, oltre ai modelli di auto che producono. Possiamo ben immaginare che la pagina sulla Ford conterrà molte informazioni.

È anche ovvio che ci sono molte informazioni da comunicare sui modelli di auto; per esempio, la pagina sulla Lupo sarà ricca di foto, testi, filmati eccetera. Immaginiamo però di voler dedicare una pagina mastro a mostrare le correlazioni tra marche e modelli. Bisognerà, per esempio, mostrare insieme le informazioni fondamentali sulla Fiat e anche l'elenco dei modelli prodotti da Fiat. La pagina mastro "marche-modelli.html" e farà riferimento alle marche. Se la chiamiamo a visualizzare l'id 3 vedremo tutto ciò che si può voler sapere su Honda, compresi i link alle pagine di Civic e Insight.

Per mostrare dentro questa pagina mastro tutti e soli i modelli prodotti dalla Casa selezionata, creiamo una Vista sui modelli in questo modo:

**Filtra, e mostra i soli record in cui**

il valore

della pagina corrente [?](#)

un valore scelto dal visitatore, e la pagina HTML invocherà questa Vista passando il parametro *search* [?](#)

In altre parole: la Vista "modelli\_della\_marca" apparirà all'interno della pagina "marche-modelli.html". Quando Sar-At la incontra andrà a selezionare tutti e soli i modelli per cui il campo "id\_marca" è uguale allo id della pagina corrente.

Ecco il codice HTML che potremo inserire dentro "marche-modelli.html":



Versione classica	Versione Twig
<pre> La casa {title} produce i seguenti modelli: &lt;ul&gt;{BEGIN modelli_della_marca}   &lt;li&gt;     &lt;a href='/cartella_magica/{saratUrl}'&gt;       {title}     &lt;/a&gt;   &lt;/li&gt; {END modelli_della_marca} &lt;/ul&gt; </pre>	<pre> La casa {{ title }} produce i seguenti modelli: &lt;ul&gt;   {% for m in modelli_della_marca %}   &lt;li&gt;     &lt;a href='/cartella_magica/{{ saratUrl }}'&gt;       {{ m.title }}     &lt;/a&gt;   &lt;/li&gt;   {% endfor %} &lt;/ul&gt; </pre>

Le viste di tabelle correlate sono automaticamente utilizzabili per i portali, che descriviamo immediatamente di seguito.

### Viste e portali

Quando avete creato una vista di tabella correlata (come descritto nel paragrafo precedente di questo manuale), potete scegliere di visualizzarne i risultati non solo dentro al sito pubblico ma anche dentro Sar-At stesso. Tornando al nostro esempio di marche e modelli di automobili, è piuttosto comodo che un redattore possa inserire le informazioni su un nuovo modello Ford dalla scheda generale di casa Ford. In altre parole, lavorando dalla sezione "marche" potrà inserire un "modello", e quest'ultimo sarà automaticamente collegato alla sua marca con l'id numerico corretto.

I portali si creano nella linguetta *Struttura* della sezione contenitrice “madre”, sempre e solo dopo aver definito la vista di tabella correlata. È un semplice processo in tre fasi. Nella prima, si assegna al portale un nome di fantasia e si sceglie dal menu a scomparsa il nome della sezione figlia (“modelli” nel nostro esempio).

<b>Nome campo</b>	modelli_prodotti
<b>Note</b>	Tutti i modelli di questa marca
<b>Tipo</b>	Portale... 
<b>Creazione portale - fase 1 di 3 ?</b> Per mostrare valori da una tabella collegata (per esempio, tutti i modelli di una marca) scegli quale tabella collegare a questa: 	

Fase due: a questo punto Sar-At mostra tutte le viste che correlano la sezione madre con la sezione figlia e ci permette di sceglierne una.

Nella terza ed ultima fase, Sar-At ci chiede quali funzionalità debbono essere disponibili attraverso il portale. Se non spuntiamo niente, il redattore potrà vedere tutti i modelli di Fiat dentro alla scheda della casa automobilistica, ma senza poterli manipolare. Se spuntiamo la casella “Consenti aggiunta record”, apparirà un pulsante che consente di aggiungere una nuova pagina alla sezione figlia (“modelli” nel nostro esempio). Al redattore apparirà in pop-up una nuova finestra Sar-At per effettuare l’inserimento.

Se spuntiamo la casella “Consenti cancellazione”, il redattore potrà cancellare i record correlati dalla tabella figlia (“modelli” nel nostro esempio) senza lasciare la scheda della tabella madre (la casa produttrice nel nostro esempio).

Il permesso di cancellare record attraverso il portale sovrasta il permesso di intervenire sulla sezione figlia assegnato nel menu Utilizzatori, voce Amministratori. Per esempio, un redattore che normalmente non vede neppure la voce “modelli” dentro al menu *Pagine* di Sar-At può cancellarne i record se il Portale glielo consente.

- Consenti aggiunta record**
- Consenti cancellazione**

Completata la terza fase e confermata la creazione del campo portale, il portale appare nella scheda Sar-At. L’aspetto è quello della figura sottostante (qui vediamo un portale che mostra due soli record correlati e nel quale sono state consentite in fase tre sia le aggiunte che le cancellazioni, queste ultime attraverso l’icona del divieto).

	Title	Modification Date	Video_mp4	
<b>video</b> Nome file caricato sul cloud di Amazon	Sar-At in cinque minuti	2014-08-25 14:25:00	sarat5.m4v	
	Sar-At uso avanzato	2014-08-25 14:32:00	sarat_avanzato.m4v	

 [Aggiungi...](#)

### Uso avanzato dei portali

È possibile inserire contenuto attraverso il portale anche prima di registrare per la prima volta il record contenitore. Per esempio, chi sta creando la scheda della Nissan può inserire informazioni sulla Note prima di aver cliccato “Invia modifiche”. Naturalmente, potrebbe accadere che il redattore si interrompa dopo aver salvato i dati sul modello d’auto ma prima di aver registrato la scheda della casa madre. In questo caso, il contenuto del portale è orfano di un genitore, poiché è scollegato. Sar-At proporrà in questo caso i record orfani non appena si ricomincerà a inserire una nuova scheda nella sezione che contiene il portale, ma (per evitare errori umani e motivi di confusione) proporrà la ricomposizione del lavoro interrotto esclusivamente al redattore originale, quello che aveva inserito i record correlati rimasti temporaneamente orfani.

Se due amministratori accedono contemporaneamente a un record che contiene un portale, soltanto il primo può modificare il record, ma anche il secondo può aggiungere e togliere righe dal portale.

Se i record mostrati nel portale fanno riferimento a una sezione menu, Sar-At mostra il testo umanamente comprensibile e non il riferimento interno (per proseguire il nostro esempio di marche e modelli di automobili visto in precedenza, se un amministratore guarda la scheda della FIAT, e se la tabella con i modelli contiene un campo chiamato “in commercio” che vale 0 per indicare un modello obsoleto e invece 1 per indicare un’automobile in produzione, allora la colonna in questione non mostrerà i numeri 0 ed 1 ma bensì il testo “In produzione” o “Fuori commercio”).

### Ricerca a tutto testo

Le Viste studiate sin qui hanno un limite, che a volte può farsi seccante: ci permettono di ricercare solo all’interno di un campo tra i molti che costituiscono la sezione. In alcuni casi i nostri visitatori vorranno cercare una informazione senza sapere dove sia stata inserita. Per esempio, nel campo ingredienti, in ricetta o in note della nostra sezione gastronomia? Un altro esempio: in una sezione FAQ, è importante segnalare una pagina quando contiene nella domanda il testo cercato dal visitatore, ma anche se il testo cercato appare solo nella risposta.

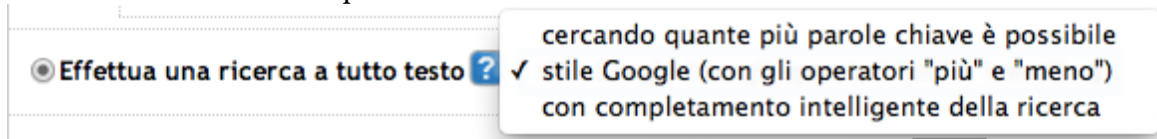
Se desiderate creare una vista che opera a tutto testo, dovete selezionare una volta per tutte quali campi debbono venire ricercati e quali esclusi. Cliccate il link *Seleziona i campi per la ricerca a tutto testo* nella schermata delle Viste, oppure selezionate la linguetta *Proprietà* e premete il pulsante *Ricerca a tutto testo*....

Setup ricerche a tutto testo - faq	
Campo	Incluso Nella Ricerca
title Titolo della pagina	<input checked="" type="checkbox"/>
keywords Parole chiave	<input checked="" type="checkbox"/>
description Sintesi della pagina	<input checked="" type="checkbox"/>
saratUrl URL	<input type="checkbox"/>
domanda Domanda del lettore	<input checked="" type="checkbox"/>
questionante Nome del lettore	<input checked="" type="checkbox"/>
testo Mia risposta alla domanda x	<input checked="" type="checkbox"/>

Ciò fatto, potete attivare l'opzione della ricerca a tutto testo nella schermata della Vista. Alcune osservazioni importanti:

1. Sar-At costruisce uno speciale indice di tutte le informazioni contenute nella sezione quando voi definite l'elenco dei campi. Questa operazione può richiedere alcuni minuti e occupa spazio sul disco rigido del server.
2. Nella costruzione dell'indice vengono escluse le parole brevi (tre lettere o meno) in quanto molto frequenti. Questo significa che se un visitatore del vostro sito cerca "DNA" o "DOC" non troverà nulla.
3. Per ogni sezione è consentito un solo indice, quindi non potete avere due Viste a tutto testo differenti per la medesima sezione.
4. Nelle comuni Viste, Sar-At cerca esattamente quello che viene specificato. Nelle viste a tutto testo, i termini possono essere disgiunti. Per esempio, se la scheda di Aristide Pallazzi contiene il testo "Il grande Pallazzi interpreta il ruolo del Barbiere di Siviglia per la prima volta durante una recita alla scuola materna nel 1986", allora usando una Vista normale la ricerca di "Palazzi Barbiere" non troverà nulla, perché nel testo i due nomi non sono vicini. Usando una Vista a tutto testo, la frase viene localizzata e segnalata.

Esistono tre differenti modalità per le Viste a tutto testo:



Il primo caso è quello base. Nel secondo caso, Sar-At permette che la ricerca utilizzi simboli analoghi a quelli accettati da Google. La tabella sottostante illustra le possibilità della ricerca stile Google.

Esempio	Spiegazione?
<b>motore di ricerca</b>	Trova anche la frase "La ricerca delle case automobilistiche ha prodotto un motore a basso consumo". Infatti le parole cercate non debbono essere limitrofe e la parola "di" viene ignorata in quanto breve.

Esempio	Spiegazione?
<b>“motore di ricerca”</b>	La frase dell'esempio precedente non viene trovata. Sar-At cerca esattamente l'espressione “motore di ricerca” nei testi.
<b>+motore ricerca</b>	La frase “Il motore diesel” viene trovata. La frase “La ricerca sul motore non inquinante” viene trovata e listata per prima. La frase “La ricerca scientifica produce nuovi risultati” non viene trovata, perché Sar-At considera obbligatoria la parola “motore” a causa della presenza del simbolo “+”
<b>+motore -ricerca</b>	La frase “Il motore diesel” viene trovata. La frase “La ricerca sul motore non inquinante” non viene trovata perché Sar-At esclude i campi che contengono la parola “ricerca” a causa della presenza del simbolo “-”
<b>&gt;motore &lt;ricerca</b>	Trova tutte le frasi che contengono la parola motore, o la parola ricerca, o entrambe. Lista per prime quelle frasi dove sono presenti entrambe le parole, poi quelle dove c'è la sola parola “motore” (il segno di maggiore indica che è maggiormente importante).
<b>motor*</b>	Vengono trovati tutte le frasi che contengono le parole motore, motori, motorizzazione, motorino e Motorhead. L'asterisco rappresenta una successione qualsiasi di caratteri
<b>+motore +(&gt;diesel &lt;benzina)</b>	L'uso di parentesi permette di strutturare le ricerche. In questo caso troveremo le frasi che parlano di motore a benzina e quelle che parlano di motore diesel. Le frasi che parlano genericamente di motore non verranno rilevate, a causa della presenza del segno +. I motori diesel saranno citati per primi.

La terza possibilità per le Viste, il completamento intelligente, offre autocorrezione degli errori e delle approssimazioni del visitatore al prezzo di un raddoppio dei tempi di ricerca (in un sito disneyano "cane che parla" trova le pagine su Pippo) .

## Viste: risposte a domande frequenti

### La Vista non viene risolta nella mastro classica

Ho creato una Vista in Sar-At. Quando la provo con lo strumento incorporato funziona correttamente. Quando la inserisco dentro la pagina mastro non viene risolta. Il mio HTML ha questo aspetto:

```
<ul>
  {begin la_mia_vista}
  <li>
    <a href='showPage.php?template=sezione&id={id}'>{title}</id>
  </li>
  {end la_mia_vista}
</ul>
```

Il codice HTML che vedo produrre a Sar-At (dentro al browser) ha questo aspetto:

```
<ul>
  {begin la_mia_vista}
  <li>
    <a href='showPage.php?template=sezione&id=102'>Nissan</id>
  </li>
  {end la_mia_vista}
```

</ul>

Questo comportamento appare quando Sar-At non può risolvere la Vista a causa di un errore di sintassi. Gli errori più frequenti sono:

1. Utilizzo delle pagine BEGIN e END in minuscolo (devono essere maiuscoli).
2. Spazi bianchi prima di BEGIN e END oppure prima della graffa chiusa (non devono essercene)
3. Uso di una entità SGML o HTTP al posto della graffa. Questo accade soprattutto a chi usa Dreamweaver per creare la pagine mastro. Dreamweaver talora sostituisce il carattere { con la combinazione %07B e il carattere } con la combinazione %07D

### Viste multiple

*Posso usare molte viste dentro la medesima pagina mastro? Posso riutilizzare la medesima Vista in pagine mastro diverse, anche cambiando l'impaginazione (il codice HTML)? Posso usare per due volte la medesima Vista dentro una singola pagina mastro?*

Sì e sì e ancora sì.

### Viste su più sezioni

*Posso creare una Vista che ricerca dentro più sezioni contemporaneamente? Vorrei fare un unico motore di ricerca che lavori su tutte le informazioni del mio sito e non un motore di ricerca per ogni sezione.*

Possiamo ottenere il risultato creando più viste e usandole insieme nella medesima pagina “risultato della ricerca”. Ogni vista lavorerà su una sezione distinta: per esempio avremo una vista “motore\_ricerca1” che opera sulla sezione “prodotti”, una vista “motore\_ricerca2” che lavora sulla sezione “risposte” e una vista “motore\_ricerca3” che lavora sulla sezione “recall”.

Nella pagina che mostra i risultati della ricerca, concateneremo le Viste. In una pagina classica, per esempio, avremo:

```
{IF motore_ricerca1}
  <p>Elenco dei prodotti il cui marchio contiene la parola {search}:</p>
  {BEGIN motore_ricerca1}
  <p><a href='showPage.php?template=prodotti&id={id}'>{title}</a></p>
  {END motore_ricerca1}
{ENDIF motore_ricerca1}
{IF motore_ricerca2}
  <p>Elenco delle risposte il cui titolo contiene la parola {search}:</p>
  {BEGIN motore_ricerca2}
  <p><a href='showPage.php?template=risposte&id={id}'>{title}</a></p>
  {END motore_ricerca2}
{ENDIF motore_ricerca2}
{IF motore_ricerca3}
  <p>Elenco dei recall il cui nome contiene la parola {search}:</p>
  {BEGIN motore_ricerca3}
  <p><a href='showPage.php?template=recall&id={id}'>{title}</a></p>
  {END motore_ricerca3}
{ENDIF motore_ricerca3}
```

### Vista di viste

*Posso fare una vista di viste? Cioè mettere una Vista dentro a un'altra?*

In una pagina mastro classica no, ma nella quasi totalità dei casi questo scopo si può ottenere con una vista ad hoc. In una pagina mastro Twig sì, al prezzo di un po' di complicazione. Per esempio, immaginate di avere una sezione “news” e di voler rappresentare queste notizie, pressate con una Vista, ma raggruppandole per mese. Twig mette a disposizione variabili e cicli, e la cosa diventa facilmente realizzabile:

```
{% set mese_corrente = "Nessuno" %}

{% for faq in ultime_cinquanta_news %}
    {% if ((faq.lastMod | date("F")) != mese_corrente) %}
        {% set mese_corrente = faq.lastMod | date("F") %}
        <h1>{{faq.lastMod | date("F Y")}}</h1>
    {% endif %}
    <li>{{faq.lastMod | date("d/m/Y")}} <em>{{faq.title}}</em></li>
{% endfor %}
```

### Cancellare una Vista

*Voglio cancellare una Vista, che credo non serva più, ma temo di sbagliare. Ho paura che sia in uso in una pagina mastro di cui mi sono dimenticato. Che faccio?*

Sar-At impedisce la cancellazione delle Viste indispensabili al funzionamento del sito. Il programma inoltre mostra nella scheda della Vista un elenco di tutte le pagine mastro che incorporano quella vista:

per vederlo basta cliccare il pulsante 

Se desiderare ricreare una vista da zero, la protezione contro la cancellazione delle Viste dà fastidio. In questo caso cambiate etichetta alla vista cancellanda, create la nuova versione della vista usando il nome che intendete mantenere, cancellate la vista originale e rinominata, infine validate nuovamente le pagine mastro coinvolte.

# Parte terza: i form

## Introduzione: cosa sono i moduli

Nelle pagine HTML è possibile inserire moduli o, in inglese, *form*. Un modulo può contenere diversi elementi di input.

In termini HTML, il modulo di esempio appare così:

```
<form action="showForm.php" method="post">
  <fieldset>
    <legend>
      Un modulo HTML
    </legend>
    Linea di testo:
    <input type="text" name="testo" size="20" />
    <br />
    Area di testo:
    <textarea name="at" rows="5" cols="30">
    </textarea>
    <br />
    <fieldset>
      <legend>
        Caselle radio
      </legend>
      RAI1: <input type="radio" name="r" value="rai1" />
      RAI2: <input type="radio" name="r" value="rai2" />
    <br />
    </fieldset>
    Casella di spunta: <input type="checkbox" name="cas" />
    <br />
    Selezione multipla: <select name="sm" multiple="multiple">
      <option selected>
        Braccio di Ferro
      </option>
      <option>
        Goldrake
      </option>
```



```
        <option>
            Topolino
        </option>
        <option selected>
            Cocco Bill
        </option>
    </select>
    <br />
    Menu a tendina: <select name="menu">
        <option selected='selected'>
            Primi
        </option>
        <option>
            Secondi
        </option>
        <option>
            Contorni
        </option>
        <option>
            Dolci
        </option>
    </select>
    <br />
    <input type="submit" value="Invio" />
</fieldset>
</form>
```

Per esempio, i moduli si possono utilizzare per consentire ai visitatori del sito di inviare un messaggio alla redazione del sito.

Tradizionalmente problematica sul web è la gestione del controllo. Se chiediamo al visitatore di lasciarci il suo indirizzo email, non vogliamo accettare che un burlone scriva “sono@ffari-miei.it” in quello spazio.

Sar-At permette la creazione e gestione di moduli complessi e sofisticati e risolve alla radice tutti i problemi citati. Non è richiesta alcuna forma di programmazione.

## La pagina mastro per il modulo

Sar-At consente di associare alla sezione modulo più pagine mastro. La pagina di default deve essere una pagina che contiene il form vero e proprio. Essa può contenere Viste e altre struttura di controllo tipiche di Sar-At.

Sar-At offre un componente, chiamato `showForm.php`, che viene usato per generare lo HTML da inviare al browser dopo aver risolto le Viste. Lo stesso `showForm`, inoltre, riceve l’input dei visitatori e lo controlla sulla base delle vostre indicazioni. Se tutti i controlli sono soddisfatti (tutti i campi obbligatori sono compilati, i valori minimi e massimi rispettati, eccetera) allora `showForm` manda immediatamente il visitatore a una pagina di atterraggio; tipicamente questa contiene un messaggio di ringraziamento per la compilazione avvenuta. Se qualcosa non va `showForm` ripresenta al visitatore il modulo, precompilato con i valori digitati poco prima, ma mettendo in evidenza la risposta (o le risposte) inaccettabili attraverso grafica e messaggio d’errore scelti da voi.

La redazione della pagina mastro per `showForm` assomiglia parecchio alla redazione della pagina mastro `showPage`, con alcune importanti differenze.

1. `showForm` dispone di una serie importante di funzionalità altamente specializzate per la produzione di *form*, che vedremo tra poco. Da questo punto di vista il linguaggio Twig è,

invece, poco dotato. Nulla vi impedisce di creare un form con Twig, ma Sar-At non vi aiuta in questo.

- Quando chiamate `showPage`, Sar-At ha sempre a disposizione una sezione con contenuti compilato dalla redazione e da essa estrae i dati da mostrare all'utente. Al contrario `showForm` parte da una sezione che è vuota quando il sito va inizialmente in linea.
- Di conseguenza, `showPage` viene chiamato quasi sempre indicando una `id` o un `masterId`, che selezionano il contenuto della base dati da mostrare nella pagina mastro. Al contrario `showForm` non viene chiamato quasi mai con un `id` e mai con un `masterId`.

Se chiamate `showPage` senza un `id` normalmente volete produrre una pagina home di sezione e recuperate contenuto dalla base dati solo attraverso Viste. Se chiamate `showForm` con un `id` state indicando che quell'`id` corrisponde al visitatore registrato che ha fatto login sul vostro sito e che vuole aggiornare la sua anagrafica (Sar-At verificherà che questo sia il caso, per sicurezza). Tutti i dati inseriti nella base dati da `showForm` hanno `masterId` uguale a zero.

- Nulla e nessuno vi impedisce di accedere alla medesima sezione usando una combinazione di `showPage`, `showForm` e `showTwig`. Per esempio, in un blog, la tabella che ospita i commenti dei visitatori potrebbe venire popolata con `showForm`, i commenti in calce a un post del blog potrebbero venire recuperati con una Vista e mostrati da `showPage`, la home page del blog potrebbe venire fatta con `showTwig`.

#### Notifica della compilazione e pagina di ringraziamento

Quando chiediamo a Sar-At di creare una *sezione modulo*, il programma ci chiede di indicare un indirizzo di posta elettronica a cui verranno notificate istantaneamente tutte le compilazioni effettuate dai visitatori, a fusa di `showForm`. Questo indirizzo è opzionale e, se assente, `showForm` non effettua notifiche.

**Sezioni modulo** ?

Modulo compilabile dai visitatori  
La sezione può venire riempita dai visitatori che compilano un modulo gestito da Sar-At.

**E-mail** :

Indirizzo (opzionale) a cui viene immediatamente notificata la compilazione di un modulo

**URL** :

Indirizzo a cui inviare il visitatore che ha compilato con successo il modulo

Viene anche richiesto l'indirizzo URL di una pagina web da mostrare a ogni visitatore che compila il modulo. Questo indirizzo è obbligatorio. Possiamo utilizzare sia l'indirizzo di una pagina web statica (per esempio `http://www.ilmiosito.it/grazie.html`) sia l'indirizzo di una pagina dinamica composta da Sar-At stesso (per esempio `http://www.ilmiosito.it/nome-cartella-magica/grazie-mille-1234567.html`).

Create ora la sezione vera e propria, avendo cura di definire un campo per ogni voce del modulo HTML. Fate attenzione alle specifiche di validazione, come mostrate qui di seguito:

<b>Nome campo</b>	scadenza_carta_credito
<b>Note</b>	Anno, mese e giorno di validità
<b>Tipo</b>	Data
<b>Opzionale</b>	Può essere nullo, zero o vuoto <input type="checkbox"/> Deve essere unico <input type="checkbox"/>
<b>Visibile</b>	Editabile dagli amministratori <input checked="" type="checkbox"/>
<b>Massimo</b>	2024-12-31
<b>Minimo</b>	
<b>Default</b>	
<b>Validazione</b>	futura

Il valore di **default** viene pre-compilato dentro al modulo. Il visitatore può modificarlo. Se il visitatore cancella il valore del campo ma non lo rimpiazza con un altro, Sar-At inserisce nuovamente il valore di default.

Se il campo **può essere nullo, zero o vuoto**, Sar-At accetterà il modulo quando il visitatore preme il pulsante di invio senza aver scritto o selezionato alcunché. Viceversa il campo non è opzionale, Sar-At non accetterà l'invio del modulo sinché il visitatore non ha compilato il campo con un valore significativo. Un campo a cui avete assegnato un valore di default non resterà mai vuoto, quindi non è necessario indicarlo come opzionale.

Se il campo ha un valore **minimo**, Sar-At non accetterà alcun valore più basso. Nel caso di campi numerici, il significato è evidente. Nel caso di campi testuali, Sar-At utilizza un criterio alfabetico, quindi "Auto" si considera minore di "Bicicletta". Il valore **massimo** è analogo al minimo.

La **validazione** può venire utilizzata per altri scopi specializzati. Per esempio, le date possono venire verificate come appartenenti al passato (per esempio, una data di nascita) o al futuro (per esempio, la data di scadenza di una carta di credito). Oltre alle validazioni pre-installate con ogni copia di Sar-At, altre possono venire codificate su misura e su richiesta. Contattate il nostro servizio commerciale per un preventivo.

Quando create la sezione modulo, vi consigliamo di selezionare "pagina mastro sintetica" nella prima riga. Sar-At genera un prototipo di modulo che contiene già i menu a discesa, le caselle di spunta, testi lunghi (textarea) e brevi (input type text) come più appropriato, e inoltre aggiunge un semplice ma efficace sistema di gestione degli errori (che tratteremo nel prossimo paragrafo). È molto semplice prendere questa pagina mastro modulo sintetica e impaginarla: davvero più semplice che crearla da zero.

## Errori di compilazione

Se il visitatore del vostro sito compila il modulo commettendo errori e invia la pagina senza che questi errori vengano intercettati — in pratica, se non avete aggiunto del Javascript nella pagina mastro per intercettare gli errori di compilazione — showForm riconosce il problema. In questo caso l'inserimento del post nella base dati non avviene. Sar-At reagisce ripresentando il form, e inserisce automaticamente l'input del visitatore al suo interno di modo che egli possa correggerlo senza dover re-introdurre tutto daccapo.

Potete inserire nello HTML dei messaggi d'errore per aiutare il visitatore che appaiono solo questo caso. Per una spiegazione dettagliata e completa di esempi, si veda il paragrafo ONERR in questo stesso capitolo.

State testando il modulo, vi pare di aver compilato tutto correttamente, ma questo riappare? Se siete bloccati alla ricerca dell'errore, di compilazione o di progettazione, Sar-At può aiutarvi. Fate log-in come amministratore in Sar-At e tentate nuovamente la compilazione del form. Vedrete apparire una serie di indicazioni, in rosso, in calce alla pagina, precedute dalla scritta *Informazioni extra per l'amministratore attualmente collegato*.

Viene indicato il campo inaccettabile (o i campi inaccettabili) del modulo, e il motivo per cui showForm non può procedere. La tabella qui sotto riporta i codici di errore.

Codice	Significato
0	Campo valido
1	Campo non valido. Per esempio un testo nello spazio riservato a un numero
2	Campo non valido per il plugin di validazione (vedi paragrafo precedente)
3	Campo obbligatorio lasciato in bianco, per il quale non è stato neppure indicato un valore di default nella linguetta <i>Struttura</i> del Sar-At
4	Campo di tipo <i>file</i> al quale corrisponde una cartella sul server che non è accessibile in scrittura a Sar-At
5	Tipo di file inaccettabile in upload
7	Immagine in upload di dimensioni non consentite (confrontare le dimensioni minime e massime previste nella linguetta <i>Struttura</i> del Sar-At per questo campo)
8	Nella linguetta <i>Struttura</i> del Sar-At per questo campo si è deciso che ogni record deve avere un valore unico, ma il dato ora inserito è già presente nella base dati.
502	La sezione contiene un campo obbligatorio, per il quale non è stato neppure indicato un valore di default nella linguetta <i>Struttura</i> del Sar-At, ma la pagina mastro non contiene uno spazio per esso nel suo HTML
503	Il campo andrebbe valicato con un plug-in, ma il plug-in con questo nome è stato cancellato sul server

Gli errori con codice superiore a 100 non possono venire visualizzati durante l'esecuzione di showForm, ma appaiono solo all'interno di Sar-At durante la validazione della pagina mastro.

## Modulo a tappe

Se i dati che dovete chiedere al visitatore sono parecchi, il modulo che lui o lei dovrà compilare saranno difficilmente stipabili in una singola pagina web. In questo caso Sar-At vi consente di dividere la raccolta dati in più sottopagine. L'esperienza utente è quella tipica dei *wizard* della Microsoft, popolarizzati in Windows.

Se create un modulo a tappe, Sar-At eseguirà sempre e comunque i suoi controlli sul contenuto di ogni campo. Sarà vostro esclusivo compito verificare che tutti i campi siano presenti in una e una sola delle pagine-tappa. Per i campi obbligatori, Sar-At verificherà che non vengano immessi con valore nullo o zero.

Ogni tappa è contenuta in una pagina mastro, e ogni pagina mastro va validata per la sezione modulo in questione. Immaginiamo che un sistema di registrazione venga diviso in tre tappe: si pone il problema di come spiegare a Sar-At in quale ordine vadano eseguite le tappe.. La risposta: è sufficiente inserire un parametro nel form, chiamato `sarat_sf`, che punta alla pagina mastro della tappa successiva. Il parametro sarà presente ma vuoto nell'ultima tappa.

Facciamo un esempio, immaginiamo dunque di aver creato una sezione modulo chiamata registrazione e tre pagine mastro chiamate `registrazione1.html`, `registrazione2.html` e `registrazione3.html`, ciascuna con una tappa.

Il codice HTML rilevante sarà realizzabile con semplicità. Dentro `registrazione1.html`:

```
<form action='showForm.php' method='post'>
```

```
<input name='template' type='hidden' value='registrazione' />
<input name='masterPage' type='hidden' value='registrazione1.html' />
<input name='sarat_sf' type='hidden' value='registrazione2.html' />
```

Dentro `registrazione2.html`:

```
<form action='showForm.php' method='post'>
  <input name='template' type='hidden' value='registrazione' />
  <input name='masterPage' type='hidden' value='registrazione2.html' />
  <input name='sarat_sf' type='hidden' value='registrazione3.html' />
```

Dentro `registrazione3.html`:

```
<form action='showForm.php' method='post'>
  <input name='template' type='hidden' value='registrazione' />
  <input name='masterPage' type='hidden' value='registrazione3.html' />
  <input name='sarat_sf' type='hidden' value='' />
```

Tutte le tappe del sito possono comprendere upload di file (ovvero la popolazione di campi che ospitano immagini e altri documenti caricati dal visitatore).

## Autoregistrazione e moduli

Se il vostro sito contiene pagine riservate a membri di un gruppo, vi è possibile creare una sezione che permette ai visitatori di iscriversi. Per esempio, chi desidera che i visitatori lascino i loro dati anagrafici spesso inserisce sul sito alcune pagine interessanti e ne permette la lettura solo a chi compila un modulo con i suoi dati.

Quando definite un modulo di autoregistrazione, è opportuno che Sar-At abbia qualche basilica cognizione di quel che state facendo. Se uno dei campi serve a registrare l'indirizzo email del visitatore, Sar-At potrà venire utilizzato per inviare una newsletter a tutti gli iscritti. La linguetta "modulo" che appare a video vi permette di identificare quale sia il campo usato per l'email.

Analogamente, se Sar-At viene istruito su quale campo contenga il cognome del visitatore, o la sua città nativa, allora potrà autocompilare il modulo di registrazione per visitatori che si fanno riconoscere grazie a una interconnessione con Facebook.

L'autoregistrazione è fondamentale anche nel commercio elettronico, dove noi abbiamo bisogno di tutti i dati di un visitatore per vendergli qualche cosa. Ne ripareremo nel capitolo dedicato all'e-commerce.

L'opzione di autoregistrazione appare come opzione nelle sezioni modulo quando voi avete creato un gruppo di utenti registrati. Sar-At richiede che il modulo contenga un campo "password" nella sezione modulo e un campo qualsiasi, di tipo breve testo e caratterizzato come *Unico* nella linguetta *Struttura* da usare come username. Per esempio, potreste usare l'indirizzo email dei visitatori. Se registrate la mail dei visitatori ma non la usate come username, potete indicare quale campo usate per memorizzare l'indirizzo di posta elettronica, e permetterete a Sar-At di scrivere ai visitatori quando ce n'è bisogno, come vedremo nel seguito del manuale - per esempio per inviare le conferme d'ordine del commercio elettronico.

Se abilitate questa opzione, ogni utente che compila con successo il modulo viene inserito nel gruppo indicato (automaticamente o dietro conferma, come vedremo tra poco).

Un utente registrato attraverso il modulo di autoregistrazione può tornare sul vostro sito semplicemente accreditandosi con la password che si è scelto in fase di registrazione e con uno qualsiasi tra i campi unici (vedere il paragrafo "i campi unici" in questo stesso capitolo).

**Autoregistrazione ?**  
Chi compila questo modulo viene iscritto al gruppo

Mostra anche i gruppi creati automaticamente

Automaticamente

Dopo conferma per email

Previa tua approvazione

Manualmente

Per un periodo di  mesi.

Il visitatore registrato può fare log-in usando come nome utente il campo *receiver\_email*

**Newsletter**  
Sar-At potrebbe inviare messaggi agli utenti registrati qui.

**Recupera indirizzo visitatore dal campo:**

Indirizzo mittente e nome mittente vengono indicati in **questa schermata.**

**Promemoria per email ?**  
L'utente registrato può recuperare la password

**Consenti il recupero tramite e-mail di una password dimenticata**

**Oggetto dell'email spedita:**

Tramite un form con action `sarat/rememberByEmail.php`, e parametri chiamati *username* e *template*.

**Se non si trova la username mostra la pagina:**

**In caso di successo invece mostra la pagina:**

ed invia per e-mail le informazioni impaginate con

**Indirizzo mittente dell'e-mail:**

**Nome mittente**

**Registra i tentativi di recupero in un log. Consentine uno ogni**  **minuti.**

Se abilitate l'autoregistrazione e selezionate l'opzione "Dopo conferma per email", l'utente che compila il modulo appare nella tabella degli utenti registrati ma non può accedere immediatamente con la password che si è scelto ai contenuti riservati. Sar-At manda all'utente una mail dove gli chiede la conferma della volontà di iscriversi (in italiano, inglese e russo). La mail contiene un link da cliccare per

accettare l'iscrizione. Quando l'utente fa il clic, torna sul vostro sito. Sar-At automaticamente ne completa l'iscrizione, lo abilita all'accesso alle sezioni riservate al gruppo dei registrati se ce ne sono, e gli mostra la pagina indicata in questa stessa schermata, "Modulo", sotto la voce URL.

L'utente che fa clic sul link di accettazione fa anche automaticamente log-in sul vostro sito.

## Registrazione e social network

Possedere un parco di utenti registrati è un obiettivo valido per qualsiasi sito web, ma il modo migliore per raggiungere questo obiettivo cambia di caso in caso.

Un commercio elettronico che intende spedire merce a casa dovrà conoscere tutti i recapiti della sua clientela e avrà vantaggio da informazioni su quali siano le marche preferite, le passioni, la propensione a spendere, la fascia d'età e via dicendo -- informazioni che sostanzialmente il cliente ci fornisce in forma diretta, esplicitamente o implicitamente. Esempio opposto: un blog ha interesse a ricevere commenti intelligenti ma poco importa che di ogni commentatore esista una scheda dettagliata.

Per i siti più orientati verso questa seconda estremità dello spettro, Sar-At introduce la possibilità di accettare visitatori registrati altrove -- su un *social network*.

Il social network offrirà alcune informazioni di base sulla persona, la quale potrà semplicemente autorizzare il social network a passarcele. Le informazioni che potremmo ricevere sono:

Campo	Valore
identifier	Una sequenza di numeri e lettere univoca per il network consultato
profileURL	Link alla pagina del profilo dell'utente sul social network
photoURL	Fotografia (o avatar) del visitatore
displayName	Nome utente da mostrare a video
description	Auto-presentazione dell'utente
firstName	Nome di battesimo
lastName	Cognome
gender	Sesso (M o F)
language	Lingua preferita
age	Età
birthDate	Data di nascita (a volte solo giorno e mese)
email	Indirizzo di posta elettronica dichiarato
emailVerified	Indirizzo che il social network ha verificato corrispondere all'utente
phone	Numero di telefono
address	Indirizzo di casa
country	Nazione di residenza
region	Provincia
city	Città (mentre questo manuale viene scritto, Facebook fornisce però la città di nascita e non quella di residenza)
zip	Codice di avviamento postale

Bisogna sottolineare che non tutti i social network forniscono tutte queste informazioni (in generale, Facebook tende ad essere il più prolifico) e non tutti gli utilizzatori dei social network compilano completamente la loro scheda di profilo utente. Quindi se si sceglie questa strada bisogna accontentarsi di ciò che arriva.

Vediamo brevemente alcuni casi comuni e come si comporta Sar-At di conseguenza.

#### PRIMO CASO - GRUPPO SEMPLICE NON COLLEGATO AD UNA TABELLA DI ANAGRAFICHE

Avete definito nel menu Utilizzatori di Sar-At un gruppo “lettori”, a cui avete dato accesso alla sezione “ricette” del vostro sito. Non avete fatto altro, ovvero questo gruppo non è collegato un modulo nel quale è richiesto di compilare i propri dati anagrafici. Magari potreste aver importato direttamente in questo gruppo alcune username o password di vostri conoscenti o clienti storici.

Se ora abilitate l'accesso da Facebook e Twitter, chiunque acceda attraverso questi social network gode di tutti i privilegi associati ai lettori, ovvero può consultare le ricette. I visitatori così autenticati non sono affatto registrati in Sar-At. Chi utilizza una connessione Facebook per entrare sul vostro sito risulta automaticamente abilitato per un certo periodo di tempo (tipicamente la durata di una sessione di navigazione). L'abilitazione può venire riattivata automaticamente all'accesso successivo.

#### SECONDO CASO - GRUPPO SEMPLICE COLLEGATO A UNA ANAGRAFICA

Avete definito nel menu Utilizzatori di Sar-At un gruppo “commentatori” a cui avete dato accesso alla sezione “commenti” del vostro blog dove vengono registrati interventi in risposta ai post del blog. Avete stabilito che ogni commento va firmato dall'autore. Avete creato una sezione modulo aggiornabile chiamata “anagrafiche\_commentatori” con le anagrafiche dei visitatori, prevedendo i campi indicati nella tabella precedente (da identifier a zip), tutti non obbligatori.

Se ora abilitate l'accesso da Google e Foursquare, chiunque acceda attraverso questi social network gode di tutti i privilegi associati ai commentatori, ovvero può postare commenti. I dati provenienti dai network vengono automaticamente registrati da Sar-At nella tabella anagrafiche\_commentatori.

#### TERZO CASO - GRUPPO COMPLESSO COLLEGATO A UNA ANAGRAFICA

Avete definito nel menu Utilizzatori di Sar-At un gruppo “clienti” a cui avete dato accesso al commercio elettronico del vostro sito. Avete creato una sezione modulo aggiornabile chiamata “anagrafiche\_clienti” con le anagrafiche dei visitatori, prevedendo campi banali come il cognome ma anche campi extra come il codice fiscale, che nessun social network gestisce.

Se ora abilitate l'accesso da Yahoo e Facebook, chiunque acceda attraverso questi social network si vede presentare la schermata di registrazione (il modulo gestito da showForm), ma in questo modulo alcuni campi sono precompilati -- quelli riservati ai dati il cui valore ci è inviato dai social network.

#### CASI PARTICOLARI

Immaginate di intervenire sul blog descritto nel caso 2, indicando che l'informazione sull'email del visitatore è obbligatoria. Di primo acchito sembrerà che nulla sia cambiato: normalmente un utente si collegherà attraverso il social network e potrà immediatamente lasciare un commento.




La medesima esperienza descritta per il caso 3 accadrà però -- automaticamente -- a quegli eventuali visitatori che non avessero condiviso con il social network il proprio indirizzo email.

### Configurazione per social network

Se desiderate accettare registrazioni da network esterni dovete innanzitutto chiedere l'abilitazione al network stesso. Nel menu Utilizzatori di Sar-At troverete alla voce Social Network un elenco di possibili partner predisposti. Cliccando il pulsante “Configura” accedete alla scheda del network e da lì incontrate un link diretto alla pagina web dove dovrete stipulare l'accordo di partnership gratuito, in modo da ricevere una coppia di valori (di solito chiamati





“id” e “secret”) che dovrete incollare dentro Sar-At. Una volta configurato un network è potenzialmente utilizzabile per il log-in dei visitatori

 <p>Google</p>	<p>Sar-At dispone delle informazioni necessarie per collegarsi a questo network.</p> <p>Riconfigura...</p>
 <p>LinkedIn</p>	<p>Sar-At dispone delle informazioni necessarie per collegarsi a questo network.</p> <p>Riconfigura...</p>
 <p>Live</p>	<p>Configura...</p>

I social network non configurati appaiono con una icona in bianco e nero, che diventa colorata quando la configurazione è completata. La configurazione effettuata resta editabile (il pulsante diventa “riconfigura”).

Oltre ai social network più importanti, già predisposti per la connessione, Sar-At vi consente di scambiare credenziali con altri siti web, utilizzando un protocollo standard chiamato OpenID. Cliccate il pulsante “aggiungi nuovo network...” in basso per abilitare la interconnessione. Nella schermata che appare digitate il nome del network e l’identificatore OpenID da esso fornito.

I network abilitati via OpenID sono eliminabili dalla schermata principale.

 <p>StackOverflow</p>	<p>Sar-At dispone delle informazioni necessarie per collegarsi a questo network.</p> <p>Riconfigura...  Elimina</p>
--	--

### Gruppi di utenti e social network

Sar-At vi consente di accettare alcuni network solo in alcune circostanze. Per esempio, se in una sezione del vostro sito state cercando contatti da liberi professionisti potreste accettare il solo LinkedIn, mentre nell’area dedicata ai consumatori potreste prediligere Facebook e per i commenti del blog puntare su Twitter.

Per ciascun gruppo di utenti registrati che il vostro sito definisce (menu Utilizzatori) potete abilitare e disabilitare individualmente i social network.

#### Autenticazione Esterna

Puoi consentire l’accesso a utenti registrati altrove, se lo desideri. Per abilitare gli utenti di un social network che qui appare in grigio, [configura il collegamento](#) con quel sito. Se il tuo sito richiede ai visitatori anonimi che vogliono diventare membri del gruppo *abbonati* una informazione che il social network non ci passa, verrà loro richiesto di compilare il modulo di registrazione già parzialmente riempito con quei pochi dati che il social network ci offre.

- AOL
- Facebook
- Foursquare
- Google

### Presentazione al visitatore dell’opzione social

Come si è visto, Sar-At evidenzia ogni network con una icona. Le icone dei social network predefiniti sono inserite in Sar-At mentre le icone dei network a cui vorrete collegarvi attraverso il protocollo OpenID naturalmente non lo sono.

Potete comunque sostituire le icone predefinite e aggiungere nuove icone per i network OpenID in modo semplicissimo. Nella cartella che ospita il vostro sito create una cartella chiamata `img`, se già non ne esiste una. Registrare al suo interno una immagine in formato PNG per ciascun network, dandole il medesimo nome con cui Sar-At identifica il network in questione. Prestate attenzione alla differenza tra lettere maiuscole e minuscole. Per esempio, LinkedIn richiede in maiuscolo sia la L iniziale che la I nel mezzo.

La meccanica precisa del log-in, è spiegata al paragrafo “loginBuyer” dell’ultima parte di questo manuale; ovvero il codice HTML da incollare nel vostro sito, con tutte le opzioni disponibili, è trattato in quella sede. Si veda anche il plug-in standard “network\_authentication” discusso nella parte quinta.

#### Anagrafica e social network

Abbiamo spiegato nelle pagine precedenti quali e quante informazioni i social network ci forniscano. È spesso opportuno accoppiare queste informazioni ad altre. Per esempio, se sto scrivendo un commercio elettronico che si basa su PayPal, scopro che PayPal memorizza l’indirizzo di posta elettronica dei clienti in un campo chiamato `payer_email`. La tabella vista nelle pagine precedenti mi informa del fatto che Facebook invece utilizza la nomenclatura `emailVerified`. È opportuno che questi due dati siano fermamente correlati nella nostra base dati, di modo che quando un cliente entra da Facebook, il suo indirizzo email venga automaticamente comunicato a PayPal al momento del pagamento, senza che il cliente debba digitarlo.

Notate che, quando si abilita il pagamento PayPal, Sar-At crea automaticamente una sezione anagrafica adatta (chiamata `paypal_transaction`) e già predisposta per l’uso con i social network secondo la modalità che verrà discussa nel resto di questo paragrafo. Quindi nel caso dell’esempio non è richiesta alcuna azione da parte vostra).

<b>Nome campo</b>	<input type="text" value="payer_email"/>
<b>Note</b>	<input type="text" value="Email identificativa acquirente"/>
<b>Tipo</b>	<input type="text" value="Breve testo"/>
<b>Dato anagrafico corrispondente</b>	<input type="text" value="emailVerified"/>
<b>Opzionale</b>	Può essere nullo, zero o vuoto <input type="checkbox"/> Deve essere unico <input checked="" type="checkbox"/>
<b>Visibile</b>	Editabile dagli amministratori <input checked="" type="checkbox"/>
<b>Massimo</b>	<input type="text"/>
<b>Minimo</b>	<input type="text"/>
<b>Default</b>	<input type="text"/>
<b>Validazione</b>	<input type="text" value="email"/>

Se avete importato le anagrafiche dei clienti da un foglio Excel o da una base dati fornita dal Centro Elaborazione Dati dell’azienda a cui appartiene il sito, i campi dell’anagrafica avranno nomi arbitrari. Per esempio, probabilmente esisterà il campo “cognome”, che andrebbe accoppiato a ciò che i social network chiamano “lastName”. Visitando la linguetta Struttura della vostra sezione modulo contenente dati anagrafici vedrete apparire un rigo intitolato “dato anagrafico corrispondente”. In questo rigo potete (e dovrete) indicare se questo campo della vostra anagrafica ha un qualche significato per i social network. Ciò fatto, quel campo verrà automaticamente compilato

da Sar-At quando un visitatore si fa autenticare da uno dei network abilitati.

Nel menu a tendina “dato anagrafico corrispondente” appaiono in grigio i nomi dei campi già accoppiati e dunque non ulteriormente accoppiabili.

### Modulo aggiornabile

Se il vostro sito permette ai visitatori di registrarsi e accedere così a un’area riservata, ovvero se serve per registrare dati anagrafici del visitatore, allora potete consentire ai vostri visitatori già registrati di tornare nel modulo di registrazione per aggiornare i loro dati.

Immaginiamo che il processo di registrazione lasci l’anagrafica della persona dentro una sezione chiamata `membri_club`. Se questo è il caso allora la URL del modulo da compilare sarà

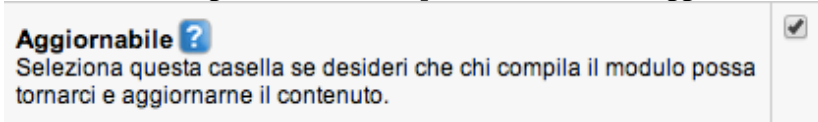
`showForm.php?template=membri_club`

e l’indirizzo a cui mandare i già soci perché aggiornino i propri dati sarà

`showForm.php?template=membri_club&id=ANY`

La clausola `id=ANY` viene riconosciuta da `showForm` e significa “tra le tante, trova e mostra la scheda di questo specifico visitatore registrato”.

La ri-registrazione va abilitata dall’interno di Sar-At. Nel menu Pagine scegliete la sezione dove si registrano i visitatori. Ciccate la linguetta Modulo. Spuntate la casella “Aggiornabile”.



Se consentite la ri-registrazione, dovete assolutamente fare uso del plug-in “sarat\_stopgap” che è presentato più avanti in questo stesso manuale, nella sezione dedicata ai plugin. In caso contrario, la sicurezza del vostro sito è compromessa.

Spesso potrete riutilizzare per l’aggiornamento dei dati la stessa pagina mastro che gli utenti compilano per registrarsi la prima volta. Può però succedere che voi preferiate lasciar aggiornare soltanto alcune informazioni, mentre le altre devono restare invariate. Per esempio, spesso non vogliamo che l’utente possa cambiare la sua username (si veda anche, a questo proposito, la voce `renameBuyer` nel capitolo sette di questo stesso manuale).

Se è questo il vostro caso, voi creerete una pagina mastro alternativa per la sezione modulo aggiornabile, e manderete i visitatori registrati ad aggiornare la propria scheda chiamando un indirizzo del tipo

`showForm.php?template=membri_club&masterPage=mastro_alternativa.html&id=ANY`

Tenete presente che in Sar-At, una pagina mastro usata per l’aggiornamento può contenere i soli campi editabili, e trascurare completamente l’esistenza degli altri, i cui valori saranno automaticamente mantenuti.

### Registrazione alla newsletter

Tutti i gruppi di visitatori registrati sono usabili come bersaglio di un invio newsletter. Create un campo chiamato “newsletter” di tipo *Breve testo* in modo che sia possibile per i visitatori auto-registrati gestire da soli la propria iscrizione o disiscrizione alla newsletter. Ecco un esempio di codice HTML valido per una pagina di registrazione

```
<p>Vuoi ricevere la nostra newsletter?</p>
<input type='radio' id='A' name='newsletter' value='A' checked='checked' />
<label for='A'> Iscritto</label>
<input type='radio' id='U' name='newsletter' value='U' />
<label for='U'> Non iscritto</label>
```

Ecco un equivalente esempio un po’ più complesso per una pagina aggiornabile di `showForm`. Se usate questo esempio ricordate di indicare un valore di default A nella struttura del campo Newsletter

```
<p>Vuoi ricevere la nostra newsletter?</p>
{RADIOBUTTON newsletter A}
<label for='sarat_A'> Iscritto</label>
{RADIOBUTTON newsletter U}
<label for='sarat_U'> Non iscritto</label>
```

Naturalmente potete scegliere di gestire l'interfaccia utente anche con una select o un checkbox.

Il significato esatto dei flag A, U ed X è spiegato nel manuale "Governare il sito" nel paragrafo "importazione dei visitatori registrati".

### Gestione degli errori

Interessanti studi dimostrano che i moduli HTML sono per i consumatori uno degli aspetti più scoraggianti del web. Sono due le problematiche che spesso rendono frustrante l'esperienza della compilazione. Primo: i messaggi di errore sono mal posti, sia per linguaggio che per collocazione fisica all'interno della pagina, e non rendono banale la correzione degli errori di compilazione come invece dovrebbero. Secondo: a volte gli errori costringono il visitatore a reinserire più volte i dati.

Sar-At risolve completamente la seconda problematica e vi offre tutti gli strumenti per affrontare la prima.

La tecnologia più frequentemente utilizzata per eseguire il controllo a livello browser è un mix di HTML5 e JavaScript. Alcuni controlli riescono bene in quel modo (per esempio, è facile controllare che il visitatore sia costretto a compilare un certo campo). Altri sono praticamente impossibili (per esempio, rifiutare un indirizzo email inesistente come "sono@ffari-miei.it", verificando che non esiste un sito chiamato ffari-miei.it su Internet). La soluzione Sar-At incarica il server centrale di eseguire i controlli e risolve tutti questi problemi alla radice.

Per dimostrare in pratica la gestione degli errori in Sar-At, immaginiamo di voler chiedere una opinione a chi ha acquistato un nostro prodotto. Cominceremo, naturalmente, definendo una sezione modulo con tre campi.

Voto: numero intero, minimo 5, massimo 7, default 6

Email: breve testo, non può essere lasciato in bianco, validazione email.

Commento: testo lungo, può essere lasciato in bianco.

Il modulo da compilare, molto semplice, ha questo aspetto:

Il codice HTML puro (per ora privo di istruzioni specifiche Sar-At) è il seguente:

```
<form style='width: 350px; background: pink;' action='showForm.php'>
  <fieldset>
    <legend><strong>Come valuti questo prodotto?</strong> </legend>
    Voto:
      <label><input type="radio" name="voto" value="5" />insuff.</label>
      <label><input type="radio" name="voto" value="6" />suff.</label>
      <label><input type="radio" name="voto" value="7" />buono</label>
      <br />
      <label>Commento: <textarea name='commento'></textarea></label><br />
      <label>La tua e-mail: <input type='text' name='email' /></label><br />
      <input type='submit' value='Spedisci' />
    </fieldset>
  </form>
```

Così com'è, il modulo funziona già. Però, se il visitatore lo compila e dimentica di scrivere il suo indirizzo email, Sar-At non lo accetta (perché noi gli abbiamo detto che la email è obbligatoria) e ripresenta il modulo bianco. I commenti già scritti dal visitatore sarebbero persuti: una bella seccatura, per lui.

Preoccupiamoci allora di inserire le istruzioni Sar-At per migliorare l'esperienza del nostro visitatore. Innanzitutto, inseriremo i segnaposto Sar-At in modo che quando il modulo si ripresenta (in caso di errore di compilazione) i commenti già inseriti non vengano persi. È molto semplice, dobbiamo solo fare un cambiamento in due righe:

```
<label>Commento: <textarea name='commento'>{commento}</textarea></label><br />
<label>La tua e-mail: <input type='text' name='email' value='{email}' /></label>
```

Il risultato è molto migliorato ma non è ancora perfetto. I pulsanti radio verrebbero azzerati a ogni nuova riapparizione del modulo, così:

Voto:  insufficiente  sufficiente  buono

### Radiobutton

Per assegnare un valore iniziale ai pulsanti radio e per ricordare il valore assegnato dal visitatore in caso il modulo si presenti più di una volta (in caso di errore di compilazione) lasceremo che sia showForm a generarli. Cancelliamo questo codice HTML:

```
<label><input type="radio" name="voto" value="5" />insufficiente</label>
<label><input type="radio" name="voto" value="6" />sufficiente</label>
<label><input type="radio" name="voto" value="7" />buono</label>
```

Rimpiazziamolo con questo:

```
<label>{RADIOBUTTON voto 5}insufficiente</label>
<label>{RADIOBUTTON voto 6}sufficiente</label>
<label>{RADIOBUTTON voto 7}buono</label>
```

Che differenza fa? Un mondo di differenza. Alla prima apparizione, il visitatore vedrà:

Voto:  insufficiente  sufficiente  buono

Sar-At ha preselezionato il voto “sufficiente” perché in fase di definizione dei campi noi gli abbiamo detto che questo è il default. Se il visitatore clicca “buono”, inserisce un commento, dimentica di scrivere la e-mail e preme “Spedisci”, allora il Sar-At ricorda tutte le sue scelte e gli ripresenta il modulo per la correzione, senza costringerlo a reintrodurre nulla.

Per completare la creazione di una esperienza perfetta dobbiamo presentare al visitatore un messaggio d'errore significativo e posizionato nel punto esatto in cui l'errore avviene.

### Onerr

Il blocco ONERR viene inviato al visitatore se e solo se ha commesso un errore di compilazione. Dobbiamo specificare il nome del campo da controllare. Per esempio, potremmo avere:

```
E-mail: <input type='text' name='email' value='{email}' /><br />
```

{ONERR email}L'indirizzo di posta elettronica è obbligatorio!{ENDERR email}  
 Per esempio, se il visitatore compila il modulo come nella figura precedente e preme Spedisci, vedrà:

Naturalmente il contenuto di ONERR può venire personalizzato a piacere. Possiamo far apparire il messaggio di errore in rosso, aggiungere una icona, persino includere un suono... qualsiasi cosa sia consentita in HTML è valida. Per esempio, potremmo colorare di rosso il campo non compilato:

La tua e-mail: `<input type='text' name='email' value='{email}' {ONERR email} style='background: red;' {ENDERR email} />`

Disgraziatamente, non tutti i browser supportano la colorazione degli elementi di un modulo (la schermata impeccabile qui sopra è presa da Firefox 0.9: Internet Explorer 5,5 e Safari 1.2 non mostrano il fondo rosso).

È possibile avere più blocchi ONERR relativi al medesimo campo in una pagina mastro ma solo se sono collocati su righe differenti della pagina mastro. Due ONERR sulla stessa riga non sono supportati.

### Selectedif

La istruzione SELECTEDIF è analoga a RADIOBUTTON visto sopra, ma viene usato per i menu (le istruzioni "select" dello HTML). Cambiamo il nostro esempio, così:

```
Voto: <select name='voto'>
      <option value='5'>insufficiente</option>
      <option value='6'>sufficiente</option>
      <option value='7'>buono</option>
</select><br />
```

Anche in questo caso, se il visitatore fa una scelta dal menu e poi invia il modulo senza completarlo, allora perde la selezione. Possiamo rimediare, trasformando il codice visto sopra inserendo l'istruzione Sar-At SELECTEDIF:

```
Voto: <select name='voto'>
      <option {SELECTEDIF voto 5}>insufficiente</option>
      <option {SELECTEDIF voto 6}>sufficiente</option>
      <option {SELECTEDIF voto 7}>buono</option>
</select><br />
```

Quando Sar-At trova l'istruzione SELECTEDIF la sostituisce con il codice HTML che seleziona l'opzione indicata se il campo ha il valore specificato. Nel nostro esempio, il modulo verrà rappresentato alla prima apparizione così:

Modifica struttura della sezione *paypal\_transaction* ?

Nome campo	visible
Note	Visibile ai visitatori?
Tipo	<input type="text"/>
Opzionale	Può venire lasciato in bianco? <input checked="" type="checkbox"/> Deve essere unico <input type="checkbox"/>
Massimo	<input type="text"/>
Minimo	<input type="text"/>
Default	No
Validazione	<input type="text"/>

Aggiorna

**Come valuti questo prodotto?**

Voto

Commento:

La tua e-mail:

Spedisci

Il valore “sufficiente” è pre-selezionato perché nel creare la sezione abbiamo detto a Sar-At che il valore di default di “voto” è sei.

SELECTEDIF può venire utilizzato in combinazione con una Vista. Immaginate di avere una sezione che contiene tutti i personaggi di Walt Disney. Voi volete creare un menu prendendo tutti i nomi e li presenta dentro un menu. Create una vista che seleziona tutti i record e chiamatela “disney”, poi scrivete:

```
<select name='disney'>{BEGIN disney}<option {SELECTEDIF disney {id}}>
  >{nome_personaggio}</option>{END disney}</select>
```

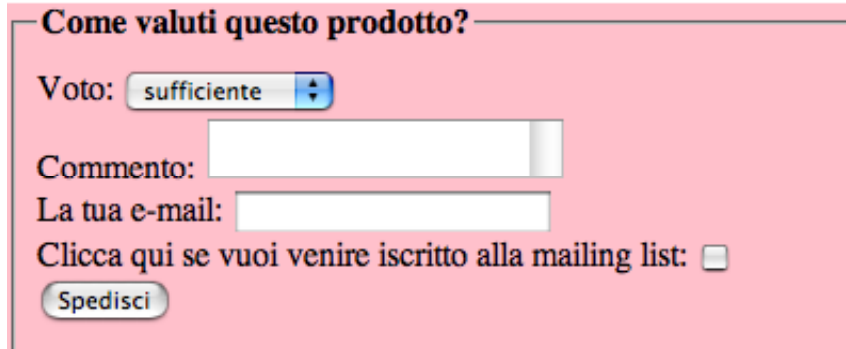
### Checkbox

CHECKBOX completa la gestione della selezione utente in Sar-At. Concettualmente si tratta di un parente stretto di RADIOBUTTON e SELECTEDIF visti poco sopra. La sintassi è semplicemente {CHECKBOX nomecampo}

Modifichiamo ancora il nostro modulo rosa per fare un esempio. Aggiungendo lo HTML:

```
<label>Clicca qui se vuoi venire iscritto alla mailing list:  
  <input type='checkbox' name='mailing' />  
</label>
```

L'aspetto del modulo diventa:



Per consentire a showForm di ricordare lo status della casella di selezione “mailing list”, il codice diventa:

```
<label>Clicca qui se vuoi venire iscritto alla mailing list:  
  {CHECKBOX mailing}  
</label>
```

## Upload

Con la stessa semplicità con cui voi potete usare Sar-At per caricare un documento sul server, è possibile consentire ai vostri visitatori di fare lo stesso. Il visitatore registrato del sito può caricare una foto nel proprio profilo (e realizziamo una community), o il partecipante a un concorso può mandare il suo elaborato, o chi scrive al modulo “contattaci” del sito può allegare il suo curriculum vitae. ”,

### Nelle sezioni modulo ordinarie

La vostra sezione modulo deve contenere un capo di tipo file, naturalmente. Questo permette a Sar-At di sapere in quale posizione sul disco rigido vadano registrati i documenti dei visitatori. Immaginiamo di aver creato una sezione modulo chiamata “autoregistrazione” e dentro di essa abbiamo messo un campo di nome “autoscatto” per consentire ai visitatori di pubblicare il proprio primo piano.

Nel vostro HTML dovrete aggiungere questa riga di codice:

```
<input type='file' name='image_autoscatto' />
```

In altre parole, dovete usare come name dell’input la parola image\_ seguita dal nome del campo.

### Nelle sezioni modulo aggiornabili

Immaginiamo adesso che sia necessario per un visitatore di sostituire la sua foto con una versione più recente. Se volete consentirgli questa manovra, la vostra sezione modulo deve essere dichiarata aggiornabile, come è spiegato in questo stesso capitolo poche pagine indietro.

Lo HTML dovrà contenere due righe, così:

```
<input type='file' name='image_autoscatto' />
```

```
<input type='hidden' name='autoscatto' value='{autoscatto}' />
```

La prima riga consente al visitatore di caricare una nuova foto. La seconda riga memorizza il nome della foto precedente in modo che venga conservato se il visitatore cambia altre informazioni (per esempio, il suo indirizzo email) ma non questa.

### Consultare il materiale dei visitatori

Abbiamo spiegato come il Sar-At possa spedire al vostro indirizzo di posta elettronica tutto il contenuto di tutti i moduli compilati dai visitatori.

Se il visitatore carica documenti nel compilare il modulo, il messaggio email che vi raggiunge contiene un link seguendo il quale potrete scaricare i documenti.

### Mantenere riservato il materiale dei visitatori



Anche il materiale caricato dai visitatori può venire registrato da Sar-At in una zona del disco rigido del server inaccessibile via web. Si usa il meccanismo di `showFile.php`, che è spiegato nel capitolo sei di questo libro.

### Un esempio: il forum

Come abbiamo già spiegato in questo stesso capitolo, una sezione modulo è utilizzabile anche come sorgente di pagine Sar-At. Vediamo, per esempio, come costruire un semplice forum. Tutti i contributi dei visitatori vengono aggiunti l'uno di seguito all'altro.

Benvenuti nel nostro forum.

```
<table>
<tr>
  <th>
    Data
  </th>
  <th>
    Numero IP
  </th>
  <th>
    Messaggio
  </th>
</tr>
{BEGIN forum}
<tr><td>
  {lastMod}
</td>
<td>
  {authorId}
</td>
<td>
  {testo}
</td></tr>
{END forum}
</table>
<hr />
<p>Inserisci qui il tuo contributo:</p>
  <form action="showForm.php" method="post" name="myDemoForm">
    <input type="hidden" name="template" value="forum" />
    <textarea name="testo" rows="5" cols="30"></textarea>
    <br />
    <input type="submit" value="OK" accesskey="0" />
  </form>
```

Per far funzionare l'esempio sono necessarie una sezione modulo, chiamata `forum`, che contiene solo un campo chiamato "testo" di tipo Testo Lungo. La pagina mastro riportata qui sopra è sia la pagina mastro della sezione, sia la pagina destinazione del modulo. La vista "forum" si limita a riportare tutti i record, senza filtro, in ordine di data.

### Il blog

Dal menu *Pagine*, la voce *Aggiungi modulo...* vi permette di caricare nel vostro sito soluzioni precotte composte da sezioni, viste e HTML pronte per l'uso. Uno di questi moduli è il blog. Se proverete a caricarlo vedrete che esso è composto di due tabelle: una per i post dell'autore del blog, l'altra per i commenti dei visitatori. Questi ultimi vengono ricevuti con `showForm`.

È interessante notare che il meccanismo realizzato in quel caso combina l'uso di un Portale (si veda in merito il capitolo 2). Quando stiamo mostrando il post numero X del blog con `showPage` (sarebbe lo stesso con `showTwig`), mostriamo il record con id X della tabella blog, e aggiungiamo nella sua pagina mastro uno HTML simile a questo:

```
<form action='/showForm.php' method='post'>
  <input name="template" value="comments" type="hidden" />
  <input type="hidden" value="{id}" name="id_post_commentato" />
  <label for="title">Titolo</label></br>
  <input name="title" id="title" type="text" /></br>
  <label for="reply">Commento</label></br>
  <textarea id='reply' title="reply"></textarea></br>
  <input type="submit" value="Invia" />
</form>
```

In questo modo il campo “id\_post\_commentato” della tabella commenti conterrà automaticamente l’id del pst, e ciò consente al Portale di funzionare.

### Moduli: risposte a domande frequenti

#### Campi nascosti

*Oltre alla compilazione da parte dell’utente, Sar-At accetta all’interno dei moduli i campi nascosti??*

Sì. In questo caso alcuni campi della sezione si autocompilano. Ovviamente, tutte le inserzioni compilate da tutti i visitatori conterranno lo stesso valore. Per esempio, in una sezione che contiene il campo chiamato “sorgente”, tutti i contributi dei visitatori conterranno il valore “web” se nel modulo HTML io scrivo:

```
<input type="hidden" name="sorgente" value="web" />
```

Questa metodologia può essere piuttosto utile in diversi casi. Per esempio, se abbiamo un sito multilingue, potremmo scrivere nel modulo in italiano:

```
<input type="hidden" name="language" value="1001" />
```

Nel modulo in inglese inseriremmo invece:

```
<input type="hidden" name="language" value="1002" />
```

In questo modo, i contributi in italiano e quelli in inglese verranno registrati nella medesima sezione ma ciascuno sarà correttamente etichettato.

Nel manuale *Mantenere il sito* troverete un elenco esaustivo dei codici di lingua.

Un’altra possibilità: inviare i visitatori del nostro sito che hanno prenotato un acquisto al sito di un albergo partner, comunicando attraverso i campi nascosti i dati come il nostro codice venditore, in modo da incassare la provvigione.

#### Campi incorporati di Sar-At e valori di default

*Posso chiamare “title” uno dei campi del modulo, in modo che quello che scrive il mio utente finisca proprio in quel campo??*

Certamente, è anzi una ottima idea. In generale, tutti i campi predefiniti di Sar-At possono venire utilizzati nei moduli, tanto quanto i campi definiti dall’utente, con l’eccezione di *id*. Tutti sono considerati opzionali e hanno un valore di default. Se il campo non viene compilato o autocompilato, assume il valore riportato nella tabella seguente:

Campo	Valore
Id	<i>Un numero di serie autocompilato</i>
masterId	Zero
lastMod	<i>Data di compilazione del modulo</i>
AuthorId	<i>Numero IP del visitatore o nome del visitatore se utente registrato</i>
ModifierId	<i>Numero IP del visitatore o nome del visitatore se utente registrato</i>
Language	1001 (italiano Unicode)

Campo	Valore
Visibile	Y (sì)
SaratPremium	o (Non riservata, leggibile da chiunque)
Title	Modulo senza titolo
Keywords	<i>(nessuna)</i>
Description	<i>(nessuna)</i>
Creation	<i>Data di compilazione del modulo</i>
Hit Counter	Disabilitato
masterPage	<i>(pagina mastro di default di questa sezione)</i>
saratUrl	<i>Indirizzo amichevole della pagina</i>

### Notifiche multiple

*Quando definisco un modulo posso indicare una casella mail che riceve notifiche ogni volta che il modulo viene compilato. Ma se desidero che le notifiche arrivino a più di una casella delle nostre?*

Potete indicare più caselle, separate da virgole.

### Trasformazione

*Posso trasformare una sezione modulo in sezione ordinaria e viceversa?*

Sì. A partire dalla versione 11.0 di Sar-At si possono trasformare le sezioni modulo in sezioni non-modulo e viceversa, usando la linguetta *Proprietà* della sezione. Attenzione, però, le sezioni modulo create in questo modo sono automaticamente *a massima sicurezza* -- in pratica: una sezione modulo creata con un vecchio Sar-At; che non è ancora stata posta in condizioni di massima sicurezza; e che trasformiamo prima in sezione non-modulo e poi nuovamente in sezione modulo non torna alle condizioni iniziali, ma diventa a massima sicurezza.

# Parte quarta: gestione eventi

Il sistema di gestione eventi è una tecnologia molto potente che permette di creare con Sar-At siti sino a ieri inimmaginabili, come aste, *wiki* in cui ciascun membro può intervenire eppure l'autore originale viene immediatamente informato che c'è stata una modifica, *liste di discussione* in cui gli interventi di ciascuna persona vengono spediti per email a tutte le altre, comunità e altro ancora.



L'idea è che il programma può automaticamente reagire a eventi prestabiliti. Gli eventi sono di basso livello (per esempio: è stata cancellata una pagina) o di alto livello (per esempio: è stato acquistato un articolo nel commercio elettronico da un visitatore registrato). Le reazioni possibili sono molto flessibili e possono toccare il database, il sito e la newsletter.






Qualche esempio di eventi possibili in un sito.

1. Il sito è un commercio elettronico. A ogni cinquecentesimo acquisto, il proprietario del sito riceve una email con le generalità dell'acquirente che riceve un omaggio.
2. Il sito è un quotidiano. Ogni sera, parte automaticamente una newsletter impaginata in HTML a tutti gli abbonati
3. Il sito è un blog. Ogni volta che il proprietario del blog aggiunge una bloggata, viene automaticamente e invisibilmente generato un gruppo di lettori interessati a quell'argomento. Chiunque visiti la pagina può iscriversi al gruppo con un clic. Ogni volta che qualcuno aggiunge un commento la pagina, tutti i membri del gruppo ricevono una email con il contenuto del commento.

## Creazione di un evento

La gestione eventi si raggiunge dalla omonima voce del menu *Sito*. Appare una tabella con tutti gli eventi definiti: possiamo editarne uno esistente oppure aggiungerne uno nuovo cliccando il pulsante in fondo.

Eventi ?				
Nome	Evento	Sezione	Risposta	Distruggi
Rimuovi il "senza titolo" che l'impo	insert	paragrafi	trigger	
Sconto al millesimo acquirente	sale	applicazioni	php	

10    Pagina 1 su un totale di 1   

Vogliamo fare una prova? Create un gruppo fittizio di visitatori registrati. Nel menu *Utilizzatori* selezionate la voce *Visitatori registrati* | *Aggiungi gruppo*. Scegliete un nome facilmente ricordabile, come "Prova eventi" e una qualsiasi sezione del vostro sito (o più d'una).


Nella schermata che appare, cliccate il nome della sezione che avete associato al gruppo: appare l'elenco (vuoto) dei membri del vostro gruppo. Aggiungete voi stessi, in modo da poter fare qualche inserimento. Nel campo *nome* inserite il vostro indirizzo email, e digitate una password qualsiasi (tanto, non ci servirà). Prendete una data di scadenza arbitrariamente nel futuro.

Ora possiamo aggiungere un evento per capire come funziona. Nel menu *Sito* scegliete "Gestione eventi" e cliccate il pulsante "Aggiungi evento". Cominceremo creando il più semplice degli eventi: una

email avvisa tutti i membri di un gruppo (nel nostro esempio, soltanto tu) quando viene aggiunta una pagina nella sezione indicata.

#### Aggiunta di un evento

Nome evento:

L'evento avviene quando... 

... una pagina (record) viene inserita nel sistema [INSERT]

...un gruppo di pagine (record) viene importato da un file [BULK-INSERT]

... una pagina (record) esistente viene editata e modificata [UPDATE]

... un amministratore cancella una pagina (record) esistente [DELETE]

...un gruppo di pagine (record) viene cancellato o la sua sezione distrutta [BULK-DELETE]

...una vendita ecommerce viene chiusa con successo [SALE]

... viene richiamato lo script sarat/sar-at\_evtmgr.inc.php [USER]

all'interno della sezione:  e solo se viene manipolata la pagina (record) numero:  .

Scrivi zero se vuoi che l'evento accada per qualsiasi pagina (record).

...una pagina non può venire generata e viene mostrata la pagina d'errore [ERROR]

### Tipi di evento

Sar-At riconosce due tipologie di evento: quelli collegati al database (per esempio, arriva un amministratore e cancella una pagina) e quelli collegati soltanto alle azioni dei visitatori (per esempio, un visitatore registrato clicca un pulsante speciale sul sito).

Un evento sul database può venire scatenato in molti modi differenti. Una pagina, per esempio, può venire inserita in molti modi: può succedere quando un amministratore va nel menu *Pagine* di Sar-At e clicca *Aggiungi pagina* ma anche quando un altro amministratore importa dal pulsante *Importa pagine*. Se la sezione interessata è una sezione modulo, anche i visitatori del sito possono aggiungere pagine, semplicemente compilando il modulo. Quindi gli eventi di tipologia database possono accadere anche in risposta alle azioni dei visitatori, quando nessun amministratore è connesso a Sar-At.

Tipologia	Nome	Scatta quando...
Database	INSERT	Una pagina viene creata o importata per la prima volta nel database, da un amministratore o da un visitatore. Notate che l'evento scatta anche se la pagina è invisibile ai visitatori.
Database	UPDATE	Una pagina esistente viene editata e aggiornata da un amministratore. Nelle sezioni modulo aggiornabili, anche i visitatori registrati possono aggiornare una pagina: la loro.
Database	DELETE	Una pagina esistente viene rimossa da un amministratore
Visitatori	USER	Un visitatore accede allo speciale indirizzo sarat/sar-at_evtmgr.inc.php
Visitatori	ERROR	Un visitatore accede a un indirizzo Sar-At non esistente, per esempio showPage.php?template=inesistente
Visitatori	SALE	Un visitatore completa un acquisto attraverso il sistema di commercio elettronico integrato

Tipologia	Nome	Scatta quando...
Database	BULK-DELETE	Un amministratore cancella un gruppo di pagine (record). Questo attualmente può accadere in due casi. Primo, quando il proprietario del sito distrugge una intera sezione del sito stesso. Secondo, quando un amministratore importa nuove pagine (record) da un file esterno e nel farlo sovrascrive o distrugge molte pagine esistenti

Gli eventi collegati al database scattano automaticamente, senza che si debba fare alcuna lavorazione sullo HTML del sito.

Quando un redattore lavora su un sito multilingue e cancella una traduzione in una lingua, il sistema automaticamente si offre di cancellare tutte le altre traduzioni. In questo caso viene mandato un evento delete per ciascuna localizzazione, non un bulk-delete per tutte.

Gli eventi USER vanno fatti scatenare attraverso un eccesso allo speciale indirizzo

/sarat/sar-at\_evtmgr.inc.php

Ecco un esempio in HTML:

```
<a href='sarat/sar-at_eventmgr.inc.php?template=prod&id={id}
&amp;returnURL={sarelURL}'>
Ricevi una mail quando la scheda prodotto viene aggiornata
</a>
```

La tabella successiva presenta i tre parametri richiesti da sar-at\_evtmgr.inc.php (sono tutti e tre obbligatori e possono venire passati indifferentemente in modalità GET o i modalità POST).

Nome	Significato
Id	Identificatore numerico dell'evento. Deve essere un numero intero
Template	Sezione collegata all'evento. Deve essere il nome di una sezione effettivamente presente nel menu Pagine di Sar-At
returnURL	Indirizzo della pagina da mostrare al visitatore in risposta al clic. Può essere sia l'indirizzo di una pagina HTML statica (come grazie.html) sia una pagina sintetizzata da Sar-At (come showPage.php?template=xyz). Se desiderate che il visitatore resti nella pagina corrente potete scrivere {sarelURL}

Ricordate che {sarelURL} non funziona dentro showForm

In ciascun caso, gli eventi possono venire filtrati e parcellizzati. Cioè, possiamo creare un evento che scatta ogni volta che viene editata una qualsiasi pagina della sezione "prodotti" oppure un evento specifico per il prodotto 23 e uno ben diverso per il prodotto 24. Nel primo caso basta indicare lo id zero.

all'interno della sezione:  e solo se viene manipolata la pagina (record) numero: .  
 Scrivi zero se vuoi che l'evento accada per qualsiasi pagina (record).

### Reazioni possibili a un evento

Reazione	Parametri	Note
MAILTO	Testo della email; gruppo di visitatori registrati destinatario.	Per gli eventi INSERT e UPDATE, Sar-At aggiungerà automaticamente alla email l'indirizzo della pagina web appena aggiunta o aggiornata.

Reazione	Parametri	Note
NEWSLETTER	Indirizzo di una pagina HTML da spedire; gruppo di visitatori registrati destinatario.	L'indirizzo può essere dinamico. Per esempio <code>showPage.php?template=news</code> L'idea è di creare una newsletter che parte automaticamente quando accade qualcosa – magari quando un nuovo prodotto appare a catalogo.
ADDGROUP		Viene creato un nuovo gruppo di visitatori registrati. Vedi anche il paragrafo “gruppi creati automaticamente” poco più avanti. Questa reazione è appropriata alla creazione di una pagina; l'idea è di creare gruppi di interessati a un prodotto, a un cantante, a una voce di blog, insomma a una singola pagina del vostro sito.
SUBSCRIBE	Gruppo a cui abbonare il visitatore	Ha senso solo per visitatori autoregistrati al sito. Permette loro di abbonarsi a una newsletter.
UNSUBSCRIBE	Gruppo da cui rimuovere il visitatore	Ha senso solo per visitatori autoregistrati al sito. Permette loro di revocare una iscrizione.
PHP	URL dello script da eseguire	Può venire anche eseguito uno script su server remoto: basta indicare la URL completa. La chiamata avviene in modalità GET.
TRIGGER	Codice SQL	Richiede MySQL 5.0 o superiore sul server. Permette l'uso della sintassi trigger di MySQL descritta in <a href="http://dev.mysql.com/doc/refman/5.0/en/create-trigger.html">http://dev.mysql.com/doc/refman/5.0/en/create-trigger.html</a> I trigger creati da Sar-At sono tutti del tipo BEFORE, in modo da consentire al trigger di ispezionare un record editato prima che venga cancellata la vecchia versione.

Chi fosse interessato al caso PHP, veda la descrizione di `SaratVerifyCallback()` nel seguito

Vediamo due casi tipici di un blog: l'iscrizione a un gruppo di interesse e la disiscrizione.

Per la disiscrizione, immaginiamo che il visitatore sia nella pagina relativa a un argomento (la pagina X del blog, tanto per capirci) e clicca un pulsante “Revoca iscrizione alla mailing”.

Il pulsante potrà avere questo semplice codice HTML:

```
<a href='sarat/sar-at_eventmgr.inc.php?template=prod&id={id}
&amp;returnURL={sarelURL}'>
Ricevi una mail quando la scheda prodotto viene aggiornata
</a>
```

#### Log degli eventi

Sar-At mantiene internamente un log di basso livello degli eventi che si sono scatenati. Questo log può venire consultato, se necessario, premendo il pulsante in basso a destra di questa schermata. Il log può venire cancellato da un amministratore di server usando il menu *Server*.

#### Gruppi creati automaticamente

Un evento può creare un gruppo di visitatori registrati. Torniamo al nostro esempio del blog. Immaginate di aver scritto un sito per un famoso *blogger*. Alcuni fan vorranno venire avvisati quando appare un nuovo commento (per esempio, chi scrive una opinione forte gradisce sapere quando qualcun altro commenta sul medesimo argomento). Potreste dedicare una pagina web a ogni intervento del

*blogger* e chiedere a Sar-At di creare automaticamente un gruppo per ogni pagina del genere. I fan già registrati potranno iscriversi al gruppo con un sol clic e venire avvisati automaticamente quando un nuovo commento appare in questa pagina. Anche in questo caso lo HTML è banale:

```
<a href='sarat/sar-at_eventmgr.inc.php?template=blog&id={id}
&returnURL={sarelURL}'>
```

Il numero di gruppi automaticamente creato in un sito può diventare molto grande. Per questo motivo, Sar-At nasconde normalmente i gruppi automatici quando deve mostrare quelli esistenti (per esempio, nel menu Visitatori | Vistatori registrati). Dove ha senso, Sar-At mostra una casella di spunta chiamata “mostra anche i gruppi creati automaticamente”, cliccando la quale appaiono i gruppi in più.



# Parte quinta: plug-in per mastro classiche

## Introduzione ai plug-in

Sar-At è, in sostanza, un meccanismo che sposa dinamicamente i contenuti di una base dati con pagine mastro per produrre un sito web. Occasionalmente però il sito web ha bisogno di informazioni che non possono venire stipate nella base dati (per esempio: il valore di una azione in Borsa, con aggiornamento istantaneo) oppure le informazioni della base dati vanno manipolate prima di venire servite nella base dati (per esempio, disegnando una immagine PNG con i risultati di un sondaggio che i visitatori hanno compilato).

Per risolvere questo tipo di problematiche, Sar-At supporta il concetto di plug-in. Analogamente a quanto avviene in altri popolari programmi come per esempio Photoshop, Quark XPress o i browser, un plug-in per Sar-At è un pezzo di codice sviluppato da un programmatore professionista che estende il sistema e le sue capacità.

Se per lo sviluppo del vostro sito vi è necessaria una funzionalità avanzata che non è possibile realizzare altrimenti, contattate il personale tecnico di Accomazzi.net. La documentazione sullo sviluppo di plug-in in linguaggio PHP viene fornita su richiesta, ma Accomazzi.net si riserva il diritto di modificare queste specifiche in qualsiasi momento, per migliorare le prestazioni e le funzionalità del Sar-At.

Sar-At ammette cinque distinti tipi di plug-in: quelli per showPage, quelli Twig, quelli di Vista, quelli di compilazione e quelli di validazione (per i moduli). Il resto di questo capitolo spiega in cosa consistono i plug-in per showPage e presenta i plug-in standard incorporati e forniti di serie con ogni copia di Sar-At. Gli altri plug-in sono presentati nel seguito del manuale.

## I plug-in per pagine mastro classiche

I plug-in classici (chiamati anche *plug-in di showPage*, perché si attivano dall'interno di quel programma) si usano in una pagina mastro esattamente come i campi della sezione corrente. Basta scrivere il nome del plug-in circondato da parentesi graffe.

Il nome dei plug-in è sensibile alla differenza tra lettere minuscole e maiuscole. Se il vostro sito dispone di un plug-in chiamato "esempio" e voi inserite in una pagina mastro il segnaposto {Esempio}, la pagina mastro non verrà correttamente prodotta e visualizzata.

I plug-in disponibili con ogni copia del Sar-At si distinguono in incorporati e standard. I primi vengono sintetizzati dal Sar-At stesso, sono irrinunciabili e non appaiono nell'elenco in prima pagina. I secondi invece sono semplicemente pre-installati e volendo possono venire rimossi o alterati.

Il resto di questo paragrafo presenta i plug-in forniti di serie con Sar-At, insieme a trucchi e consigli di utilizzo.

Per ciascun plug-in standard riportiamo una breve descrizione, seguita da quattro paragrafi. DEFINISCE riporta tutti i segnaposto utilizzabili grazie a quel plug-in. PARAMETRI riporta le informazioni che si possono o debbono passare al plug-in per configurarlo – tipicamente attraverso un FORM dello HTML. Lo ESEMPIO infine mostra un brano di HTML dove il plug-in viene dimostrato.

### Plug-in intestazione

Il plug-in di intestazione {HEAD} non è utile né necessario in HTML5. Continua a funzionare in Sar-At 11 ma l'uso è deprecato. Il codice che genera HEAD verrà rimosso in Sar-At 12.

### Plug-in visitatore registrato

Il plug-in va utilizzato per testare se un visitatore va parte di un gruppo di utenti registrati (o autoregistrati, nel caso del commercio elettronico).

Quando il visitatore guarda una pagina riservata, Sar-At richiede e convalida il suo nome utente e la sua password, che un amministratore avrà precedentemente inserito nel sistema usando la schermata Utenti e Gruppi.

Il plug-in incorporato standard {sarel\_username} può venire utilizzato per far inserire al Sar-At il nome del visitatore nella pagina mastro, in modo da salutarlo.

L'uso di questo plug-in rende la pagina incompatibile con la cache di Sar-At, e quindi molto meno veloce ad apparire. Questo è inevitabile, perché il nome dell'utente corrente cambia per ogni utente, e quindi ogni copia della pagina inviata a ciascun distinto visitatore è differente...

#### DEFINISCE

{sarel_userlogged}	Vero oppure falso, a secondo che il visitatore corrente sia registrato o meno
{sarel_username}	Identificatore univoco del visitatore. Username non modificabile
{sarat_adminlogged}	Vero oppure falso, a secondo che il visitatore corrente sia un amministratore
{sarat_adminname}	Identificatore univoco dell'amministratore connesso al server

#### PARAMETRI

Nessuno.

#### ESEMPIO HTML

All'interno di una pagina mastro classica scriviamo:

```
{IF sarat_adminlogged}
  informazioni privilegiate
{ENDIF sarat_adminlogged}
```

O anche:

```
{IF sarel_userlogged}
  Benvenuto, {sarel_username}.
{ELSE sarel_userlogged}
  Benvenuto, visitatore anonimo
{ENDIF sarel_userlogged}
```

### Plug-in motore di ricerca incorporato

Il plug-in {search\_engine}, permette la realizzazione di un motore di ricerca che lavora su tutte le sezioni del sito. È pienamente personalizzabile dal progettista del sito (che può anche decidere di lasciare che sia il visitatore a selezionare le opzioni di ricerca).

Come sempre, i risultati della ricerca possono venire impaginati visivamente con la massima libertà.

#### DEFINISCE

Tutti i campi dei record trovati, per nome. Per esempio, {title} è il titolo della pagina trovata e {lastMod} la data dell'ultima modifica.

{search\_engine} I risultati della ricerca.

{searchengine\_next} URL per la schermata successiva di risultati. Vedi esempi nel prossimo paragrafo

{searchengine\_prev} URL per la schermata precedente di risultati. Vedi esempi nel prossimo paragrafo

{searchengine\_usenext} Se esiste una schermata precedente di risultati, vale 1, altrimenti vale 0  
 {searchengine\_useprev} Se esiste una schermata precedente di risultati, vale 1, altrimenti vale 0  
 {searchengine\_row} Vale odd nelle righe dispari e even nelle righe pari del risultato. Questo parametro può venire utilizzato per impaginare i risultati utilizzando i Cascading Style Sheet. Vedi sotto per un esempio.

{searchengine\_resultsfound} Mostra quante pagine sono state trovate.

{searchengine\_fromto} Mostra da quale record a quale record stiamo mostrando nella paginata corrente. Per esempio, se il motore di ricerca ha trovato 51 pagine e se recordsReturned vale 10 e se stiamo guardando la seconda pagina di risultati, allora {searchengine\_fromto} vale 11-20.

{searchengine\_navigator} Mostra un navigatore interno. Per esempio, se il motore di ricerca ha trovato 51 pagine e se recordsReturned vale 10 e se stiamo guardando la seconda pagina di risultati, allora {searchengine\_navigator} produce questo risultato:

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#)

Il tag {search\_engine} si deve trovare dentro la pagina mastro (dei risultati). Tutti gli altri tag devono trovarsi in un documento distinto, chiamato documento ihtml, che contiene le regole di formattazione dei risultati in linguaggio HTML.

#### PARAMETRI

Nome parametro	Valori accettabili	Spiegazione
search	un testo qualsiasi	Cosa cercare
comparison	equals differs contains gt lt gte lte fulltext	dev'essere uguale a dev'essere diverso da deve contenere deve essere maggiore deve essere minore deve essere maggiore o uguale deve essere minore o uguale La ricerca avviene <i>a tutto testo</i>
searchFld1 searchFld2 searchFld3 ...	Nome di un campo	primo campo in cui cercare secondo campo in cui cercare ...
recordsReturned	Numero intero	numero record per pagina da restituire
sortBy	Numero intero	Numero del campo da usare come chiave d'ordinamento
sortOrder	ascending descending	ordinamento A..Z ordinamento Z..A
template	Sezione entro la quale effettuare le ricerche	Parametro obbligatorio, ma viene ignorato se è presente il parametro multiSectionSearch
results	Nome di un file di testo con brano HTML dell'interfaccia utente.	Se questo parametro manca, il motore di ricerca si aspetta per default un file chiamato results_searchEngine.ihtml
noresults	Nome di un file di testo con brano HTML da mostrare se nulla viene trovato	Se questo parametro manca e nulla viene trovato appare la scritta <i>La tua ricerca non ha ottenuto alcun risultato / No results found.</i>

Nome parametro	Valori accettabili	Spiegazione
multiSectionsSearch[]	Elenco di sezioni nelle quali effettuare la ricerca	Opzionale. Se manca, la ricerca viene effettuata nella sola sezione indicata dal parametro template. Vedi esempio alla fine di questo sottocapitolo.
creation	Un numero positivo	Opzionale. Se presente, il motore di ricerca non mostra pagine più vecchie di questo numero di giorni.
groupby	Il nome di un campo	Opzionale. Se presente, il motore di ricerca raggruppa nei risultati tutti i record che hanno identico valore in questo campo

Se il parametro “multiSectionsSearch” ha un valore, Sar-At cerca in tutti i campi che esistono in tutte le sezioni indicate.

Se il parametro “comparison” ha valore uguale a “fulltext” e se “multiSectionsSearch” non viene passato, search\_engine esegue una ricerca a tutto testo (di cui abbiamo parlato nel capitolo 2) sulla sezione indicata dal parametro template.

All'interno del file HTML, si utilizza l'indicazione one\_foundpage\_block per caratterizzare il blocco di HTML da ripetere per ogni pagina trovata.

#### ESEMPIO HTML

Questo codice HTML realizza un motore di ricerca che va a cercare il testo digitato nel titolo delle pagine della sezione prodotti.

```
<form action="showPage.php" method='post'>
  Cerca:<input type='text' name='search' value="" size='12' />
  <input type='submit' value="Trova" />
  <input type='hidden' name='template' value='prodotti' />
  <input type='hidden' name='masterPage' value='foundResult.html' />
  <input type='hidden' name='comparison' value='contains' />
  <input type='hidden' name='searchFld1' value='title' />
  <input type='hidden' name='recordsReturned' value='10' />
  <input type='hidden' name='sortBy' value='4' />
  <input type='hidden' name='sortOrder' value='descending' />
</FORM>
```

Questo codice HTML realizza un modulo di ricerca più complesso, dove l'utente può selezionare i parametri richiesti e generare una ricerca complessa:

```
<form action="showPage.php" method='post'>
  <input type='hidden' name='template' value='prodotti' />
  <input type='hidden' name='masterPage' value='foundResult.html' />

  Cerca i prodotti in cui
  <select name='searchFld1'>
    <option selected value='title'>il nome</option>
    <option value='description'>la scheda</option>
    <option value='prezzo_lit'>il prezzo</option>
  </select>
  <select name='comparison'>
    <option selected value='contains'>contiene</option>
    <option value='equals'>&egrave; uguale a</option>
    <option value='differs'>&egrave; diverso da</option>
```

```

        <option value='gt'>&grave; maggiore di</option>
        <option value='gte'>&grave; maggiore o uguale a</option>
        <option value='lt'>&grave; minore di</option>
        <option value='lte'>&grave; minore o uguale a</option>
    </select>
    <input type='text' name='search' value="" size='12' />

    <p>Mostra i risultati a gruppi di
    <select name='recordsReturned'>
        <option>5</option>
        <option selected>10</option>
        <option>25</option>
    </select>
    ordinandoli per
    <select name='sortBy'>
        <option selected value='title'>nome</option>
        <option value='lastMod'>data di aggiornamento</option>
        <option value='prezzo_lit'>prezzo</option>
    </select>
    in modo
    <select name='sortOrder'>
        <option selected value='ascending'>crescente</option>
        <option value='descending'>decrescente</option>
    </select></p>
    <p><input type='submit' value="Trova" /></p>
</form>

```

Questo è un esempio di un possibile modulo `searchEngine.html`. Esso visualizza una tabella con i prodotti trovati. Il risultato della ricerca viene piazzato in una vista autodefinita, chiamata *one\_foundpage\_block*.

```

<table border="1">
    <tr>
        <th>
            Nome prodotto
        </th>
        <th>
            Breve descrizione
        </th>
        <th>
            Prezzo
        </th>
    </tr>
    {BEGIN one_foundpage_block}
    <tr><td align='left'><a href='showPage.php?
template=prodotti&id={id}'>{title}</a></td>
    <td align='left'>{description}</td>
    <td align='left'>{prezzo_lit}</td>
    </tr>
    {END one_foundpage_block}
</table>

<table width='100%'>
<tr><td>
<a href="{searchengine_prev}">&lt;-Precedenti</a>
</td><td>
<a href="{searchengine_next}">Successivi -&gt;</a>
</td></tr>
</table>

```

Nell'esempio precedente, i link "precedenti" e "successivi" appaiono anche se non esiste pagina precedente o successiva di risultati della ricerca. I link generati dal plug-in in questo caso sono inattivi. Se questo è un problema dal punto di vista estetico è possibile usare codice HTML un poco più complicato, simile a questo:

```
<script>
if ({searchengine_useprev} == 1)
    document.write ("<a href='{searchengine_prev}'>&lt;-precedenti</a>");
</script>
<noscript>
    <a href="{searchengine_prev}">&lt;-precedenti</a>
</noscript>
```

Ed ecco, infine, un esempio in cui la righe trovate vengono mostrate in colori alternati, grazie all'uso di {searchengine\_row}.

Pagina mastro:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
    "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="it">
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <title>Esempio</title>
    <style type="text/css">
        td.odd { background-color: white; }
        td.even { background-color: #E2E2E2; }
    </style>
</head>
<body>
    {search_engine}
</body>
</html>
```

File results\_searchEngine.ihtml

```
<table>
    <tr>
        <th>
            Nome prodotto
        </th>
        <th>
            Breve descrizione
        </th>
        <th>
            Prezzo
        </th>
    </tr>
    {BEGIN one_foundpage_block}
    <tr>
        <td class="{searchengine_row}">
            <a href="showPage.php?template=x&id={id}">{title}</a>
        </td>
        <td class="{searchengine_row}">{description}</td>
        <td class="{searchengine_row}">{prezzo_lit}</td>
    </tr>
    {END one_foundpage_block}
</table>
</body>
```

Ecco, infine, un esempio di ricerca su più sezioni contemporaneamente. Si noti che in un caso come questo Sar-At può effettuare ricerche solo all'interno di campi presenti in tutte le sezioni cercate. In altre parole, i campi incorporati Sar-At (title, keywords, description...) sono sempre ricercabili quando multiSectionsSearch è indicato. I campi definiti da voi sono ricercabili se presenti in tutte le sezioni scelte via multiSectionsSearch. Per esempio, se definite una sezione "articoli", una sezione "cronaca" e una sezione "approfondimenti" e se tutte queste sezioni contengono i campi "firma\_articolista" e "testo", allora potete usare multiSectionsSearch.

```
<form action="showPage.php" method='post'>
  <fieldset><legend><strong>Ricerca avanzatissima</strong></legend>
  <input type='hidden' name='template' value='faq' />
  <!--Per indicare in modo fisso le sezioni su cui ricercare, si usa:
input  type='hidden'  name='multiSectionsSearch'  value='articoli,  faq,  news,
notizie'
  -->
  <input type='hidden' name='masterPage' value='risultati.html' />
  <input          type="hidden"          name="results"          value="ihtml/
results_searchEngine.ihtml" />
  <input type="hidden" name="noresults" value="ihtml/nothing_found.ihtml" />

  <!--Per lasciare che sia il visitatore a scegliere le sezioni: -->
  Seleziona le pagine che si trovano nelle sezioni
  <select name='multiSectionsSearch[]' multiple='yes'>
    <option selected>articoli</option>
    <option selected>faq</option>
    <option selected>news</option>
    <option selected>notizie</option>
  </select>
  <br />
  in cui
  <select name='searchFld[]' multiple='multiple'>
    <option selected value='keywords'>le parole chiave</option>
    <option selected value='title'>i titoli</option>
    <option value='description'>le descrizioni</option>
  </select>
  <select name='comparison'>
    <option selected value='contains'>contengono</option>
    <option value='starts'>cominciano con</option>
    <option value='ends'>finiscono con</option>
    <option value='equals'>sono uguali a</option>
  </select>
  <input type='text' name='search' value="" size='25' />

  <p>Mostra i risultati a gruppi di
  <select name='recordsReturned'>
    <option>5</option>
    <option selected>10</option>
    <option>15</option>
    <option>20</option>
    <option>25</option>
  </select>
  <br />
  ordinandoli per
  <select name='sortBy'>
    <option selected value='title'>titolo</option>
    <option value='lastMod'>data di aggiornamento</option>
    <option value='creation'>data di creazione</option>
    <option value='sarelTemplate'>sezione di appartenenza</option>
```

```
</select>
in modo
<select name='sortOrder'

<option selected value='ascending'>crescente</option>
<option value='ascending'>decescente</option>

</select></p>
<p><input type='submit' value="Trova" /></p>
</fieldset>
</form>
```

### Plug-in put\_flash

Il plug-in put\_flash, deprecato in Sar-At 10, è rimosso in Sar-At 11.

### Plug-in rectangular\_view

Il plug-in rectangular\_view, deprecato in Sar-At 10, è rimosso in Sar-At 11.

#### Plug-in hit\_counter

Sar-At permette agli amministratori del sito di sapere esattamente quanti visitatori hanno visualizzato una specifica pagina (si veda il manuale “Governare il sito” al capitolo 4, paragrafo 3, “Le statistiche”, per una descrizione della funzionalità SarelHitCounter).

Se vi interessa mostrare questa informazione anche ai visitatori potete fare uso del plug-in standard {hit\_counter}.

#### DEFINISCE

{hit\_counter}, il numero intero di visitatori della pagina corrente. Vale sempre zero se la funzione SarelHitCounter non è stata abilitata dall'amministratore nella scheda della pagina

#### PARAMETRI

{hit\_counter}, fa sempre capo alla sezione (template) e alla pagina (id) corrente. Quindi l'inserimento di {hit\_counter}, in una pagina mastro visualizzata come showPage.php?template=esempio&id=101 provocherà l'apparizione del valore SarelHitCounter visibile nella pagina id 101 della sezione esempio.

#### ESEMPIO HTML

Sei il visitatore numero {hit\_counter} di questa pagina.

#### Plug-in del carrello della spesa

Questo plug-in è utilizzabile soltanto nei siti che fanno uso del carrello della spesa Sar-At (abilitato attraverso la linguetta “Ecommerce” visibile nelle schermate Sar-At).

Immaginate per esempio un visitatore che in un negozio online acquista due penne da 4 euro ciascuna e una matita da 3 euro. L'IVA è fissata al ventidue per cento e le spese di spedizione sono impostate al valore fisso di quattro euro a prescindere dal numero di pezzi acquistati. Il visitatore ha digitato un codice coupon che gli dà diritto a uno sconto di un euro se acquista merce per almeno 10 euro.

#### DEFINISCE

{sarelcart\_expense}, il valore totale di tutti gli oggetti presenti nel carrello. Nel nostro esempio, vale 11.00



{sarelcart\_num\_items} il numero di oggetti nel carrello. Nel nostro esempio, vale 3.  
{sarelcart\_shipping\_handling}, le spese di spedizione. Nel nostro esempio, vale 4.00.  
{sarelcart\_taxesbeforepromo}, le tasse sulla vendita se non vengono considerati gli sconti. Nel nostro esempio, vale 2,42.  
{sarelcart\_beforepromo}, il valore totale del carrello prima che vengano applicati gli sconti. Nel nostro esempio, vale 17,42 (che è il valore di sarelcart\_expense + sarelcart\_shipping\_handling + sarelcart\_taxesbeforepromo)  
{sarelcart\_promo} è il valore assoluto dello sconto dopo che sono state applicate tutte le limitazioni del caso. Nel nostro esempio, vale 1.  
{sarelcart\_promo\_percent} è il valore dello sconto come percentuale sul valore della merce. Nel nostro caso, un euro su undici significa uno sconto del 9,09%.  
{sarelcart\_taxes} sono le tasse sulla vendita dopo aver considerati gli sconti. Nel nostro esempio, vale 2,20 (il 22% di 10 euro).  
{sarelcart\_grandtotal}, il totale del valore della merce, più le spese di spedizione, più le tasse, meno lo sconto. Nel nostro esempio, vale 16,20 (cioè 10 euro la merce, 4 euro la spedizione, 2,20 le imposte).

#### LIMITAZIONI

Per il corretto funzionamento del plugin è indispensabile che il tag {sarelcart\_expense} sia presente nella pagina e si incontri per primo. Per esempio, non è accettabile scrivere

Il tuo carrello contiene {sarelcart\_num\_items} oggetti per un valore totale di {sarelcart\_expense} Euro.

Bisogna invece scrivere:

Il tuo carrello ha un valore di {sarelcart\_expense} Euro e contiene {sarelcart\_num\_items} oggetti.

Il limite può venire aggirato con un uso giudizioso dei commenti HTML. Per esempio:

```
<!--Scrivo qui {sarelcart_expense} per soddisfare i requisiti -->
```

Il tuo carrello contiene {sarelcart\_num\_items} oggetti per un valore totale di {sarelcart\_expense} Euro.

#### Plug-in sareURL

Questo plug-in è utile normalmente soltanto nei siti che fanno uso del carrello della spesa Sar-At (abilitato attraverso la linguetta “Ecommerce” visibile nelle schermate Sar-At), ma funziona con qualsiasi pagina mastro usata attraverso showPage (non va invece usato dentro showForm).

Esso restituisce l’indirizzo URL della pagina corrente.

#### DEFINISCE

{sareURL}, l’indirizzo relativo della pagina corrente. Un esempio: showPage.php?template=prodotti&id=101&masterPage=homeProdotti&search=gelati

#### Plug-in template

Il plug-in {template} si limita a rendere disponibile il nome della sezione corrente. Può essere utile quando usiamo la medesima pagina mastro per due o più sezioni simili tra di loro.

#### DEFINISCE

{template} Nome della sezione corrente

#### Plug-in motore di ricerca rapido

Il plug-in di ricerca rapida, {search\_site7} è una soluzione alternativa a search\_engine ma molto più veloce e facile da impostare e mettere a regime.

#### DEFINISCE

{search\_site7} si comporta esattamente come {search\_engine}, che è stato progettato per rimpiazzare. C’è però una differenza degna di nota. Se il parametro “comparison” ha valore uguale a “fulltext” e se “multiSectionsSearch” non viene passato, search\_engine esegue una ricerca a tutto testo (di cui abbiamo

parlato nel capitolo 2) sulla sezione indicata dal parametro `template` mentre invece `search_site7` la esegue su tutte le sezioni che hanno definiti i campi per la ricerca a tutto testo.

All'interno di `results_searchEngine.ihtml` possono venire utilizzati i costrutti IF.

#### ESEMPIO HTML

Per impostare un motore di ricerca sull'intero sito seguite questa prassi. 1: scegliete tutte le sezioni in cui il motore di ricerca deve lavorare contemporaneamente. Per ciascuna di esse visitate il menu `Pagine`, e nella linguetta `Informazioni` cliccate il pulsante "Ricerca a tutto testo". Nella schermata che appare spuntate tutti e soli i campi dentro i quali il motore di ricerca deve guardare.

2. Copiate il codice seguente e incollatelo dentro a un file chiamato `results_searchEngine.ihtml`:

```
Risultati {searchengine_fromto} di {searchengine_resultsfound}
<ul>
  {BEGIN one_foundpage_block}<li><a href='showPage.php?
template={saratTemplate}&id={id}'>{title}</a><br />
  {description}{END one_foundpage_block}
</ul>
{searchengine_navigator}
```

3. Preparate la pagina che deve contenere i risultati della ricerca. Nel punto in cui questi debbono apparire scrivete semplicemente `{search_site7}`. Al punto 4 immagineremo che questa pagina si chiami `risultatiRicerca.html`, ma potete chiamarla come preferite.

4. Copiate il codice seguente e inseritelo in tutte le pagine che deve contenere il motore di ricerca:

```
<form action='showPage.php' method='get'>
  <input type='hidden' name='template' value='chisiamo' />
  <input type='hidden' name='masterPage' value='risultatiRicerca.html' />
  <input type='text' name='search' class="Stile13" />
  <input type='image' src='home/bottone_off.gif' />
</form>
```

#### Plug-in `sarat_stopgap`

Quando un visitatore autoregistrato (si veda il manuale "Governare il sito" nella sezione dedicata ai visitatori registrati) inserisce i suoi dati, egli compila un *form* HTML. Questo modulo viene mostrato a video attraverso l'uso del metodo `showForm` di Sar-At, come è spiegato in questo manuale nella sezione sui moduli. In sostanza e per esempio, il modulo di autoiscrizione chiamata "autoregistrati" apparirebbe all'indirizzo

```
showForm.php?template=autoregistrati
```

Noi, come proprietari del sito, potremo consultare le anagrafiche dei visitatori dall'interno del programma Sar-At. È importante, naturalmente, che nessun altro possa leggere questi dati. Però, se ci pensate bene, vedrete che c'è un rischio. Cosa succede se qualcuno digita l'indirizzo

```
showPage.php?template=autoregistrati&id=1
```

```
showPage.php?template=autoregistrati&id=2
```

```
showPage.php?template=autoregistrati&id=3
```

...e così via?

Per prevenire questo abuso, la soluzione più semplice e classica in Sar-At sta nel visitare dentro Sar-At la sezione `autoregistrati`, linguetta `Struttura`, voce "visibile" e indicare `Default: No`.

Con questo semplice stratagemma noi rendiamo non visibili ai visitatori le registrazioni.

Sar-At tuttavia permette la riregistrazione, ovvero l'auto-aggiornamento dei dati di registrazione da parte dei visitatori. Le pagine delle registrazioni vanno mantenute visibili per consentire ai loro proprietari di accedervi nuovamente e aggiornare i dati che vi sono contenuti. In questi casi è opportuno l'uso del plug-in `{sarat_stopgap}`

DEFINISCE

{sarat\_stopgap}: consente la visione della pagina solo da parte del legittimo proprietario e soltanto nel contesto di showForm, mai con showPage.

#### ESEMPIO HTML

Inseriremo nella pagina mastro con il modulo di iscrizione, all'inizio del <body>:

```
{sarat_stopgap}
```

#### Plug-in clouder

Nei blog si è diffusa l'abitudine di pubblicare un riquadro contenente gli argomenti più gettonati. Nel riquadro le parole più popolari appaiono solitamente più in grande, o con un colore differente. Un cloud è pensato per mostrare le keyword del sito, dimensionate per importanza e ipercollegate alle pagine. Nei siti ricchi di contenuto piace molto, perché invita ad esplorare il contenuto ed è molto utile se il contenuto è generato da una comunità

I cloud in circolazione, sostanzialmente, sono di tre tipi. I primi due sono in circolazione dal 2006 circa, mentre il terzo tipo è apparso molto di recente essendo più complesso da gestire e realizzare.

1. Si mostrano alcune keyword a caso, ogni volta diverse
2. Si mostrano tutte e sole le keyword più importanti
3. Si mostrano tutte, però siccome son molte bisogna avere un meccanismo di paginazione

Sar-At ha un plug-in incorporato, Clouder, che realizza e gestisce tutti e tre i tipi di cloud. L'aspetto grafico del cloud è personalizzabile attraverso l'uso dei CSS.

#### DEFINISCE

{clouder@cloud} inserisce un cloud del primo tipo

{clouder@tag} inserisce un cloud del secondo tipo

{clouder@alpha} inserisce un cloud del terzo tipo

#### PARAMETRI

{clouder}, fa sempre capo alla sezione (template) corrente. Le parole chiave da listare vengono prese dal campo keywords di tutti i record (pagine).

La versione 10 di Sar-At introduce alcuni parametri ulteriori che possono venire configurati attraverso una riga di codice PHP

Parametro	Scopo	Valore default	di
MAX_TAG_IN_CLOUD	Numero di tag nel cloud (casi 1 e 2)		30
SMALLEST_FONT_SIZE	Dimensioni in pixel dei link più piccoli		10
GREATEST_FONT_SIZE	Dimensioni in pixel dei link più grandi		22

#### ESEMPIO HTML

Esempio di uso dei parametri ulteriori:

```
<?php
define ('MAX_TAG_IN_CLOUD', 20);
?>
{clouder@cloud}
```

#### Plug-in network\_authentication

Come visto nella parte terza di questo manuale, Sar-At consente il log-in anche attraverso interconnessione con tutti i più diffusi social network. Il plug-in network\_authentication è stato scritto per semplificare al massimo la realizzazione di questa interconnessione.

Quando richiamato, provoca l'apparizione dei pulsanti che consentono il log-in attraverso tutti i network abilitati in Sar-At, menu Utilizzatori, per il gruppo indicato.

#### DEFINISCE

{network\_authentication@nome gruppo}, modulo di log-in via social network

#### PARAMETRI

Nome del gruppo di utenti registrati (come riportato nel menu Utilizzatori di Sar-At) al quale gli utenti vogliono fare log-in.

#### ESEMPIO HTML

<p>Per lasciare un commento al blog, fai log-in</p>

```
<form method="post" action="/sarat/cart/loginBuyer.php">
<input name="group" type="hidden" value="commentatori" />
<input name="koURL" type="hidden" value="registrati.html" />
<label for="username">Email</label>
<input type="email" name="username" id="username" />
<label for="password">Password</label>
<input id="password" type="password" name="password" />
<a href="/dimenticato.html">Hai dimenticato la password? Clicca qui per
riaverla</a><br />
<a href="/showForm.php?template=commentatori">Non sei ancora registrato?
Registrati!</a></div>
<input type="submit" value="Log-in" />
</form>
```

<p>Oppure autenticali con il tuo social network preferito</p>
{network\_authentication@commentatori}

#### Plug-in fusi orari

Il plug-in saratTimezone vi permette di gestire un sito che opera in tutto il mondo e vuole mostrare a ogni visitatore l'ora nel suo fuso orario locale.

Immaginate un blog in cui l'autore italiano scrive un post a mezzogiorno del 1 gennaio, poi un commentatore russo aggiunge una osservazione un'ora dopo alle 15:00 ora di Mosca e infine un commentatore di New York aggiunge un ulteriore commento un'ora dopo alle 7:00 del mattino ora locale.

Sar-At, automaticamente, memorizzerà nella sua base dati tutti gli orari normalizzandoli (si veda il manuale *Mantenere il sito*. Sarebbe però bene che gli orari vengano mostrati in modo comprensibile ai visitatori, per esempio l'americano vorrà vedere che il post italiano iniziale è delle 5:00 e il commento russo è delle 6:00.

#### DEFINISCE

{saratTimezone@nome campo di tipo ora}, mostra l'orario nel fuso orario locale.

#### LIMITAZIONI

È necessario un JavaScript nel browser che ottiene quale sia il fuso locale del visitatore e lo invia al server al momento del login.

#### ESEMPIO HTML

Nel form di login, aggiungere un campo hidden chiamato sarat-VisitorTimeZoneOffset. popolarlo con JavaScript. Qui vediamo un metodo che utilizza la libreria jQuery.

```
<form method='post' action='carat/cart/loginBuyer.php'>
  <input id='tz' name='sarat-VisitorTimeZoneOffset' type='hidden'
value='0' />
```

```
[...]
<script>
    $(document).ready(function(){
        var visitortime = new Date();
        $('#tz').val(visitortime.getTimezoneOffset());
    });
</script>
```

Nei form che accettano in input informazioni di tipo *Ora* o *Data* e ora, va inserito un parametro hidden di nome `sarat_tz` che contiene il nome del campo o dei campi (separati da virgola) da manipolare.

```
<form action="showForm.php" method="post">
    Digitare l'ora in cui è stato ritirato il pacco:
    <input type='hidden' name='sarat_tz' value='DataOraRitiro' />
    [...]
```

Nelle pagine mastro classiche che mostrano al visitatore un orario, usare il plug-in per convertire l'orario nel fuso del visitatore.

Il pacchetto è stato ricevuto dal corriere alle  
{saratTimezone@DataOraRitiro}

#### Plug-in: risposte a domande frequenti

##### HTML5

*I plug-in di Sar-At che generano codice sono compatibili HTML5?*

Sì. Tutto il codice generato da Sar-At (per esempio, attraverso l'uso di plug-in) e inserito nelle pagine web è compatibile con lo standard XHTML e con HTML5.

Per semplicità, scorrevolezza e rispetto della consuetudine, questo manuale fa continuamente uso del termine "linguaggio HTML".

# Parte sesta: altri plug-in





Questa sezione del manuale documenta gli altri tipi di plug-in che possono venire aggiunti alla vostra copia di Sar-At e spiega come gestirli.

## Il manager di plug-in

Il manager di plug-in viene richiamato scegliendo la voce *Plug-in* dal menu *Sito*; permette di abilitare e disabilitare i plug-in installati, sito per sito. È utilizzabile solo da *root* e dal responsabile del sito: tutti gli altri amministratori non vi hanno accesso.

Quando richiamiamo il manager dei plug-in appare una schermata divisa in quattro pannelli (con quattro linguette). Ogni pannello mostra uno tra i quattro tipi di plug-in: quelli per showPage già presentati nel precedente capitolo, quelli per showTwig, editAdd, validate descritti nel seguito di questo capitolo.

Quando richiamiamo il manager dei plug-in per la prima volta, tutti i plug-in incorporati di Sar-At sono abilitati (e non possono venire disabilitati); tutti gli altri sono disabilitati e inutilizzabili, ma potete abilitarlo con un clic. Appare un plug-in per riga; per i plug-in incorporati c'è anche un link al manuale.

Plug-in showPage ?			
Nome	Aiuto	Cache	Abilitato
template			<input checked="" type="checkbox"/>
sareURL			<input checked="" type="checkbox"/>

## Plug-in per showPage

I plug-in per le pagine mastro classiche si installano in `sarat/plugin`.

La colonna *Cache* indica se il plug-in è compatibile con la cache di Sar-At per l'accelerazione delle pagine. Per esempio, `search_site7` non lo è, perché viene usato per creare un motore di ricerca ed evidentemente la pagina con i risultati della ricerca non è identica per tutti i visitatori (non tutti i visitatori cercano sempre esattamente la stessa cosa).

I plug-in non incorporati, appena installati, appaiono con la cache disabilitata per prudenza: chiedete all'autore del plug-in se esso sia compatibile con la cache. Se del caso, cliccate la spia rossa: essa diventa verde (cliccandola di nuovo torna rosso). Per abilitare un plug-in, spuntate la casella di controllo a destra e premete il pulsante *Registra*.

Se una riga appare in viola, significa che quel plug-in era stato precedentemente usato ma è ora indisponibile (è stato disinstallato dal server). Questo non è un problema, purché le vostre pagine mastro non cerchino di farne uso... Cliccate la spia viola per far sì che Sar-At perda la memoria del vecchio plug-in e cancelli la riga incriminata.

## Plug-in per Twig

I plug-in per Twig si installano in `sarat/Twig/Extension`.

Il plug.in Debug (descritto sul sito Twig) è automaticamente abilitato quando Sar-At percepisce che dall'altra parte del browser si trova un amministratore che ha digitato username e password valide. Disabilitato altrimenti.

## Tag definiti da Sar-At in Twig

Chi è abituato a scrivere pagine mastro classiche sa di poter usare in ciascuna pagina alcuni tag come `sarel_username` per visualizzare il nome utente dei visitatori registrati. Per Twig abbiamo realizzato una soluzione analoga ma più moderna e in linea con le caratteristiche di quel linguaggio.

Introduciamo cioè l'oggetto `saratuser`, che prende il posto della coppia `sarel_userlogged` e `sarel_username` ma aggiunge ulteriori informazioni

Nome	Valore
<code>saratuser.islogged</code>	Utente registrato (vero) o anonimo (falso)
<code>saratuser.username</code>	Nome utente
<code>saratuser.provider</code>	Metodo utilizzato per fare login (per esempio, Twitter o Facebook o Sar-At)
<code>saratuser.groupname</code>	Gruppo di utenti cui appartiene
<code>saratuser.groupid</code>	Identificatore numerico del gruppo di utenti cui appartiene

### Parametri URL in Twig

Chi è abituato a scrivere pagine mastro classiche sa di poter usare in ciascuna pagina un tag omonimo dei parametri passati per URL. Per esempio se viene chiamato `showPage.php?template=x&masterPage=x.html&gelato=fragola` allora nella pagina mastro `x.html` il tag `{gelato}` verrà rimpiazzato da `showPage` con la parola "fragola". Per Twig abbiamo realizzato una soluzione analoga ma più moderna e in linea con le caratteristiche di quel linguaggio.

Abbiamo scelto di inserire i valori della URL in un vettore chiamato `_GET` (o in alternativa, per le pagine richiamate con un form `method='post'`, nel vettore `_POST`). Quindi se venisse chiamata questa URL:

`showTwig.php?template=x&id=i&masterPage=y.twig.html&gelato=fragola`  
avremmo che `{_GET.gelato}` vale "fragola".

### Plug-in per editAdd

I plug-in per `editAdd` si installano in `sarat/editadd`.

Sono i plug-in che lavorano dentro `editAddPage`, quella schermata dove un amministratore inserisce una nuova pagina web compilando un modulo, e ne modificano l'aspetto. Per abilitarne uno dovete selezionare il suo tipo base. Per esempio, un plug-in che consente di digitare soltanto numeri interi dispari ha per tipo base il numero intero. Spuntate poi la casella di controllo all'estrema destra e premete il pulsante *Registra*.

### Plug-in per validate

I plug-in per `editAdd` si installano in `sarat/validate`.

Sono i plug-in che lavorano dentro le sezioni modulo e verificano che i dati inseriti dal visitatore siano accettabili. Vengono usati anche da `editAddPage`, quella schermata dove un amministratore inserisce una nuova pagina web compilando un modulo, ma senza modificarne l'aspetto e solamente dopo che l'inserimento dei dati è terminato. Per esempio potremmo avere un plug-in che verifica quale numero intero sia stato digitato dal compilatore e lo rifiuta se non è dispari.

Per abilitare un plug-in di questo tipo selezionate dal menu il suo tipo di dati base (nel caso dell'esempio, numero intero), poi spuntate la casella sulla destra e premete il pulsante *Registra*.

Un plug-in di validazione una volta installato nel Sar-At, appare automaticamente nel menu *Validazione* in fase di creazione di un campo (nella linguetta *Struttura*). La validazione viene eseguita sia quando un amministratore aggiunge una nuova pagina usando il back-office Sar-At, sia quando un visitatore compila un nuovo record in una sezione modulo.

Se il plug-in di validazione giudica inaccettabile il dato inserito dal proprietario del sito (o da un redattore), esso risponde con il messaggio “Non accettabile come xxx”, dove al posto di xxx si trova il nome del plugin. Qui di seguito riportiamo un elenco di plug-in e per ciascuno scriviamo quali controlli esso applica.

Un plug-in di validazione funziona solo per uno specifico tipo di dati — per esempio, un plug-in che verifica che una data sia nel passato non può essere usato su un campo di tipo "numero intero".

I plug-in incorporati di Sar-At sono:

#### PDF

Applicabile ai file. Accetta solo allegati PDF.

#### Image

Applicabile ai file. Accetta solo immagini visualizzabili da tutti i browser, quindi JPEG codificate RGB (ma non CYMK), GIF e PNG.

#### no\_http

Applicabile ai testi lunghi. Vieta l’inserimento di link http. È molto utile nei forum, per impedire che i manigoldi inseriscano link ai loro siti truffaldini.

#### Futura

Applicabile a date. Verifica che la data digitata sia posteriore alla data di oggi.

#### Passata

Applicabile a date. Verifica che la data digitata sia anteriore alla data di oggi.

#### Email

Applicabile a testi brevi. Verifica che il testo costituisca un valido indirizzo di posta elettronica (tra l’altro non accetta casella@nomesito.it se su Internet nessuno ha registrato il [nomesito.it](http://nomesito.it)).

#### CAP

Applicabile a testi brevi. Verifica che il testo costituisca un valido codice di avviamento postale italiano. Accetta sia il formato interno (per esempio, 20100 per Milano) che il formato europeo (per Milano, .I-20100)

#### Codice fiscale

Applicabile a testi brevi. Accetta codici fiscali di sedici caratteri. Riconosce e valida con successo anche i rari codici assegnati dall’agenzia delle entrate in caso di omocodia.

#### Partita IVA

Applicabile a testi brevi. Accetta numeri di partita IVA nel formato italiano (undici cifre).

#### IBAN

Applicabile a testi brevi. Accetta coordinate bancarie (anche non italiane). Verifica sia la lunghezza che il codice di controllo finale.

#### IP

Applicabile a testi brevi. Accetta numeri IPv4 come 198.162.0.1.

#### URL

Applicabile a testi brevi. Accetta indirizzi URL completi http e https.



### Password

Applicabile a testi brevi. A volte desideriamo che i nostri visitatori registrati scelgano per sé una password significativa, per impedire che qualche malintenzionato la indovini per tentativi e riesca così ad accedere a sfroso al contenuto riservato del nostro sito, ma è difficile costringerli.

Sar-At introduce la validazione “password”. Possiamo abilitarla (o disabilitarla) caso per caso, nella sezione che contiene i dati dei nostri visitatori registrati, cliccando la linguetta *Struttura*, selezionando il campo *Password* e cliccando l'ultima voce in fondo, *Validazione*.

Quando la validazione della password è attiva, Sar-At non accetterà una password troppo corta o troppo semplice (come, per esempio, una sfilza di lettere uguali). Sono richiesti almeno otto caratteri di cui almeno una maiuscola, almeno una minuscola, almeno una cifra e almeno un segno di interpunzione.

### Comuni italiani

Applicabile a testi brevi. Accetta soltanto il nome, in forma standard ISTAT, di un comune italiano esistente. I comuni dell'Alto Adige accettano sia la grafia italoфона che quella tedesca.. All'interno di Sar-At e nelle pagine mastro sintetiche “a fil di ferro” realizzate da Sar-At per voi appare un sistema ad autocompletamento che suggerisce la dizione corretta dopo che l'utente ha digitato le prime lettere del nome (per esempio “regg” propone Reggio nell'Emilia e Reggio di Calabria).

### Nations

Applicabile a testi brevi. Accetta soltanto il nome, in forma angloфона, di una nazione esistente. All'interno di Sar-At e nelle pagine mastro sintetiche “a fil di ferro” realizzate da Sar-At per voi appare un sistema ad autocompletamento che suggerisce la dizione corretta dopo che l'utente ha digitato le prime lettere del nome.

### Portion

Applicabile a menu popup (select). Se viene selezionato, il sistema di autocompletamento descritto nel manuale *Mantenere il sito* funziona anche su testi contenuti al centro del testo.

Per esempio, abbiamo una sezione menu che contiene i valori “ippopotamo”, “cavallo”, “accendino Zippo”. In un campo di tipo menu popup (select) che pesca i dati da questa sezione menu accendiamo la validazione “portion”. Ora quando il redattore digita “ippo” gli vengono offerte due scelte: “ippopotamo” e “accendino Zippo”. Se la validazione “portion” viene spenta, la seconda opzione non viene offerta.

### I plug-in di inserimento

I plug-in di inserimento sono chiamati anche plug-in di editAdd, perché lavorano dentro il componente di Sar-At *editAddPage*, quella schermata dove un amministratore inserisce una nuova pagina web compilando un modulo.

Sar-At normalmente mostra le informazioni che compongono la vostra pagina in formato testuale, e permette all'amministratore di modificarle sempre in formato testuale. In alcuni casi potrebbe risultare più naturale un'altra forma di inserimento. Per esempio, se state creando il sito web di un colorificio e nella sezione prodotti volete mostrare la tinta di ogni vernice, sarebbe ideale mostrare all'amministratore una tavolozza. Sar-At potrebbe poi convertire ogni colore cliccato nei codici HTML utilizzabili dalla pagina mastro.

Un altro esempio: Sar-At mostra le date in formato internazionale standard (anno-mese-giorno) ma per alcuni scopi è più semplice per l'amministratore confrontarsi con un calendarietto a video. Il plug-in incorporato *date* ne fornisce uno.



I plug-in di inserimento hanno effetto solo all'interno di editAddPage e non possono venire utilizzati da showForm, quindi non sono utilizzabili dai visitatori del sito, ma solo dagli amministratori.

#### Plug-in file

Consente di collegare un documento presente sul disco rigido dell'amministratore al database. Il documento viene automaticamente caricato sul server e rinominato in modo da evitare conflitti (questa procedura è dettagliata nel manuale "Mantenere il sito").

Quando il documento caricato è un'immagine o comunque un formato internamente riconosciuto da Sar-At, il plug-in ne offre una anteprima.

Sar-At supporta le anteprima per i seguenti tipi di documenti: immagini, PDF, Word Excel e Powerpoint (anche nei formati XML, ovvero docx, xlsx e pptx), RTF e TIFF.

Usando tecnologia AJAX, il plug-in *file* consente di cancellare il documento sul server, senza ridisegnare la schermata corrente.

#### Plug-in data

Mostra un calendario perpetuo. Cliccando una data questa viene inserita.

#### Plug-in ora

Consente di inserire un'ora del giorno (come le 15:30 del pomeriggio) o una durata in ore:minuti:secondi.

#### Plug-in data e ora

Mostra un calendario perpetuo che include la selezione dell'ora. Cliccando una data questa viene inserita.

#### Plug-in radiobutton

Mostra una serie di alternative nella forma di pulsanti radio. Funzionalmente equivalente a menu, rispetto al quale però non offre una opzione "Altro..."

#### Plug-in multiparola

Mostra una serie di alternative e permette di selezionarne nessuna, una o più d'una. Le alternative possono venire raggruppate (per esempio, i piatti di un menu possono venire offerti ma divisi tra primi, secondi e dessert).

#### Plug-in checkbox

Mostra una serie di alternative nella forma di caselle di controllo. Funzionalmente equivalente al multiparola.

### Plug-in colore

Permette di cliccare un colore da una tavolozza interattiva. Nel database finisce il codice RGB del colore, per esempio F00 per il rosso e 0F0 per il verde.

L'uso operativo dei plug-in di inserimento incorporati è dettagliato nel manuale "Mantenere il sito".

### Sviluppo di plug-in per showPage

Un plug-in per pagine mastro classiche è una funzione PHP che riceve come parametri da Sar-At il nome della sezione corrente, lo id della pagina corrente, la lingua della medesima e opzionalmente altri parametri inviati dal chiamante.

Il plug-in esegue la sua elaborazione e assegna alla globale \$t i valori di tutti i segnaposto che esso definisce.

Per esempio, se un ipotetico plug-in chiamato volumesfera venisse chiamato per calcolare il volume di una sfera di raggio dato, così:

{volumesfera@55}

allora la sua realizzazione in linguaggio PHP sarebbe:

```
<?php
function volumesfera ($tabella, $pageId, $languageId, $raggio) {
    global $t;        // Istanza del template

    $mioNomeCompleto = "volumesfera@$raggio";
    define ('PIGRECO', 3.1415926);
    $t->set_var($mioNomeCompleto, $raggio*$raggio*$raggio*4/3*PIGRECO);
}
?>
```

### Accesso ai dati dell'utente corrente

Un plug-in, così come altro codice scritto in PHP e che fa affidamento su Sar-At, può ricevere dal sistema informazioni sull'utente che sta guardando la pagina web.

Per verificare se l'utente corrente è registrato come parte di un gruppo (oppure se ha diritto di accedere a una certa sezione).

```
$utente = SaratUserLogged($istanza, $id_gruppo=NULL, $sezione=NULL);
```

Per eseguire il login dell'utente corrente:

```
$utente = SaratLogin($istanza, $provider, $id_gruppo, $username=NULL,
$password=NULL);
```

Primo esempio: nel vostro blog volete accettare commenti ma senza costringere i lettori a compilare un modulo di iscrizione. Potete accettare le loro credenziali Facebook o Twitter

Secondo esempio: state realizzando un CRM online per i clienti di una azienda e volete consentire l'accesso a chiunque sia inserito nella base dati dell'azienda, senza doverne ricevere una copia completa e tenerla aggiornata. È sufficiente che il CRM offre una interfaccia standard OpenID per ottenere il risultato senza scrivere una sola riga di codice.

Le funzioni SaratUserLogged e SaratLogin restituiscono un record così composto

Campo	Valore
\$utente['islogged']	true o false a seconda che l'utente corrente sia registrato o meno
\$utente['username']	nome utente
\$utente['groupname']	nome gruppo utenti registrati di cui fa parte

Campo	Valore
\$utente['groupid']	id numerico del gruppo utenti registrati di cui fa parte
\$utente['provider']	fonte dell'autenticazione
\$utente['sanitizedname']	nome utente in un formato immune dalle SQL injection e che può venire usato direttamente in una query alla base dati

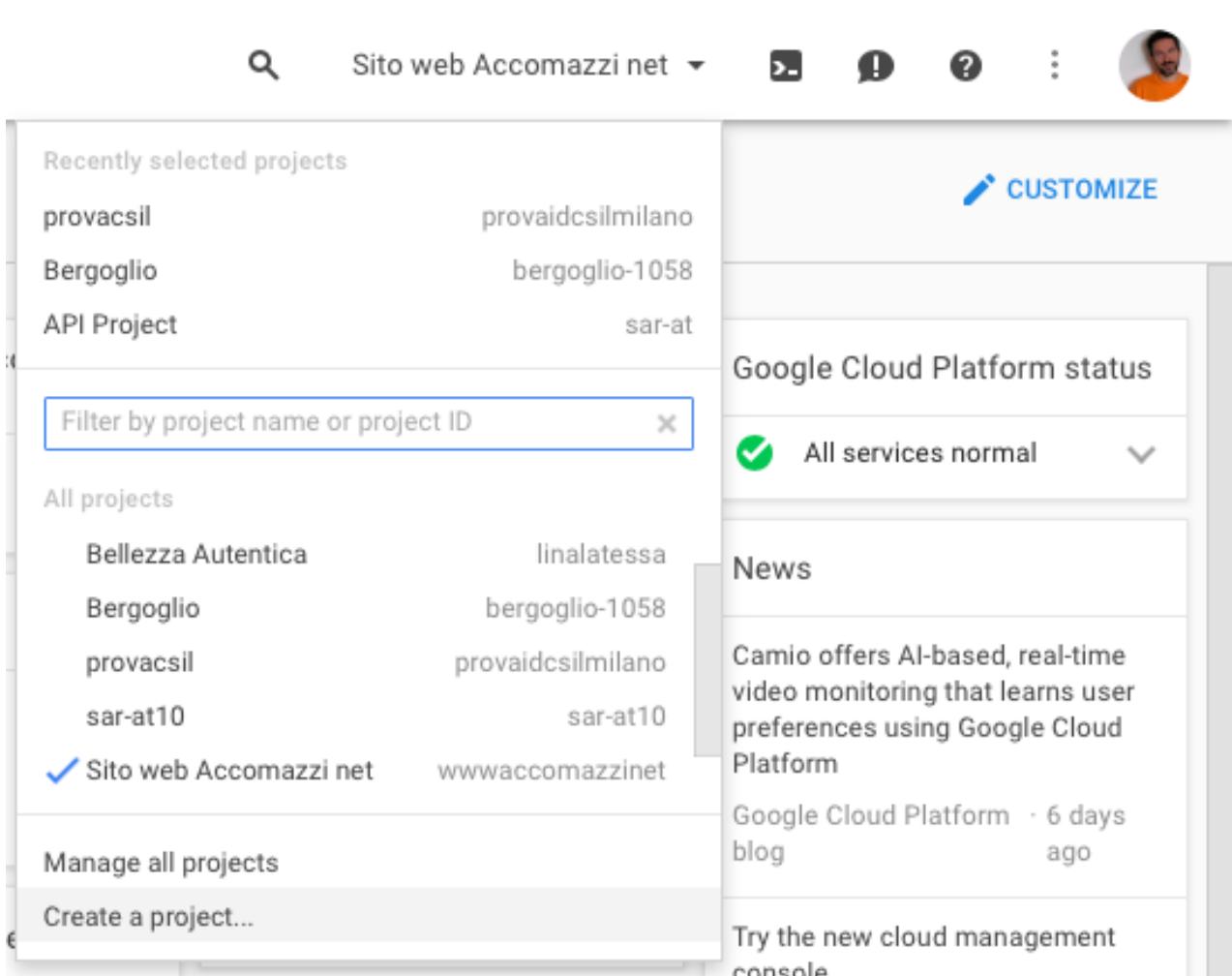
La fonte dell'autenticazione vale "Sar-At" per un login tradizionale effettuato dal nostro programma. Per log-in di nuovo tipo avremo come fonte il nome del servizio di autenticazione -- un social network o un server OpenID. Ecco i valori possibili predefiniti e preconfigurati

AOL
Facebook
Foursquare
Google
LinkedIn
Live
Twitter
Yahoo

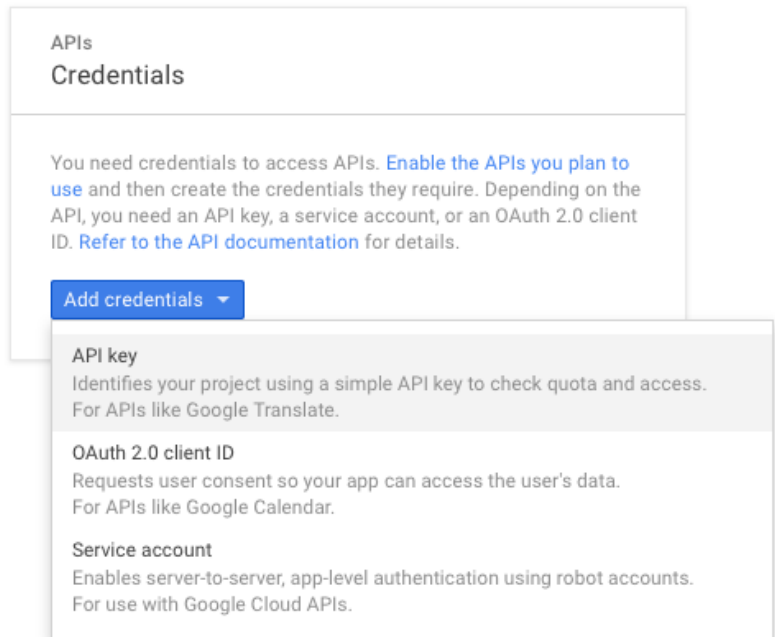
# Parte settima: funzioni avanzate

## Abilitazione dell'integrazione con Google Analytics

1. Entrate su [www.google.com/analytics](http://www.google.com/analytics). Se ancora non avevate creato un account, fatelo ora. Quando vedrete apparire il grafico dei visitatori, fate clic sulla barra degli indirizzi. In fondo a destra vedrete una lettera p minuscola, un lungo numero e una barra. Selezionate il solo numero e incollatelo nello spazio "Google Profile ID".
2. Sempre in Google Analytics, copiate l'identificativo *Universal Analytics*, quello che comincia con UA. Lo trovate un po' ovunque, in particolare sotto al menu *Admin*, colonna centrale *Property*, facendo clic sul nome del vostro sito. Copiatelo e incollatelo in Sar-At nello spazio "Google Analytics".
3. La schermata di Sar-At in basso presenta un pulsante "Abilita integrazione con Google API". Premetelo e verrete portati nel pannello Google Developers Console. Se non avevate mai lavorato, create un account. Altrimenti, nel menu centrale (a destra della vostra foto) selezionate l'ultima voce, *Create a project...*



4. Nel box blu che è apparso, *Use Google APIs*, selezionate la voce *Enable and manage APIs*. Appare un elenco di funzionalità a disposizione.
5. Nel gruppo *Advertising APIs* selezionate la voce *Analytics API* e immediatamente dopo il pulsante blu *Enable APIs*.
6. Usando i menu della fascia a sinistra visitate *Credentials*. Fate clic sul pulsante blu *Add credentials | API Key*.
7. Il sistema vi chiede quale tipo di chiave vi serve. Tra le opzioni offerte selezionate la prima, *Server key*.
8. Nella schermata che appare, in alto scrivete “Integrazione Sar-At” e in quella bassa incollate l’IP numerico del vostro server web. Premete **per due volte** il pulsante blu *Create*.



### Create server API key

#### This key should be kept secret on your server

Every API request is generated by software running on a machine that you control. Per-user limits will be enforced using the address found in each request's userIp parameter, if specified. If the userIp parameter is missing, your machine's IP address will be used instead. [Learn more](#)

Name

Integrazione Sar-At

Accept requests from these server IP addresses (Optional)

Examples: 192.168.0.1, 172.16.0.0/12, 2001:db8::1 or 2001:db8::/64

109.168.125.91



IP address

Create

Cancel

9. Appare la API Key, una specie di password che di solito comincia con le lettere AI. Copiate il valore e incollatelo in Sar-At nello spazio col medesimo nome.
10. Nella schermata in cui già vi trovate e che avete usato per creare la API key, usate nuovamente il pulsante blu *Add credentials* ma questa volta sceglierete la seconda voce del menu, *OAuth 2.0 client id*. Se non avete configurato l’integrazione con i social network (spiegata nel capitolo terzo di questo stesso manuale) vi verrà chiesto di farlo ora. Nella schermata che appare dovrete inserire nome e URL del vostro sito.
11. Google vi chiede ora a quale scopo desiderate le credenziali OAuth. Selezionate la prima risposta tra quelle possibili e cioè *Web application*.

12. Appare al di sotto un riquadro che compilerete come da istruzioni a video. Come *Name*, potete scrivere “integrazione sar-at”. Nella casella di testo a metà, *Authorized JavaScript origins*, indicate la url https di Sar-At e in basso l'indirizzo di ridirezione Oauth che trovate in Sar-At nel menu *Utilizzatori* sotto *Social Network | Google*. Premete **per due** volte il pulsante blu *Create*. Il riquadro in sovrapposizione sparisce e venite riportati alla schermata precedente, dove è apparso uno spazio intitolato *OAuth client*. Da lì copiare il valore *Client ID* che incollerete in Sar-At nell'omonimo spazio.

Ciò fatto vedrete apparire il grafico del traffico di visitatori nella schermata iniziale di Sar-At. Al primo accesso vi verrà richiesta l'autenticazione Google, per la quale userete lo stesso indirizzo email e la stessa password che avevate scelto per voi stessi al punto 1, premendo il pulsante *Authorize Analytics*.

### Google Ecommerce

Sar-At 11 vi permette di tracciare l'itinerario che il visitatore compie nel vostro sito prima di concludere con successo un acquisto, grazie all'integrazione con le funzioni e commerce di Analytics. Il cliente deve abilitarlo nel pannello Google Analytics e poi copiare lo script seguente nella pagina "grazie per l'acquisto".

Le due Viste qui impiegate, *saratGAecommerce* e *saratGAcart*, sono automaticamente create da Sar-At 11 in fase di configurazione dell'ecommerce. Su siti creati con versioni di Sar-At precedenti alla versione 11, è sufficiente rientrare nel menu Ecommerce, voce Configurazione, fare clic sull'ultima fase della configurazione (cioè Scontistica) e premere il pulsante Registra senza fare alcuna modifica per provocare la creazione delle viste. Questa procedura si può effettuare più volte se per qualche motivo cambia la configurazione generale del commercio elettronico, per esempio il codice UA-xxxxx assegnato da Google all' esercente.

```
<script>
  (function(i,s,o,g,r,a,m)
  {i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
    (i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();
  a=s.createElement(o),
  m=s.getElementsByTagName(o)[0];
  a.async=1;
  a.src=g;
  m.parentNode.insertBefore(a,m)
  })(window,document,'script','//www.google-analytics.com/
  analytics.js','ga');

  {BEGIN saratGAecommerce}
  ga('create', 'UA-257623-1', 'auto');
  ga('send', 'pageview');

  ga('require', 'ecommerce');
  ga('ecommerce:addTransaction', {
    'id': '{ID}',
    'affiliation': '{Affiliation}',
    'revenue': '{Revenue}',
    'shipping': '{Shipping}',
    'tax': '{Tax}'
  });
  {END saratGAecommerce}

  {BEGIN saratGAcart}
  ga('ecommerce:addItem', {
```

```

        'id': '{ID}',
        'name': '{name}',
        'sku': '{SKU}',
        'price': '{price}',
        'quantity': '{quantity}'
    });
{END saratGAcart}

    ga('ecommerce', 'send');
</script>

```

## showSitemap e il Sitemap protocol

Selezionando la voce “Mappa del sito” dal menu sito Sar-At esegue il programma showSitemap, un motore che esamina tutto il vostro sito e offre informazioni di sintesi. Appare un elenco di tutte le sezioni e per ciascuna tutte le pagine mastro abilitate. Un LED rosso segnala condizioni di errore (tipicamente la mancanza di pagine mastro valide), un LED giallo indica una sezione per cui la cache non è disponibile (tipicamente una sezione modulo) e un LED verde le sezioni efficacemente in uso. Per queste ultime viene anche indicato quante pagine sono segnalate automaticamente a Google usando il sitemap protocol.

showSitemap è anche il programma che viene richiamato da Google stesso per conoscere quante e quali pagine compongono il vostro sito se viene abilitata la integrazione con Google nell’omonima pagina del menu “Utilizzatori”. La descrizione di questa funzionalità si trova nel manuale *Governare il sito*.

Mapa del sito

Sezione	Pagina mastro	Nella mappa del sito
faq	default	18
istituzionale	default	6
manuale	default	246
movie	default	2

## Indirizzi pseudo-statici

Sar-At invia (negli header http, cioè in modo automatico e invisibile) ai programmi che lo richiedono la data di ultima modifica della pagina, come farebbe un server web che spedisca una pagina statica. Ciò aiuta i motori di ricerca a indicizzare il sito.

Tutte le pagine generate e servite da Sar-At hanno normalmente una forma di questo tipo:

`http://www.nomedelsito.it/showPage.php?template=prodotti&id=numero`

Sar-At permette di nascondere la natura dinamica delle sue pagine dietro a nomi assolutamente indistinguibili da quelli della pagine scritte a mano: gli indirizzi pseudo-statici.

### Traduzione meccanica

La traduzione meccanica degli indirizzi è stata rimossa in Sar-At 12.0

### URL amichevoli

Sar-At può generare per voi URL amichevoli. La introduzione di questa funzionalità è nel manuale *Mantenere il sito*, l’uso delle URL amichevoli è spiegato in questo stesso manuale nel capitolo 2 sulle Viste. La scelta di usare URL amichevoli deve avvenire quando il sito viene creato. La URL amichevole viene memorizzata nel campo incorporato saratUrl di ciascuna pagina (record).

## Inclusione di frammenti html



Questa tematica è già stata affrontata nel capitolo 1 di questo stesso manuale, ma la riprendiamo ed ampliamo qui. Riguarda esclusivamente le pagine mastro classiche.

Un grande sito che contiene molte pagine mastro è un sito che contiene molte pagine simili. Quando studiate l'immagine coordinata di un sito web finite sempre per elaborare una testata, un piè di pagina, una barra di navigazione... (in termini HTML5: header, footer e nav) tutti questi elementi finiscono, duplicati, in tutte le pagine mastro. Si verifica così, per quanto minimizzato, il problema che affligge sempre i siti sviluppati con metodo tradizionale: quando desiderate modificare l'aspetto grafico generale dovete modificare molte pagine distinte applicando le stesse modifiche a ciascuna. Usando Sar-At e un sito di centomila pagine virtuali magari dovrete modificare soltanto venti pagine mastro, ma è comunque un lavoretto noioso. Per chi usa un programma autore come Dreamweaver questo non è un problema, perché quei programmi offrono meccanismi propri di *templating* per alterare un gruppo di pagine simili in simultaneo. Gli altri troveranno utile l'istruzione INCLUDE di showPage.

Prendete una delle pagine mastro e localizzate il codice HTML della testata. Per fare un esempio banale, potremmo trovare in home page logo e titolo del sito e lo stesso HTML replicato in tutte le pagine mastro. Usando un editore di testo, tagliate la testata e registratela in un file esterno chiamato magari "header.ihtml" ("ihtml" sta per "html da includere", ma potete utilizzare qualsiasi altra estensione che desideriate, comprese html, htm, xml, css e txt). Al posto della testata tagliata inserite una istruzione Sar-At che la richiama, ottenendo questo effetto complessivo:

```
<!DOCTYPE html>
<html lang='it'>
<head>
  <meta charset='utf-8' />
  <title>Demo</title>
</head>
<body>
{INCLUDE header.ihtml}
Bla bla bla
</body>
</html>
```

La stessa istruzione INCLUDE può venire utilizzata in tutte le altre pagine mastro. Ora, quando volete modificare la testata di ogni pagina del sito, dovete editare il solo file ihtml.

Se create varianti localizzate dei file di include (per esempio: header.ihtml.it e header.ihtml.ru) showPage sceglie e inserisce automaticamente la variante adatta, sulla base della lingua della pagina corrente.

## I CSS

ShowPage e showTwig vengono normalmente usati per produrre pagine HTML, ma è possibile anche utilizzarli per produrre fogli stile. Per esempio e usando la sintassi showPage, se state facendo il sito di un colorificio e nelle schede prodotto aggiungete un colore, potete utilizzare quel codice colore anche nel CSS. Creerete un documento fogliostiledinamico.css che contiene qualcosa come

```
body {
  background-color: #{ColoreProdotto};
}
```

Il documento fogliostiledinamico.css andrà abilitato come pagina mastro in Sar-At, nella omonima linguetta.

Nello HTML, il foglio stile andrà richiamato come

```
<link rel="stylesheet" href="showPage.php?template=prodotti&id={id}
&masterPage=fogliostiledinamico.css" />
```

## XML

ShowPage e showTwig possono venire utilizzati anche per produrre documenti XML.

Ecco un esempio di pagina mastro classica in linguaggio XML. Come si vede, sono utilizzabili tutti i consueti costrutti Sar-At come le Viste e gli IF. Qui vengono mostrate le informazioni di un prodotto e c'è una vista che recupera tutta la gallery di foto di quel prodotto da una sezione correlata

```
<?xml version="1.0" encoding="UTF-8"?>
<prodotto title="{title}" sottotitolo="{description}">
<testo>
{IF testo}
<paragrafo><![CDATA[{testo}]]></paragrafo>{ENDIF testo}
</testo>
{BEGIN foto_produzione}
<foto didascalia="{description}" immagine="{immagine}" />
{END foto_produzione}
</prodotto>
```

### showFile: il motore Sar-At per la distribuzione di documenti riservati

Abbiamo come sia possibile con Sar-At creare sezioni del sito (o pagine individuali) accessibili soltanto ai visitatori registrati (ulteriori dettagli si trovano del manuale *Governare il sito*).

Abbiamo anche visto come Sar-At ci permetta di registrare nel suo database non soltanto testi e numeri, ma interi documenti (i campi di tipo “file”, visti nell’omonimo paragrafo della prima parte).

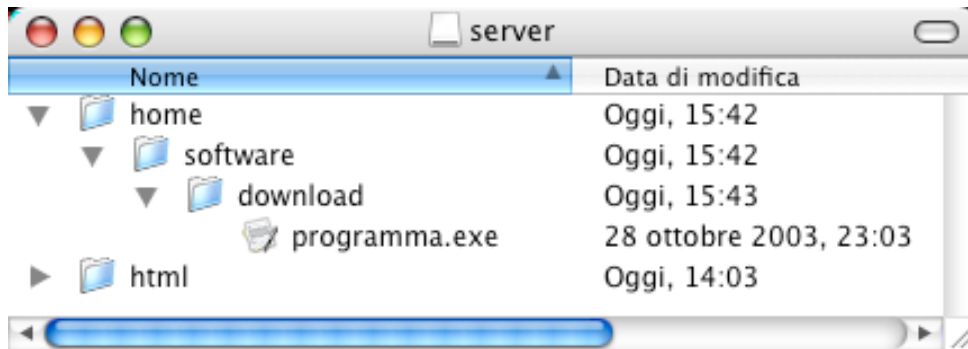
È lecito chiedersi come sia possibile combinare le due funzionalità. Cioè, come sia possibile rendere un file accessibile soltanto agli utenti registrati.

Per esempio, potremmo decidere di vendere libri in formato elettronico (PDF) solo ai clienti del sito.

Nel seguito di questo paragrafo immagineremo che nel sito [www.software.it](http://www.software.it) esiste una sezione programmi. Ogni pagina della sezione programmi contiene la descrizione di una applicazione di nostra produzione, ed è riservata a chi l’ha acquistata. Avremo dunque un campo “descrizione”, un campo “memoria richiesta”, un campo “sistema operativo”, un campo “numero di versione” eccetera. La sezione programmi conterrà anche un campo chiamato “download”, di tipo file, che useremo per conservare una copia del programma.

La soluzione più semplice in questo caso è quella sbagliata: creiamo una pagina riservata nella sezione “programmi” e carichiamo sul server il programma eseguibile dentro al campo “download” relativo. Se scegliamo questa strada, tuttavia, il documento ha un indirizzo Internet (URL) non protetto: qualche cosa del tipo [http://www.software.it/cartella\\_download/programmi\\_download\\_101.exe](http://www.software.it/cartella_download/programmi_download_101.exe). L’utente registrato “Mario” può scaricare legittimamente il documento, copiare quella URL, inviarla per posta elettronica all’utente non registrato “Pippo”. Quest’ultimo allora potrà scaricare il file, senza neppure conoscere la password di Mario. La cosa è inaccettabile.

Per mantenere i nostri programmi al sicuro li vorremo registrare in una zona del server che non è accessibile via browser: i programmi dunque non hanno un indirizzo URL e non possono venire scaricati con semplicità. Per convenzione, Sar-At vuole trovare i file ad accesso protetto dentro a una cartella chiamata home, nella radice del disco rigido. La cartella home contiene una sottocartella per ogni tipologia di documento scaricabile, chiamata come il campo in questione. La figura sottostante rappresenta schematicamente il setup:



Il file verrà scaricato dal visitatore attraverso un indirizzo come questo:

`https://www.software.it/showFile.php?id=101&template=programmi&field=download`

Quando un visitatore accede a un indirizzo di questo tipo, Sar-At ne verifica username e password, controlla che la data di scadenza sia nel futuro e garantisce che l'utente appartenga a un gruppo con diritto d'accesso sulla pagina e dunque sul documento.

### showRSS: creazione di flussi in syndication

Il protocollo RSS consente ai vostri visitatori più affezionati di ricevere le ultime notizie dal vostro sito senza collegarsi con il browser, quasi in tempo reale.

Sar-At vi permette di creare un flusso RSS con pochi clic del mouse. Nel menu Sito trovate la voce RSS. Compilate il modulo che appare, come si vede nell'esempio sottostante. Il significato di ciascuna riga di informazioni è spiegato nel seguito.

Informazione RSS	Valore
Sezione consultata	news
Invia nel flusso RSS	le notizie dell'ultima settimana
site	www.accomazzi.net
home	http://www.accomazzi.net/
description	Accomazzi.net, soluzioni per Internet
icon	[layout] ico_video.png
copyright	© 2014, Accomazzi.net Srl
managingEditor	luca@accomazzi.net (Luca Accomazzi)
webMaster	luca@accomazzi.net (Luca Accomazzi)
ttl	24:00

Nome	Obbligatorio?	Usi e note
<b>Sezione consultata</b>	Sì	Selezionate la sezione del vostro sito che viene aggiornata più di frequente — tipicamente la sezione delle ultime notizie.
<b>Invia nel flusso</b>	Sì	Potete scegliere quali e quante informazioni compariranno nel flusso RSS
<b>Site</b>	Sì	Nome con cui il flusso RSS apparirà sul computer del visitatore
<b>Home</b>	Sì	Indirizzo della vostra pagina home, o comunque della pagina che il visitatore vedrà aprirsi nel browser se sceglie di visitare il vostro sito
<b>Description</b>	No	Potete qui inserire un breve testo di presentazione del vostro sito, che apparirà nel software RSS
<b>Icon</b>	No	Immagine in formato PNG o JPEG che rappresenta il vostro sito. Tipicamente è il vostro logo e viene mostrato nel programma RSS del visitatore
<b>Copyright</b>	No	<i>Boilerplate</i> sul diritto d'autore relativamente al materiale originale che appare sul vostro sito
<b>Managing editor</b>	No	Indirizzo di posta elettronica del responsabile dei contenuti per il vostro sito
<b>Webmaster</b>	No	Indirizzo di posta elettronica del responsabile tecnico per il vostro sito
<b>TTL</b>	Sì	Time to live: è la frequenza con cui tipicamente voi aggiornate i contenuti della sezione in questione. Se in dubbio, selezionate un valore più breve del dovuto, ma non molto più breve, poiché questo farebbe crescere il traffico da e per il vostro server, rallentandolo.

### I podcast

I podcast sono un caso speciale di RSS. Per creare un podcast non dovete far altro che configurare lo RSS per una sezione Sar-At che contiene anche un campo di tipo file chiamato “mp3”. In questo campo caricherete le puntate del podcast. Il campo deve chiamarsi mp3 anche se intendete realizzare un video podcast. I seguenti formati sono supportati in Sar-At:

Formato	Estensione	Note
Audio MP3	.mp3	
Audio AAC	.m4a	
Audio AAC con immagini, link e sottotitoli	.m4b	Compatibile con iOS, segnalabile ad iTunes.
Flash video	.flv	
QuickTime	.mov	
Cellulare 3GPP	.3gp	
MPEG-4 codifica H.264 o H.265 (HEVC)	.mp4	Raccomandato.

I podcast generati da Sar-At contengono tutte le metainformazioni richieste da Apple per l'inclusione in iTunes. La URL da segnalare è  
`showPodcast.php?template=nome_sezione`

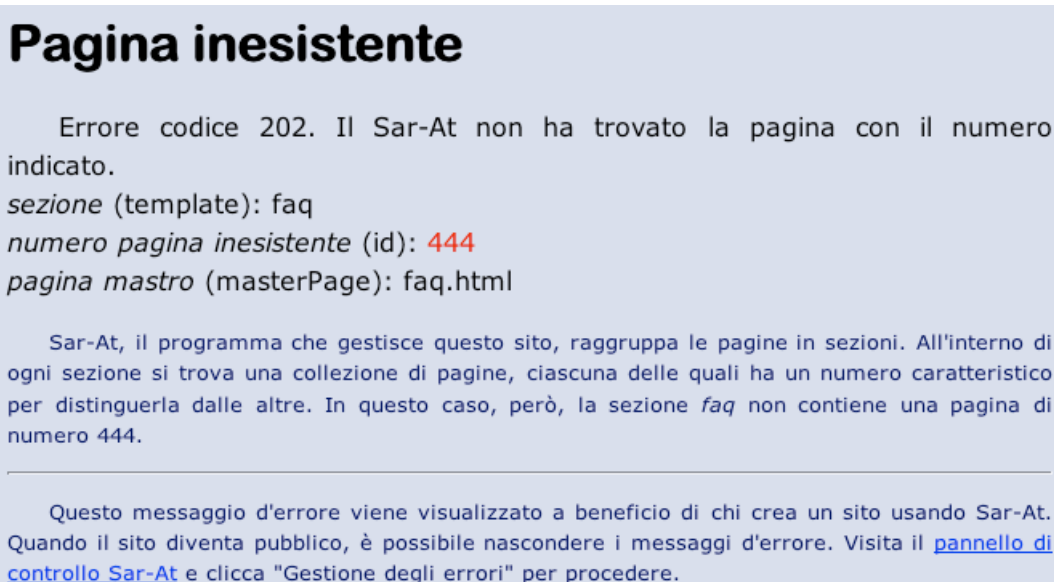
### showErr e la gestione degli errori

Come sapete, se un visitatore accede al vostro sito Sar-At chiedendo una pagina HTML statica che non esiste, il programma Apache o IIS, insomma il server web che invia le pagine attraverso l'Internet, gli manda una pagina di errore codice 404.

Se lo stesso visitatore accede al vostro sito Sar-At chiedendo una pagina HTML dinamica della sezione "faq" (che esiste) dal numero progressivo 444 (che per qualche motivo non esiste: non c'è mai stata oppure un amministratore l'ha cancellata), allora il programma Sar-At gli manda una pagina di errore. Questo comportamento viene gestito da un programma chiamato showErr ed è configurabile da voi.

#### showErr per gli amministratori

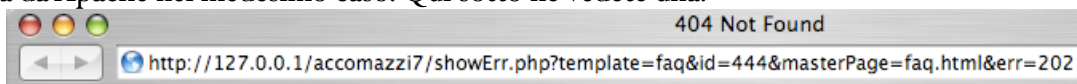
Quando vi collegate a un sito Sar-At e avete inserito la vostra password di amministratore, gli errori provocano l'apparizione di una pagina esplicitiva di errore simile a quella che abbiamo raffigurato qui di seguito:



Le informazioni contenute nella pagina dovrebbero servirvi a capire perché Sar-At non è stato in grado di generare la pagina per voi.

#### showErr per gli utenti

Quando Sar-At riceve una richiesta di generare una pagina che non può soddisfare e si rende conto che l'utente corrente non è un amministratore, genera una pagina di errore che assomiglia molto a quella prodotta da Apache nel medesimo caso. Qui sotto ne vedete una.



## Not Found

The requested URL was not found on this server.

Sar-At/700b1 Server at 127.0.0.1 Port 80

Se osservate con attenzione la URL vedrete che è esattamente la stessa del caso precedente e contiene le medesime informazioni di debugging. showErr è il nome del modulo Sar-At responsabile per la gestione degli errori. È stata richiesta una pagina nella sezione faq, numero 444 e ciò ha provocato un errore codice 202 (i codici di errore sono elencati nel paragrafo successivo). Nella URL c'è anche un

parametro chiamato “missing” che indica quale sia l’informazione non trovata, se del caso. Per esempio, se nella pagina mastro c’è scritto {cioccolato} ma la sezione faq non contiene un campo cioccolato allora scatta l’errore codice 302 e voi troverete nella URL l’indicazione missing=cioccolato.

### Codici di errore

Sar-At riconosce e caratterizza le seguenti condizioni di errore, che sono normalmente demandate al programma showErr per la gestione.

Nome	Codice	Spiegazione
<b>Sezione non specificata</b>	101	Non è stata indicata la sezione a cui si vuole accedere. Per esempio: showPage.php?id=101&masterPage=mastro.html
<b>Sezione inesistente</b>	201	La sezione richiesta non è presente nel sito. Per esempio: showPage.php?template=non_esiste_affatto
<b>Id inesistente</b>	202	È il caso dell’esempio precedente: dentro una sezione esistente si richiede di vedere una pagina che non c’è (forse esisteva in precedenza e un amministratore l’ha cancellata)
<b>Mastro non trovata</b>	203	È stata indicata una pagina mastro HTML non esistente. Per esempio showPage.php?template=x&id=1&masterPage=non_esiste_affatto.html
<b>Alias non trovato</b>	204	Accesso a showPage attraverso un alias il cui valore non è stato trovato tra quelli definiti nel sito
<b>Vista non trovata</b>	301	La pagina mastro richiede una vista che non è definita in Sar-At (probabilmente esisteva in precedenza e un amministratore l’ha cancellata). Potete usare il pulsante “verifica ora” nella linguetta Proprietà della sezione indicata per trovare esattamente di cosa si tratta.
<b>Elemento non trovato</b>	302	La pagina mastro richiama un campo che non è presente nella sezione indicata, oppure un plug-in inesistente o disabilitato. Probabilmente esisteva in precedenza e un amministratore l’ha cancellato. Potete usare il pulsante “verifica ora” nella sezione indicata per trovare esattamente di cosa si tratta
<b>Pagina invisibile</b>	401	L’indirizzo si riferisce a una pagina marcata come non visibile in Sar-At
<b>Pagina riservata</b>	402	L’indirizzo si riferisce a una pagina riservata, ovvero progettata come “accessibile solo a chi è in possesso di una password”; il visitatore ha inserito una password sbagliata o scaduta.
<b>Mastro non abilitata</b>	403	L’indirizzo cerca di combinare una pagina mastro esistente con una sezione esistente, ma Sar-At non è stato configurato per accettare questa mastro con questa sezione. Se non si tratta di un errore bisogna abilitare la pagina mastro visitando il menu Pagine, scegliendo la sezione indicata e cliccando la linguetta “Pagine mastro”.
<b>Login necessario</b>	404	La pagina è accessibile solo a visitatori registrati. Viene mostrato un modulo (form) in cui inserire username e password.
<b>Sezione non per moduli</b>	501	showForm è stato chiamato per una sezione che non è una sezione modulo.

Nome	Codice	Spiegazione
<b>Accesso in aggiornamento non autorizzato</b>	504	È stata visitata la URL showForm.php per una sezione modulo non aggiornabile , oppure un visitatore ha cercato di aggiornare una scheda che non è di sua proprietà
<b>Username già in uso</b>	505	Tentativo di registrarsi in una sezione modulo con uno username già scelto da altro visitatore.
<b>Ecommerce sconfigurato</b>	601	Tentativo di usare uno dei metodi del commercio elettronico presentati nel prossimo capitolo, ma il commercio elettronico non è configurato
<b>KO ecommerce</b>	602	Il sistema di pagamento è fallito e non è stata fornita una koURL per gestirlo in modo amichevole per il visitatore

### Personalizzazione della pagina di errore

L'apparizione della pagina di errore in una delle due forme viste in precedenza (per amministratori e per visitatori) non è obbligatoria. Potete sostituirla con una pagina HTML disegnata da voi, statica o dinamica.

Dal menu “Sito” selezionate la voce “Gestione errori”. Appare una pagina dove per ciascuno dei casi visti in precedenza potete selezionare una pagina di gestione errori. La pagina selezionata non verrà mostrata attraverso showPage ma invece attraverso il programma showErr. In sostanza ci sono due differenze: primo, potete pescare campi attraverso una o più Viste, quindi potete indicare solo una pagina mastro di tipo “home” (i tipi delle pagine mastro sono indicati nella omonima schermata). Secondo, sono definiti alcuni pseudo-campi che sono utili se volete personalizzare un messaggio di errore. Eccoli elencati:

Nome	Contenuto
template	Nome della sezione che era stata richiesta
id	Numero caratteristico della pagina che era stata richiesta
masterPage	Pagina mastro che era stata richiesta (implicitamente o esplicitamente)
search	Testo ricercato, e indicato nella URL, (spesso è vuoto)
referer	Nome del modulo Sar-At che ha dato errore (showTwig, showPage, showForm, showRSS, showSitemap) o testPage
missing	Nome della informazione cercata e non trovata (si veda sopra il paragrafo “showErr per gli utenti”).
err	Codice numerico di errore, tra quelli elencati nella tabella al precedente paragrafo.
sarelURL	Indirizzo della pagina da cui proviene l'errore
group	Definito solo nel caso 404 (si veda la tabella precedente). Nome del gruppo al quale è necessario appartenere per vedere la pagina a cui è stato tentato l'accesso.

Non è obbligatorio l'uso di alcuno dei {tag} indicati qui sopra. La pagina di gestione errore può essere perfettamente statica.

### Gli alias

Sar-At assegna automaticamente una URL amichevole a ogni pagina prodotta da un amministratore o redattore. Nel abbiamo parlato nel primo capitolo di questo libro. Qui vorremo riprendere la tematica approfondendola.

La funzionalità in questione richiede il modulo *mod\_rewrite* dentro Apache. La sua presenza (o assenza) sul vostro server è riportata nella prima schermata di Sar-At dentro la linguetta *Librerie*.

Una buona URL amichevole risulta facilmente ritrovabile in Google: per questo motivo consigliamo di separare le parole con un trattino, non con il segno di sottolineatura . Per esempio, se una pagina web parla di un dispositivo per sostenere il doppio mento degli obesi, quella pagina andrebbe chiamata *reggi-mento.html* e non *reggimento.html*. Solo in questo modo la pagina verrà segnalata a chi sta cercando in Google la parola *reggi* e la parola *mento*.

Le pagine che sono dotate di URL amichevole vengono automaticamente riportate nella mappa del sito con la URL amichevole e non con l'indirizzo *showPage* o *showTwig* sottostante.

#### Attenzione ai 404

Una URL parlante è basata sul titolo della pagina, ovvero sul campo *title*. Se una pagina italiana si chiama, per esempio, “la torta al limone con ricetta della nonna”, la sua URL amichevole assomiglierà a *torta-limone-ricetta-nonna-1234567.html*.

Se un redattore cambia innocentemente il titolo in “la torta al limone con ricetta della *mamma*”, cambia anche la URL parlante. Questo non è un gran problema, e anzi è desiderabile, purché voi non abbiate inserito nello HTML del sito un link assoluto al vecchio indirizzo.

Suggeriamo di evitare l'uso diretto nello HTML di URL amichevoli. Usate invece una Vista per recuperare il valore corrente della URL amichevole.

#### Verifica e riparazione alias

Nel menu *Sito*, voce *Alias*, appaiono non solo tutte le URL amichevoli definite sin qui ma anche una utility capace di verificare e riparare gli alias. Cancellare un alias potrebbe significare che smette di funzionare un link all'interno del vostro sito e per questo motivo il sistema vi obbliga ad eseguire un backup dell'intero database prima di cominciare.

L'utility lavora una sezione alla volta.

Sono disponibili quattro opzioni, di cui solo una, la meno invasiva, *Aggiungi alias mancanti e ripara le problematiche più semplici*, inizialmente selezionata. Le altre tre vengono abilitate solo se ce n'è potenzialmente bisogno e comunque dopo una prima esecuzione in cui abbiate abilitato la riparazione delle problematiche semplici.

#### Personalizzazione di menu all'interno di Sar-At e integrazione nuovi script

È possibile l'integrazione di nuove pagine e nuove voci di menu nel CMS, purché si sappia utilizzare il linguaggio di programmazione PHP e la manipolazione delle basi dati in SQL.

Per la costruzione della nuova pagina partite da questo scheletro:

```
<?php
require_once 'sarat/sar-el.inc.php';
$q = new DB_Sarel;
SecurityCheck($step, false);
echo SarElMakeHead ('TITOLO PAGINA', $q, 'sarat/');
?>
</head>
<body>
```



```
<?php
    $uilanguage=1; // Forza l'italiano
    echo SarElNavBar (8, $q, 'sarat/');
// vostro codice PHP
?>
</body>
</html>
```

All'interno del codice PHP potete inviare comandi SQL utilizzando la sintassi  
`$q->query("COMANDO SQL");`

Registrate lo script con un nome.php dentro alla cartella che ospita il vostro sito. Per farlo apparire nella barra dei menu con nome a piacere date questo comando al motore di base dati.  
**INSERT INTO `saratmenu` (`id`, `stringa`, `fathermenu`, `required\_privilege`, `link`, `note`)  
VALUES (10000,'Voce menu visibile a video',ID menu in cui deve apparire,'tipologia di amministratori che possono usare la voce di menu','http://www.nomesito.it/nome.php','');**

I parametri:

ID

un qualsiasi numero pari o superiore a 10.000

STRINGA

La voce di menu visibile a video, ovvero il testo che verrà cliccato dall'amministratore per invocare il vostro script

FATHERMENU

L'id numerico del menu nel quale deve apparire la vostra code. I possibili valori sono:

ID	Menu
1000	Sar-At
1500	Server
2000	Sito
3000	Pagine
4000	Utilizzatori
5000	Ecommerce

REQUIRED\_PRIVILEGE

Definisce quali amministratori possono vedere la voce di menu. I possibili valori sono

ID	Privilegio richiesto
root	Proprietario del server
own	Proprietario del sito
insert	Autorizzato ad aggiungere pagine in almeno una sezione
update	Autorizzato ad aggiornare pagine in almeno una sezione
delete	Autorizzato a cancellare pagine in almeno una sezione
publish	Autorizzato a pubblicare pagine in almeno una sezione

ID	Privilegio richiesto
peek	Almeno uno dei quattro precedenti
grant	Autorizzato a gestire sub-amministratori e utenti registrati
query	Autorizzato a definire e modificare le Viste
schema	Autorizzato ad alterare la sezione
ecommerce	Autorizzato a gestire l'e-commerce
newsletter	Autorizzato a inviare newsletter

LINK

URL dello script

NOTE

Testo libero

Ecco un esempio di come creare in intero nuovo menu intitolato “Preferenze” con due voci di menu chiamate “voglio più bene alla mamma” e “Voglio più bene al papà” visibili a tutti i redattori.

```
INSERT INTO `saratmenu` (`id`, `stringa`, `fathermenu`,
`required_privilege`, `link`, `note`) VALUES
(10000,'Preferenze',0,'Peek','','Titolo del menu');
INSERT INTO `saratmenu` (`id`, `stringa`, `fathermenu`,
`required_privilege`, `link`, `note`) VALUES (10010,'Voglio più bene
alla mamma',10000,'Peek','http://www.sito.it/mamma.php','');
INSERT INTO `saratmenu` (`id`, `stringa`, `fathermenu`,
`required_privilege`, `link`, `note`) VALUES (10020,'Voglio più bene al
papà',10000,'Peek','http://www.sito.it/papa','');
```

#### [Richiamo di funzioni interne di Sar-At a partire da script esterni](#)

Uno script PHP che richiama Sar-At può chiamare le funzioni interne del framework. Tutte le funzioni Sar-At che ha senso richiamare (le API di Sar-At) oggi sono compiutamente documentate attraverso il sistema PHPDoc.

#### [Messa in sicurezza degli script richiamati da Sar-At](#)

La grande attenzione che dedichiamo alla sicurezza dei siti significa che ci sforziamo di migliorare anche là dove eravamo già vincenti -- aggiungiamo difese al sistema anche in punti dove nessuno era mai riuscito ad attaccarlo.

Un buon esempio di questo atteggiamento è il sistema `SaratVerifyCallback` -- il quale interessa solo quelli che scrivono codice PHP di complemento a Sar-At stesso. Quando Sar-At deve chiamare uno script esterno, genera una password "usa e getta" e la passa nella URL al codice esterno, sotto al nome "otp".

Voi dovete semplicemente all'interno del vostro codice chiamare la funzione `SaratVerifyCallback` passando indietro la password usa-e-getta. In pratica, bastano tre righe di codice:

```
require_once ("sarat/sar-el_core.inc.php");
require_once("sarat/sar-at_eventmgr.inc.php");
SaratVerifyCallback (NOME DEL VOSTRO SCRIPT, $_GET["otp"]);
```

Questo sistema blocca l'esecuzione del vostro script (dentro `SaratVerifyCallback`) se esso non è stato attivato da Sar-At stesso, impedendo a uno hacker che conosca davvero bene il funzionamento interno del vostro sito di attivarne manualmente le componenti.

Il sistema di callback viene attivato per tutti gli script PHP chiamati da showForm e per tutti gli eventi gestiti in PHP. Se non aggiornate i vostri script, appare automaticamente un avviso nella prima pagina del backoffice che vi invita a farlo e menziona quali sono gli script da aggiornare.

Script insicuri	<p>blog-callback.php ⓘ          Va inserita la chiamata:  <pre>require_once("sarat/sar-at_eventmgr.inc.php"); SaratVerifyCallback ("blog-callback.php", \$_GET["otp"]);</pre></p>
-----------------	---

## Sicurezza

Nel menu Sito alla voce Sicurezza appaiono due opzioni di una certa rilevanza.

*Più persone possono accedere al sito contemporaneamente con la medesima username e password oppure Ogni credenziale autorizza un solo accesso simultaneo.* La scelta è piuttosto autoesplicativa ma vale la pena di notare che la seconda opzione offre maggiori garanzie di sicurezza rispetto alla prima e per questo motivo è selezionata per default.

Questa opzione si applica solo agli utenti registrati e mai agli amministratori, per i quali vale sempre la prima soluzione.

*Interrompi la sessione degli utenti registrati dopo...secondi,* consente di decidere quanto tempo può passare tra l'ultima consultazione di pagina da parte di un utente registrato (o un amministratore) e il momento in cui Sar-At gli richiede nuovamente la password per consentirgli l'accesso ai contenuti registrati. Questo valore non andrebbe fissato a più di 3.600 secondi (un'ora) perché ogni sessione aperta consuma memoria sul server e quindi lo rallenta.

## showCal e la generazione di appuntamenti scaricabili

Il vostro sito contiene una sezione di eventi, caratterizzati da una data (e magari ora) e inizio (e magari fine)? Sar-At vi consente di rendere più interattiva questa sezione. I visitatori del sito potranno con un sol clic aggiungere l'appuntamento alla loro agenda personale.

Il link va puntato su showCal.php, un componente Sar-At che genera un file in formato standard vcal (dunque utilizzato automaticamente da Outlook su Windows, Calendario su Mac e iOS). Eccone i parametri:

Nome parametro	Valori accettabili	Spiegazione
template	nome sezione	Il nome della sezione eventi
location	nome campo	Nome del campo nella sezione che contiene il luogo dove l'evento si svolge
from	nome campo	Nome del campo dove si trova la data dell'inizio evento
to	nome campo	Nome del campo dove si trova la data di fine evento
id	numero intero	Opzionale. Se presente, showCal esporta i dati di un singolo evento. Se assente, esporta tutti i dati di tutti gli eventi non ancora conclusi

I parametri from e to possono essere di tipo "data" o di tipo "data e ora".

## Saratizzatore

Sar-At II introduce un nuovo strumento, di sicura utilità per chi deve creare un sito Sar-At partendo da un database SQL che gli viene fornito. Lo strumento richiede un utente

particolarmente smalizzato e per questo motivo non potete accedervi attraverso la barra dei menu.

Se nella barra degli indirizzi, dopo al nome /sarat/ digitate il nome del programma, sarattizzatore.php, appare una schermata che lista tutte le tabelle della base dati.

Elenco tabelle nel database					
Nome Tabella	N° Record	Dimensione	Sar-At?		
applicazioni	20	22 kB			
articoli	716	298 kB			
articommenti	82	71 kB			
aziendale	29	58 kB			
bibliografia	12	16 kB			
bonifico_transaction	51	18 kB			
cantanti	3	48 kB			
contattaci	18	11 kB			
dicunt	15	16 kB			
errata	6	11 kB			
faq	1060	4 MB			
glossario	291	127 kB			
inlinea	115	122 kB			
intestazionequiz	4	2 kB		Sarattizza	Distruggi

Per ciascuna tabella è indicato il nome, il numero di record che essa contiene, la dimensione sul disco. Se si tratta di una tabella che è stata creata con Sar-At nessun'azione è possibile. Se la tabella invece è sconosciuta a Sar-At appaiono anche due pulsanti. *Distruggi* elimina la tabella dalla base dati. *Sarattizza* va a modificare la tabella aggiungendo i campi incorporati che Sar-At richiede per lavorarci (come id, masterId, title eccetera), poi la aggiunge al menu *Pagine* rendendola così accessibile come sorgente di pagine web.




# Parte ottava: e-commerce

Se una delle vostre sezioni contiene un catalogo prodotti, Sar-At vi permette di mettere in vendita quei prodotti. Dovete soltanto badare a un paio di dettagli informativi nel definire la sezione (in sostanza, indicare il prezzo di vendita e la quantità di scorte a magazzino) e prendere alcune decisioni a proposito della vendita — per esempio, la presenza o assenza di spese di spedizione.

**Setup commercio elettronico – fase 7 di 7** ?

✓ Prodotti > ✓ IVA > ✓ Spese di spedizione > ✓ Clienti > ✓ Notifiche > ✓ Risposte > ✓ Scontistica

Sezione dei prodotti: guida.  
Numero prodotti in vendita: ≈ 1 

Imposte applicabili: vat  
Aliquota: 22%

Calcolo spese di spedizione: no\_additional\_cost

Gruppo dei clienti registrati: potenziali  
Numero clienti registrati: 272  
Registrazione nuovi clienti: **io&tul'ho\_capito**

Carrelli in uso: 1761  
Carrelli in compilazione modulo: 219  
Carrelli in cassa: 0  
Carrelli venduti sin qui: 91  
E-mail a cui notificare vendite: [amministrazione@accomazzi.net](mailto:amministrazione@accomazzi.net)

Sar-At gestisce automaticamente il carrello della spesa, le spese di spedizione, le tasse di vendita (IVA, in Europa) l'autoregistrazione degli acquirenti, la connessione al server sicuro della banca per il pagamento.

Nel momento in cui questo manuale viene scritto, Sar-At può collegarsi ai seguenti sistemi bancari:

1. Banca Sella (Italia)
2. BNL Paribas (Italia)
3. CartaSi KeyClient (circuito interbancario italiano)
4. Constriv (circuito interbancario italiano)
5. Authorize.net (USA)
6. Braspag (Brasile)
7. HiPay (tutta Europa)
8. PayPal (mondo intero)

Altre banche possono venire aggiunte su vostra richiesta dai tecnici di Accomazzi.net, sviluppando un apposito plug-in. Il costo approssimativo di tale operazione è di 1.500 euro: consultateci per un preventivo preciso.

Il modulo ecommerce di Sar-At può venire utilizzato anche senza connessione a una banca, per esempio in soluzioni di e-commerce *business to business*, quando il pagamento avverrà a posteriori per esempio con bonifico bancario o RIBA.

Quando definite la sezione catalogo prodotti, dovete semplicemente badare a che siano presenti due campi con questo specifico nome:

*price*, di tipo numero reale. E' il prezzo al pubblico del prodotto. Potete inserire i prezzi IVA inclusa o esclusa, come preferite

*stock*, di tipo numero intero. E' il numero di copie del prodotto che abbiamo a magazzino. Se i vostri prodotti sono disponibili in un numero illimitato di copie, (per esempio, se vendete suonerie per cellulari disponibili in download) indicate un numero arbitrariamente elevato.

Anziché spiegare passo passo come creare un sito di ecommerce con Sar-At, abbiamo scelto di crearne uno noi stessi, completo di tutte le funzionalità. Voi dovrete soltanto editare le pagine mastro Sar-At aggiungendo la grafica che preferite. Come sempre in Sar-At, è anche possibile ideare il sito in modo completamente originale e con grafica completamente innovativa, progettata da zero. Semplicemente, per un sito altamente interattivo com'è necessariamente un sito ecommerce, ci è sembrato più produttivo mostrare un esempio funzionante piuttosto che pontificare sulla teoria della costruzione.

Quando attivate l'ecommerce nel vostro sito. Sar-At crea automaticamente una sezione chiamata sarelcart dove vengono inseriti i carrelli acquistati dai visitatori. Il suo riempimento e la sua configurazione sono interamente automatici, ma se volete vi è possibile aggiungere campi personalizzati al suo interno, usando la linguetta Struttura come fareste per una sezione definita da voi stessi. La voce sarelcart appare automaticamente nel menu Pagine.

Quando attivate l'ecommerce nel vostro sito. Sar-At crea automaticamente una Vista chiamata sareShowCart. Può venire utilizzata per mostrare il contenuto del carrello della spesa del visitatore, sul web e nella mail di conferma (vedere il modulo "Governare il sito" a questo proposito). Ecco un esempio:

Il tuo carrello contiene

```
<ul>
{BEGIN sareShowCart}
  <li>{numItems} bottiglie di {title}</li>
{END sareShowCart}
</ul>
```

### Configurazione del commercio elettronico

La maggior parte delle istruzioni a questo proposito si trova nel manuale *Governare il sito*. In questa sede vogliamo però far presente che Sar-At consente di attivare e usare più metodi di pagamento simultaneamente e che per questo motivo è opportuno che voi configuriate e testiate tutti i metodi di pagamento appropriati.

Quando siete pronti a inviare il cliente al pagamento (ovvero, come link "vai alla cassa" del vostro sito) voi userete il sottoprogramma checkout.php di Sar-At, indifferentemente in modalità GET (con un link diretto) o in modalità POST (con un form).

Il programma checkout.php accetta un parametro chiamato method, il cui valore è il sistema di pagamento da utilizzarsi.

Per esempio: potremmo avere:

```
<a href='/sarat/cart/checkout.php?method=paypal&koURL=%2F'>Concludi l'ordine e paga con carta di credito su Paypal<
```

```
</a>
```

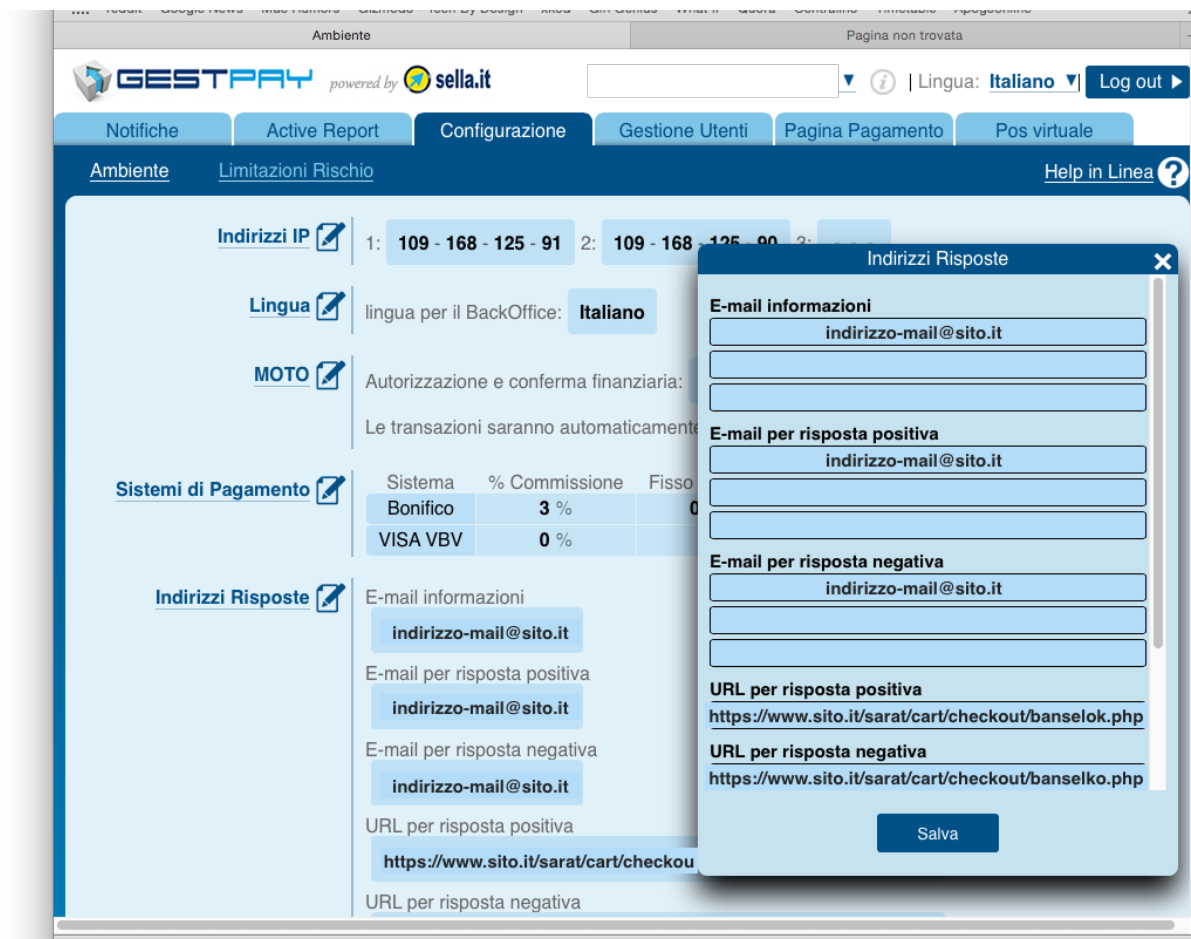
```
<a href='/sarat/cart/checkout.php?method=banca_sella&koURL=%2F'>Concludi l'ordine e paga sul sito sicuro di Banca Sella</a>
```

```
<a href='/sarat/cart/checkout.php?method=no_banking&koURL=%2F'>Concludi l'ordine e paga con bonifico bancario</a>
```

### Configurazione di Banca Sella

Sar-At supporta sia il pagamento in euro che quello in dollari statunitensi attraverso Banca Sella. Per sfruttare l'uno oppure l'altro utilizzate il method banca\_sella\_usd oppure banca\_sella (per gli euro) quando chiamate checkout.php.

Il *backoffice* di Banca Sella va configurato per inviare a Sar-At le conferme d'ordine, come si vede in figura.



### Configurazione di PayPal

L'integrazione tra PayPal e Sar-At è oggi molto sofisticata. A partire dalla versione 12, segue tutti i più moderni criteri e consente al venditore di venire certificato PayPal Pro, pagare commissioni più basse e ottenere la protezione vendite da PayPal. Nel caso in cui il cliente richieda e ottenga il rimborso da PayPal, Sar-At se ne rende istantaneamente conto e etichetta il carrello come stornato, permettendo al venditore di correre ai ripari. La descrizione della merce viene scambiata in Unicode, consentendo l'uso di alfabeti diversi dal nostro. Tutte le transazioni sono cifrate.

Il vostro cliente che acquista un carrello composto da prodotti plurimi ne vede la composizione anche dentro alle schermate di PayPal, perché Sar-At gliene invia segnalazione. Anche l'eventuale sconto guadagnato viene esplicitato e l'IVA può venire scorporata.

Lo scambio informazioni avviene anche in senso opposto. Sar-At utilizza una tecnologia PayPal chiamata IPN (Instant Payment Notification) che permette al nostro sistema di ricevere automaticamente una segnalazione quando un cliente paga una somma attesa. Sar-At reagisce alla notifica marcando il carrello come acquistato, togliendo una (o più) copie alle scorte di magazzino vendibili per ogni prodotto coinvolto nell'acquisto e inviando le mail di conferma ad acquirente e venditore.

Perché tutto ciò possa funzionare è indispensabile che voi comunichiate a PayPal l'indirizzo del componente di Sar-At che si occupa di questa integrazione. La schermata successiva mostra come fare. Naturalmente indicherete il nome del vostro sito al posto di quello accomazzi.net.



## Modifica impostazioni Notifica immediata del pagamento (IPN)

PayPal invia i messaggi IPN all'indirizzo specificato di seguito.

Per iniziare a ricevere i messaggi IPN, immetti l'URL di notifica e seleziona l'opzione **Ricevi messaggi IPN**. Per non ricevere i messaggi IPN, seleziona **Non ricevere messaggi IPN**. PayPal continuerà a generare e archiviare i messaggi IPN (o finché non disattivi la funzionalità IPN).

URL della notifica

Messaggi IPN

- Ricevi messaggi IPN (attivata)  
 Non ricevere messaggi IPN (disattivata)

Salva

Annulla

Quando configurate Sar-At per interagire con PayPal, usando il menu *Ecommerce* | *Configurazione* di Sar-At, viene automaticamente creata per voi una sezione chiamata `paypal_transaction`, la quale riceve i dati da PayPal ogni volta che una transazione viene confermata. Il suggerimento è che questa sezione venga usata come tabella anagrafica per i clienti del sito: in questo modo chi lascia nome, cognome eccetera in fase di registrazione trova questi dati precompilati nelle schermate all'interno di PayPal. Eventuali correzioni effettuate direttamente dentro PayPal vengono riportate all'indietro in Sar-At.

La tabella `paypal_transaction` creata automaticamente in Sar-At è anche predisposta per l'integrazione con i social network. In altre parole, se lo vorrete consentire, chi arriva da Facebook si troverà già predisposto con tutti i dati anagrafici incluso indirizzo, numero di telefono ed email e questi dati verranno automaticamente mantenuti nei passaggi verso PayPal e in ritorno da PayPal.

### Configurazione di HiPay

Sar-At supporta sia HiPay Direct (sistema basato su protocollo SOAP) sia HiPay Fullservice (sistema basato su connessione RESTful). Nelle schermate di Sar-At li trovate rispettivamente come "HiPay 1" e "HiPay 2". Commercialmente, in Italia viene spinto quasi esclusivamente il secondo metodo.

Attraverso HiPay, Sar-At può accettare i pagamenti SisalPay. I carrelli pagati con SisalPay appaiono nelle vostre schermate come "in attesa di pagamento". Quando il cliente passa dalla ricevitoria ed esegue il versamento, HiPay segnala immediatamente a Sar-At la transazione e Sar-At sposta il carrello in condizione "pagato", senza necessità di vostro intervento.

Se HiPay incontra un acquisto a rischio truffa, Sar-At invia una mail al venditore per consentirgli di valutare prontamente la situazione (HiPay non lo fa di suo). Se a questo punto il venditore approva il rischio nel pannello di controllo HiPay, Sar-At automaticamente segna il carrello come "pagato" anche se HiPay non gli manda immediatamente la conferma dell'accredito, per consentire la spedizione della merce o lo scaricamento del download.

### Metodi del commercio elettronico

In sostanza, il commercio elettronico in Sar-At funziona posizionando sugli elementi della interfaccia utente dei link ad *azioni*, ovvero *metodi*, che così invocati effettuano cambiamenti nel contenuto del carrello, poi portano l'utente su una pagina scelta da voi (la `returnURL`, o indirizzo di ritorno).

I metodi vanno chiamati passando loro dei parametri. Per esempio, il metodo `emptyCart`, che si usa per svuotare il carrello e ripartire col carrello vuoto, accetta solo il parametro `returnURL` e cioè l'indirizzo a cui portare il cliente a cose fatte, mentre il metodo `addToCart` che si usa per inserire un prodotto nel carrello ovviamente ha anche bisogno che voi gli comuniciate quale oggetto aggiungere tra tutti quelli presenti nel catalogo prodotti.

Alcuni parametri sono facoltativi. Per esempio, addToCart accetta un parametro facoltativo “numero di esemplari da mettere nel carrello”. Se non viene passato, Sar-At presuppone che il cliente desideri una singola copia – il caso più comune.

Il parametro returnUrl è supportato da tutti i metodi ed è sempre facoltativo. Se non viene passato, Sar-At riporta il visitatore alla medesima pagina che stava vedendo quando ha cliccato il metodo.

Tutti i metodi possono venire chiamati in modo GET o in modo POST. Per esempio, i seguenti due pezzi di HTML sono perfettamente equivalenti. Entrambi vuotano il contenuto del carrello della spesa dell'utente correntemente loggiato nel sistema, poi mostrano la home page.

```
<form action='sarat/cart/emptyCart.php' method='post'>
  <!-- torna alla home dopo aver vuotato il carrello -->
  <input type='hidden' name='returnURL' value='/index.php' />
  <input type='image' src='immagini/vuota-carrello.gif' />
</form>
```

oppure

```
<a href='sarat/cart/emptyCart.php?returnURL=%2Findex.php'><img src='immagini/
vuota-carrello.gif' /></a>
```

Il seguente elenco di metodi elenca tutti quelli disponibili. In ogni tabella la seconda colonna mostra O se il parametro è obbligatorio, F se facoltativo. Gli esempi HTML sono per pagine mastro classiche.

### addToCart

Aggiunge un oggetto al carrello.

Parametro	O/F	Contenuto
itemId	O	id dell'oggetto da aggiungere
quantity	F	numero di copie da caricare nel carrello. Se mancante, si suppone 1
title	F	nome con cui l'oggetto verrà etichettato all'interno del carrello. Se mancante, si suppone uguale al valore del campo title della tabella catalogo prodotti.
maxQty	F	Massimo numero di copie che è accettabile inserire nel carrello. Per esempio, se vendete suonerie per cellulare non ha senso acquistare la medesima tre volte, ma se vendete saponette ha molto senso comprarne tre. Se mancante, non c'è limite
koURL	F	Indirizzo cui portare il visitatore se l'articolo non può venire aggiunto al carrello. Questa strada viene seguita solo in casi straordinari, in particolare se viene aggiunta una chiave di unicità nel carrello che il nuovo articolo una volta inserito non rispetterebbe. Per esempio, un commercio elettronico che venda software scaricabile potrebbe inserire nel catalogo prodotti un ID caratteristico del prodotto e tutti le varianti di un prodotto condividerebbero la medesima ID. Se quella ID finisse nel carrello e insieme con l'indicazione dell'acquirente costituissero una chiave unica, si impedirebbe al cliente distratto di acquistare due volte il medesimo software.
Altro nome	F	Se altri campi vengono passati, Sar-At prova a registrarne il valore nella sezione sarecart. Questo meccanismo viene usato per gestire varianti di prodotto come taglia e colore. Per esempio, se io aggiungo i campi “taglia” e “colore” nella sezione sarecart, e se chiamo addToCart.php?itemId=1&qty=1&taglia=XL&colore=giallo, allora nella base dati viene registrato l'acquisto di un oggetto id 1 di colore giallo e taglia XL.

#### ESEMPIO HTML:

```
<form action="sarat/cart/addToCart.php" method="post">
  <h1>{title}</h1>
  <input type="hidden" name="title" value="{title}" />
  <p>Prezzo: {price} &euro;</p>
  <p>Quantità desiderata:
    <input type='text' name='quantity' value='1' size='3' maxlength='2'>
```

```

</p>
<input type="hidden" name="itemId" value="{id}" />
<input type="hidden" name="returnURL" value="ecom/prodotti-000001.html" />
<input type="submit" value="Aggiungi" />
</form>

```

### addToCartMultiple

Aggiunge contemporaneamente più oggetti al carrello. Pensato per chi desidera una interfaccia utente che consente di spuntare da una lista.

Parametro	O/F	Contenuto
cart{id}[set]	O	id dell'oggetto da aggiungere
cart{id}[quantity]	F	numero di copie da caricare nel carrello. Se mancante, si suppone 1
cart{id}[title]	F	nome con cui l'oggetto verrà etichettato all'interno del carrello. Se mancante, si suppone uguale al valore del campo title della tabella catalogo prodotti.
cart{id}[altro nome]	F	Se altri campi vengono passati, Sar-At prova a registrarne il valore nella sezione sarecart. Questo meccanismo viene usato per gestire varianti di prodotto come taglia e colore.

ESEMPIO HTML:

```

<form action="sarat/cart/addToCartMultiple.php" method="post">
  <ul>
    {BEGIN mostraTuttiPoster}
      <li><input type="checkbox" name="cart{id}[set]" /> {title}</li>
      <input type="hidden" name="cart{id}[title]" value="{title}" />
      <select name="cart{id}[versione]">
        <option>Italiana</option>
        <option>Inglese</option>
      </select>
    {END mostraTuttiPoster}
  </ul>
  <input type="hidden" name="returnURL" value="ecom/prodotti-000001.html" />
  <input type="submit" value="Aggiungi" />
</form>

```

### addToCartMultipleByQuantity

Aggiunge contemporaneamente più oggetti al carrello. Pensato per chi desidera una interfaccia utente che consente di specificare per tutte le voci di una lista quante copie di desiderano.

Parametro	O/F	Contenuto
cart{id}[quantity]	O	numero di copie da caricare nel carrello.
cart{id}[title]	F	nome con cui l'oggetto verrà etichettato all'interno del carrello. Se mancante, si suppone uguale al valore del campo title della tabella catalogo prodotti.

ESEMPIO HTML:

```

<form action="sarat/cart/addToCartMultiple.php" method="post">
  {BEGIN mostraTuttiPoster}
  <p><input type="text" name="cart{id}[quantity]" size='2' /> {title}
    <input type="hidden" name="cart{id}[title]" value="{title}" />
  </p>
  {END mostraTuttiPoster}

```

```



```

### changeCartQty

Consente di variare il numero di copie nel carrello, per un oggetto che già si trova nel carrello

Parametro	O/F	Contenuto
itemId	O	id nel carrello (non nella tabella catalogo prodotti!) dell'oggetto da aggiungere
quantity	F	numero di copie desiderate. Se mancante, si suppone 1

#### ESEMPIO HTML

```

<ul class='listaProdotti'>{BEGIN sareIShowCart}
<li><form action='sarat/cart/removeFromCart.php' method='post'>
  <input value='Togli dal carrello' type='image' class="deleteIcon"
src="style/img/delete.jpg"/>
  <input type='hidden' name='id' value='{id}' />
</form>
<a href='showPage.php?template=vini&id={idItem}'>
  <img alt='{title}' src='{miniatura}' />
</a>
<form action='/sarat/cart/changeCartQty.php' method='post'>
  <span>{title}<br />
  <input type='hidden' name='id' value='{id}' />
  <input type='text' name='quantity' value='{numItems}' maxlength='2' />
  bott. {numItems}/pz @ <b>{price}€/cad.</b></span>
  <input value='Aggiorna' type='image' src="style/img/refresh.jpg" />
</form>
</li>{END sareIShowCart}
</ul>

```

### emptyCart

Vuota il carrello.

#### ESEMPIO HTML:

```

<p>{IF sareIShowCart}
  Il tuo carrello della spesa contiene:<br />
  <ul>
    {BEGIN sareIShowCart}
    <li>{numItems} esemplari di {title} <a href='sarat/cart/
removeFromCart.php?id={id}'>[rimuovi]</a></li>
    {END sareIShowCart}
  </ul><br />
  Stai spendendo in tutto {sareIcart_expense} &euro; IVA esclusa per
acquistare {sareIcart_num_items} bottiglie.
  {IF sareIcart_promo}Hai digitato un codice promozionale che ti farà
risparmiare {sareIcart_promo} &euro; sull'importo prima dell'IVA.
  {ELSE sareIcart_promo}Non hai inserito alcun codice promozionale, dunque
non hai diritto a sconti.
  {ENDIF sareIcart_promo}
  L'IVA ammonta a {sareIcart_taxes} &euro;.
  Le spese di spedizione ammontano a {sareIcart_shipping_handling} euro.
  In totale dunque spenderai {sareIcart_grandtotal} &euro; (sarebbero
{sareIcart_beforepromo} se non ci fosse alcuna promozione).<br />
  <br />
  <a href='sarat/cart/emptyCart.php'>Clicca qui</a> per vuotare il
carrello.<br />
  <a href='showPage.php?template=vini&masterPage=ecom-checkout.html'>

```

```

Clicca qui</a> per passare alla cassa.<br />
<a href='showPage.php?template=vini&masterPage=ecom-home.html'>
Clicca qui</a> per la pagina home.<br />
{ELSE sareIshowCart}
Il tuo carrello della spesa &egrave; vuoto.
<a href='showPage.php?template=vini&masterPage=ecom-home.html'>
Clicca qui</a> per la pagina home.<br />
{ENDIF sareIshowCart}</p>

```

### forgetBuyer

Log-out dell'utente. Questo metodo è utile anche a chi non fa commercio elettronico, per gestire gli utenti registrati.

ESEMPIO HTML:

```

{IF sareIuserlogged}
    <p>Collegato al sistema come <em>{sareIusername}</em>. (<a href='/sarat/
cart/forgetBuyer.php'>Se non sei {sareIusername}, clicca qui</a>).</p>
{ENDIF sareIuserlogged}

```

### loginBuyer

Log-in dell'utente. Questo metodo è utile anche a chi non fa commercio elettronico, per gestire gli utenti registrati. Se si desidera utilizzarlo per consentire anche il login da social network, è consigliabile supplire con l'uso del plugin standard network\_authentication, descritto nella parte quinta di questo stesso manuale.

Parametro	O/F	Contenuto
username	O	Nome utente
password	O	Password
koURL	F	Pagine web a cui mandare l'utente se la password o la username sono sbagliate. Se mancante, lo riporta alla pagina corrente
group	F	Gruppo di utenti registrati nel quale fare log-in. Se mancante, è il gruppo di utenti registrati che rappresenta gli acquirenti dell'ecommerce, come specificato in sar-At, menu Ecommerce, voce Configurazione
byField	F	Campo il cui valore viene usato per identificare l'utente.
provider	F	Nome del network da utilizzare per l'autenticazione. Se si indica "Sar-At" o non si indica nulla, il sistema usa la propria base dati interna come sorgente di autenticazione.

LoginBuyer consente l'integrazione con i social network e l'uso di campi arbitrari per il login. Se il sito ha una tabella anagrafica clienti con tre campi unici (per esempio email, id, codice fiscale) Sar-At potenzialmente lascia fare login con uno qualsiasi di questi. E quindi il cliente può cambiare il contenuto di email e codice fiscale come meglio crede, basta che in ogni momento ciascuno di essi contenga un valore unico.

ESEMPIO HTML

```

<form action='sarat/cart/loginBuyer.php' method='post'>
    <fieldset>
        <legend align='top'>Gi&agrave; registrato?</legend>
        <input type='hidden' name='byField' value='email' />
        Indirizzo e-mail <input type='text' name='username' /><br />
        Password <input type='password' name='password' /><br />
        <input type='hidden' name='koURL' value='ecom-loginko.html' />
        <input type='submit' value='Login' />
    </fieldset>
</form>

```

Il campo unico da utilizzare per il log-in anche venire esposto all'utente, che quindi sceglie esplicitamente come farsi identificare. Nell'esempio poco sopra, lo input hidden di nome byField può venire rimpiazzato in questo modo:

```
<select name='byField'>
<option value='codice_fiscale'>Ho inserito il codice fiscale</option>
<option value='payer_email'>Ho inserito l'indirizzo mail</option>
</select>
```

Nelle vecchie versioni di Sar-At tipicamente si usava il campo email come unico possibile campo di login. Capita di trovare consumatori che dicono "voglio cambiare la mail, però la uso come username". E la risposta di solito è: se ti lascio cambiare la mail perdi tutta la storia degli acquisti perché carrelli, ricevute di pagamento, scontistiche eccetera sono legate alla username del cliente. Quindi, tradizionalmente noi creavamo siti dove i clienti si loggano usando la mail come username, e di conseguenza non gliela lasciavamo cambiare. Il problema si risolve usando un campo differente come campo chiave: l'email diventa un campo qualsiasi, magari unico, ma comunque modificabile dal visitatore.

C'è un prezzo da pagare. Quando come vero campo chiave, quello che internamente io userò come sare\_username, è l'id, se nel sito viene scritto Ciao {sare\_username} allora apparirà Ciao 12345.

L'autore del sito deve risolvere la questione con una vista, una cosa tipo

```
{IF sare_userlogged}
  {BEGIN trovanome}Ciao {nome_utente}!{END trovanome}
{ELSE sare_userlogged}
  Non hai fatto login
{ENDIF sare_userlogged}
```

### removeFromCart

Toglie un oggetto dal carrello dell'utente loggato.

ESEMPIO HTML

Vedi l'esempio al precedente paragrafo, changeCartQty

### Diagnostica

Ogni evento di commercio elettronico (oggetto aggiunto nel carrello, oggetto rimosso, log.in di un cliente, eccetera) viene registrato da Sar-At in una tabella che comprende anche data, ora, numero IP e username del cliente coinvolto; la tabella è consultabile da voi scegliendo la voce Status sistema nel menu Ecommerce.

# Parte nona: interfaccia RESTful

Sar-At da sempre è utilizzabile come *headless CMS*, ovvero separa completamente le operazioni di motore database (creazione, alterazione, popolazione delle sezioni); le operazioni di produzione delle pagine web; e la grafica HTML usata per la presentazione dei contenuti.

Per facilitare il lavoro agli sviluppatori clientside (JavaScript) e agli integratori (sviluppatori di app che caricano contenuti dalla base dati del sito e poi li manipolano sull'apparecchio mobile) le moderne versioni di Sar-At introducono una *interfaccia RESTful*.

Nel resto di questo capitolo spiegheremo come il server (Sar-At) può venire consultato dal client (una app o un JavaScript).

## Accesso alle risorse del database

Sar-At consente tre tipi di accesso REST alla base dati

1. Lettura di un record. Se il record è pubblico, per esempio una news pubblicata sul sito o la scheda di un prodotto in vendita nel commercio elettronico, Sar-At consente l'accesso sia in http che in https. Se il record è privato, per esempio l'anagrafica di un visitatore registrato sul sito, Sar-At pretende che l'accesso si svolga in https e che la richiesta sia accompagnata dalle credenziali del visitatore (username e password). La lettura viene richiesta dal cliente al server con una chiamata GET del protocollo https.
2. Scrittura di un record. Sar-At pretende che l'accesso si svolga in https e che la richiesta sia accompagnata dalle credenziali del visitatore (username e password). Naturalmente le credenziali debbono essere sufficienti allo scopo. Per esempio, in un commercio elettronico un cliente può aggiornare la sua anagrafica ma non può modificare una riga di un ordine già transato. La scrittura viene richiesta dal cliente al server con una chiamata POST del protocollo http o https.
3. Ricerca in una sezione del sito. Se i record della sezione sono pubblici, per esempio le news, Sar-At consente l'accesso sia in http che in https. Altrimenti è richiesto lo https. La scrittura viene richiesta dal cliente al server con una chiamata POST del protocollo http o https.
4. Cancellazione di un record. Simile filosoficamente al caso 2, la cancellazione viene richiesta dal client al server con una chiamata DELETE del protocollo https. Questa modalità è utile per gestire necessità come, per esempio, la cancellazione di uno degli indirizzi di spedizione associati al cliente di un commercio elettronico.

In ciascun caso, Sar-At restituisce l'OK o un codice di errore usando la sintassi classica del protocollo HTTP. Per esempio, se una app chiede al server accesso ai dati della news numero 100 ma le news sul sito sono numerate da 1 a 50, allora Sar-At restituisce l'errore 404 *not found*.

Se il risultato è positivo, il server restituisce i dati richiesti dal client in formato *JSON*. Nel caso della richiesta in scrittura, cioè il caso 2 dell'elenco qui sopra, Sar-At restituisce il JSON del record così come è salvato nella base dati dopo la modifica.

### Letture di un record

Il client richiede la lettura di un record chiamando

`http(s)://www.nome-del-sito.suffixo / API / nome della tabella (sezione) / id`

Per esempio, per leggere la news numero 73 sul sito `test.mygelsia.it`

`https://test.mygelsia.it/API/news/73`

A prescindere da quale sezione (tabella) del database stiate leggendo, e in quale sito, i primi campi della risposta JSON sono sempre i medesimi. Si tratta delle informazioni di servizio (metainformazioni) con cui Sar-At correda ogni record del database. Sono descritti nei precedenti capitoli di questo manuale ma, in sintesi, si tratta di:

Campo	L/S	Contenuto
id	L	Identificativo numerico univoco del record
masterId	L	Nei siti multilingue, identificativo numerico univoco della risorsa. Per esempio, se i record 1 e 2 hanno identico masterId 3 allora si tratta delle due traduzioni in due diverse lingue della medesima informazione
language	S	Codice della lingua in cui sono scritti i testi (1001: italiano; 1002: inglese; eccetera. Vedere il manuale <a href="#">Mantenere il sito</a> )
creation	L	Data ed ora della prima creazione del record
authorId	L	Username del creatore del record
lastMod	S*	Data ed ora della ultima modifica del record
modifierId	S*	Username del modificatore del record
visible... sarelHitCounter	S	Vedi precedenti capitoli di questo manuale.
saratUrl	S*	URL univoca sul sito per la consultazione dell'informazione

L: informazione accessibile sempre e solo in lettura

S: informazione modificabile attraverso l'operazione di scrittura sul record descritta qui di seguito.

S\*: informazione modificabile attraverso l'operazione di scrittura sul record, ma i contenuti del campo in questo caso verrebbero autonomamente popolati da Sar-At. Per esempio, modifierId diventerebbe la username dell'utente le cui credenziali sono state usate.

Seguono tutti i campi specifici della sezione (tabella) interessata.

Nelle sezioni anagrafiche Sar-At mantiene le password in formato cifrato (vedere capitolo terzo, alla voce "sezioni a massima sicurezza"). Di conseguenza, il campo "password" apparirà sempre vuoto nel JSON che vi viene restituito.

Tenete presente che, internamente, Sar-At memorizza tutti i testi in formato HTML. Quindi nei campi di testo breve e di testo lungo che vengono restituiti dal server al client dovete aspettarvi di trovare entità SGML (come &grave;) e tag HTML.

Nel caso di record riservato, la chiamata deve includere le credenziali. Per esempio, in un commercio elettronico *business to business* il prezzo riservato all'acquirente e il numero di scorta magazzino è probabilmente una informazione riservata ai membri registrati del gruppo "clienti". Il client deve includere, come parametri della chiamata GET:

- username
- password
- group, cioè nome gruppo di appartenenza

Per esempio, per leggere la anagrafica del cliente registrato numero 15159 sul sito test.mygelsia.it, avremo:

<https://test.mygelsia.it/API/registrazione/15159?username=luca@accomazzi.net&password=>



[tvumdb&group=registrati](https://test.mygelsia.it/API/registrazione/ANY?username=luca@accomazzi.net&password=tvumdb&group=registrati)

Naturalmente, il caso di una anagrafica è un caso un po' particolare. Il software client che state sviluppando probabilmente non conosce lo id univoco del cliente di cui possiede le credenziali (di solito questa informazione non traspare). Per semplificare le procedure e saltare l'inutile passaggio di una ricerca per username nella base dati, Sar-At consente in questo caso di usare "ANY" al posto dell'id numerico (in modo analogo a quanto viene fatto con la procedura showForm, vedi capitolo terzo di questo manuale). In altre parole, la URL vista qui sopra è equivalente a: <https://test.mygelsia.it/API/registrazione/ANY?username=luca@accomazzi.net&password=tvumdb&group=registrati>

**Scrittura di un record**

La scrittura di un record avviene in modo perfettamente speculare a quanto appena visto, ma il client chiama il server usando un POST anziché un GET, alla medesima URL. Per esempio, per aggiornare la scheda del cliente registrato sul sito test.mygelsia.it la URL sarà sempre <https://test.mygelsia.it/API/registrazione/15159?username=luca@accomazzi.net&password=tvumdb&group=registrati>

Se la richiesta contiene il campo id, Sar-At suppone che il client voglia aggiornare il record indicato. Se la richiesta contiene il numero zero come id, Sar-At presume che voi desideriate aggiungere un record alla base dati.

Vediamo un esempio pratico, come sia possibile aggiornare l'anagrafica del cliente già visto in precedenza sul sito test.mygelsia.it. Per massima chiarezza mostriamo l'esempio nella forma di un form HTML, perché questo rende semplice l'invio in formato POST dei dati da scrivere:

```
<!DOCTYPE html>
<html lang='it'>
<head>
  <meta charset='utf-8' />
  <title>Esempio chiamata POST di tipo RESTful</title>
</head>
<body>
<!-- post, quindi richiesta di modifica-->
<form action='https://test.mygelsia.it/API/registrazione/15159' method='post'>
  <!-- credenziali -->
  <input name='username' type='hidden' value='luca@accomazzi.net' />
  <input name='password' type='hidden' value='tvumdb' />
  <input name='group' type='hidden' value='registrati' />
  <!-- campi da sovrascrivere -->
  <input name='bolletta' type='hidden' value='email' />
  <input name='codice_fiscale' type='hidden' value='CCMLST64H08F952C' />
  <input name='newsletter' type='hidden' value='on' />
  <input name='confermato' type='hidden' value='1' />
  <input name='in_attesa' type='hidden' value='0' />
  <input name='attivato' type='hidden' value='1' />
  <input name='telefono' type='hidden' value='3497680116' />
  <input name='indirizzo' type='hidden' value='Via Scarlatti' />
  <input name='civico' type='hidden' value='3' />
  <input name='cap' type='hidden' value='20015' />
  <input name='comune' type='hidden' value='Parabiago' />
  <input name='provincia' type='hidden' value='Mi' />
  <!-- Non posso cambiare email perché è la username -->
  <input name='codice_utenza' type='hidden' value='1907479' />
  <input name='codice_cliente' type='hidden' value='022028' />
  <input name='codice_amministratore' type='hidden' value='' />
  <!-- pulsante per consentire l'inoltro -->
```

```

        <input type='submit' value='Invia richiesta di modifica' />
</form>
</body>
</html>

```

Come esempio perfettamente equivalente, ecco come la medesima richiesta si può fare con una chiamata a cURL, evitando l'uso di un browser e dello HTML:

```

curl 'https://test.mygelsia.it/API/registrazione/15159' -H 'Host:
test.mygelsia.it' -H 'Content-Type: application/x-www-form-urlencoded'
--data
'username=luca%40accomazzi.net&password=tvumdb&group=registrati&bolletta
=email&codice_fiscale=CCMLST64H08F952C&newsletter=on&confermato=1&in_att
esa=0&attivato=1&telefono=3497680116&indirizzo=Via+Scarlatti&civico=3&ca
p=20015&comune=Parabiago&provincia=Mi'

```

Come si intuisce dall'esempio, nelle sezioni anagrafiche Sar-At non vi permette di modificare la username o la password. Ovviamente, vi permette di modificare solo la anagrafica del cliente le cui credenziali avete passato e non quelle degli altri. Un po' meno ovviamente, potrebbero esistere campi dei quali non vi è concesso effettuare l'aggiornamento (per esempio il codice cliente, o la classe di sconto applicata al cliente di cui state editando l'anagrafica): se passate uno di questi campi Sar-At si limita a ignorarlo (la richiesta non va in errore).

L'amministratore del sito imposta per ciascun campo il permesso di inserirvi dei valori alla creazione o di editare i valori per i record esistenti usando la linguetta Struttura, facendo clic sul nome del campo e operando sulle caselle del gruppo "I visitatori del sito possono..."

#### Ricerca nel database

Il client richiede una ricerca nel database chiamando la URL

`http(s)://www.nome-del-sito.suffisso/API/SarAtSearch/`

Per esempio, sul sito [test.mygelsia.it](https://test.mygelsia.it) useremmo

`https://test.mygelsia.it/API/SarAtSearch/`

Come parametri, il client passa un parametro chiamato *table* col nome della tabella dentro la quale cercare e una (o più) coppie di nome campo, valore campo per indicare cosa cercare. Per esempio, per cercare gli pneumatici per auto di marca Bridgestone nella tabella prodotto useremmo `?table=prodotto&marca=Bridgestone&tipo=auto`

Il server risponde con un vettore in formato JSON di tutte le id dei record che rispondono alla ricerca.

Vediamo un esempio concreto. La app MyGelsia, avendo caricato il profilo cliente usando il metodo GET visto all'inizio di questo capitolo, decide di caricare l'elenco delle utenze del cliente Luca Accomazzi. Deve dunque fare una ricerca nella tabella IS\_UTE alla ricerca di tutti i record che fanno capo al codice cliente 192956.

Anche in questo caso per semplicità vediamo un HTML che dimostra come effettuare la chiamata:

```

<!DOCTYPE html>
<html lang='it'>
<head>
    <meta charset='utf-8' />

```

```

    <title>Ricerca utenze del cliente</title>
</head>
<body>
<!-- post https per una ricerca sicura con risultati mutevoli -->
<form action='https://test.mygelsia.it/API/SarAtSearch/0' method='post'>
    <!-- credenziali -->
    <input name='username' type='hidden' value='luca@accomazzi.net' />
    <input name='password' type='hidden' value='tvumdb' />
    <input name='group' type='hidden' value='registrati' />
    <!-- dove cercare -->
    <input name='table' type='hidden' value='IS_UTE' />
    <!-- cosa cercare -->
    <input name='AUCCLI' type='hidden' value='192956' />

    <!-- solo per la demo -->
    <input type='submit' value='Invia richiesta di ricerca' />
</form>
</body>
</html>

```

Equivalente cURL:

```

curl      'https://test.mygelsia.it/API/SarAtSearch/0'      --data
'username=luca%40accomazzi.net&password=tvumdb&group=registrati&table=IS
_UTE&AUCCLI=192956'

```

La app a questo punto può eseguire un ciclo sugli id così ottenuti e ricevere di conseguenza tutte le informazioni sulle utenze del cliente (per esempio, casa e ufficio). Seguirà un'altra ricerca sulla tabella IS\_FOR alla ricerca di ciascuna fornitura in capo alle utenze (per esempio, gas e luce di casa).

## Codici di errore

Come anticipavamo, ogni richiesta al server riceve innanzitutto un codice di esito nella forma stabilita dal protocollo HTTP: 200 in caso di successo, un codice di errore altrimenti. Ecco un esempio usando cURL:

```

curl -i 'https://test.mygelsia.it/API/registrazione/15159?username=luca
@accomazzi.net&password=tvumdb&group=registrati'

```

Ottiene come header:

```

HTTP/1.1 200 OK

```

E come corpo della risposta

```

{"masterId": "15159", "id": "15159", "lastMod": "2016-11-26 11:29:31"}
(eccetera)

```

Per un controesempio, alla richiesta di leggere l'anagrafica di un'altra persona viene opposto un rifiuto. Notate che lo id numerico qui è diverso e corrisponde a un altro utente.

```

curl -i 'https://test.mygelsia.it/API/registrazione/15160?username=luca
@accomazzi.net&password=tvumdb&group=registrati'

```

Ottiene come header:

```

HTTP/1.1 401 Unauthorized

```

Nel corpo della risposta Sar-At riporta un messaggio esplicativo, pensato per aiutare lo sviluppatore durante il suo lavoro a capire i motivi del rifiuto. I messaggi di errore possono venire cambiati nel corso del tempo, a ogni successivo rilascio di Sar-At, e lo sviluppatore non deve fare affidamento su di essi. Il codice rilasciato in produzione per la soluzione clientside deve utilizzare esclusivamente gli indicatori numerici per desumere il risultato della richiesta.

### Possibili codici di errore

La tabella seguente riporta un elenco, esaustivo al momento in cui questo manuale viene scritto, dei messaggi di errore possibili. Per massima compatibilità con le future versioni di Sar-At si raccomanda che lo sviluppatore clientside si tuteli contro la possibile apparizione di altri codici di errore

Codice	Possibili motivi del rifiuto
400 Bad Request	Manca uno dei parametri di base della chiamata
401 Unauthorized	Username o password sbagliata
403 Forbidden	Tentativo di leggere un record che il proprietario del sito ha dichiarato non (ancora) visibile ai visitatori
404 Not Found	Tentativo di leggere (o aggiornare) un record inesistente
409 Conflict	Tentativo di registrare nel database un dato inaccettabile (per esempio un codice fiscale la cui checksum è sbagliata, un CAP inesistente, l'indirizzo email es@dominio-inesistente.it)
500 Internal Server Error	Errore SQL interno
503 Too busy, try again later	Il server sta sostenendo un carico di lavoro inusitabilmente alto e al momento non può soddisfare la richiesta
505 HTTP Version Not Supported	Richiesta di accesso a dati privilegiati attraverso http. Usare invece https

### Validazione

Nel caso dell'errore 409, ovvero quando è stato chiesto di registrare un record nella base dati ma la validazione serverside è fallita, Sar-At restituisce nel JSON un campo "key" con il nome del campo inaccettabile e nel campo "error" un messaggio di errore. Per esempio, eseguendo un POST di un record, se il campo *data\_di\_nascita* obbligatorio ("required") è mancante, la risposta sarà

```
{"error":"Campo obbligatorio mancante", "field":"data_di_nascita"}
```

# Appendice A. Riferimento

Questa sezione riporta un completo elenco dei tag riconosciuti da Sar-At. Se non altrimenti indicato, tutti i tag funzionano con showPage, showForm e showErr (rispettivamente usati per gestire le pagine ordinarie, i moduli e le pagine d'errore).

Tutti i tag di Sar-At sono inclusi tra parentesi graffe e scritti interamente in maiuscolo.

## Campi del record corrente

Se nel database abbiamo definito una sezione chiamata “automobili” in cui ogni record contiene i campi “marca”, “modello” e “cilindrata”, allora scrivere nella pagina mastro {marca} provoca l'apparizione della marca del modello attualmente selezionato.

## Pseudo-campi predefiniti

Oltre ai campi della sezione esistono alcune scritture che Sar-At risolve in ogni sezione. Sono: {HEAD} (valida intestazione per la pagina, comprese keyword, description, alfabeto, lingua, generator e titolo), {template} (nome della sezione corrente), {sarel\_username} (identificatore univoco per i visitatori registrati e loggati), {search} (valore dell'omonimo parametro nella URL, se presente), {id\_field} (valore dell'omonimo parametro nella URL, se presente), {sarat\_adminname} (identificatore univoco per gli amministratori registrati e loggati)

## INCLUDE

Descritto compiutamente nel paragrafo “Inclusione di frammenti HTML” del capitolo 6, “Funzioni avanzate”. In sintesi: legge un file esterno, che tipicamente (ma non necessariamente) contiene un frammento di pagina HTML, e lo include nel punto in cui viene trovata l'istruzione INCLUDE stessa. Utilizzato per isolare in un unico punto un brano ripetuto in molte pagine mastro del sito. Il brano può contenere a sua volta istruzioni Sar-At (ma la istruzione INCLUDE viene sempre eseguita per prima, quindi un INCLUDE non può a sua volta contenere altri INCLUDE).

Gli INCLUDE possono venire localizzati. Per esempio, una pagina mastro chiamata pippo.html.it che contiene la riga {INCLUDE pluto.ihtml} finirà per includere automaticamente il file pluto.ihtml.it se questo esiste; pluto.ihtml in caso opposto).

Esempio:

```
{INCLUDE testata.ihtml}
```

## BEGIN..END

Ne abbiamo parlato nel capitolo sulle Viste. In sintesi: tutto il codice HTML che si trova tra {BEGIN etichetta\_vista} e {END etichetta\_vista} viene ripetuto da Sar-At tante volte quanti sono i record selezionati dalla vista. Per esempio {BEGIN tutti\_i\_tenori}{nome} {cognome}<br />{END tutti\_i\_tenori} fa apparire nella pagina web qualcosa come

```
Aristide Palazzi<br />
```

```
Ruggero Leone<br />
```

## NAVIGATE...END

Ne abbiamo parlato nel capitolo sulle Viste. Funziona come BEGIN..END visto sopra, ma dividendo i risultati in sottopagine. Proseguendo l'esempio visto sopra, potremmo decidere di mostrare i cantanti lirici di timbro tenorile a due per volta.

## IF per i campi della pagina

Immaginate di voler pubblicare sul vostro sito web una sezione catalogo, che ospiterà le schede dei prodotti della vostra azienda. Disgraziatamente, voi non disponete delle fotografie promozionali di alcuni prodotti, ma volete certamente utilizzare le foto di cui disponete. Come creare una pagina mastro che si adatta a questa variabilità? Una possibilità sta nel creare più pagine mastro distinte (vedere “varianti” in questo manuale), ma c'è una soluzione più semplice.

Preparate le foto dei prodotti di cui disponete, e inseritele nella base dati Sar-At. Quando create la pagina mastro, lascerete uno spazio per la immagine. In HTML avremo una riga tipo questa:

```
<img src = '{foto}' width='100' height='200' />
```

Potete ritoccare la pagina mastro in modo che lo spazio dell'immagine venga rimossa quando l'immagine manca in base dati, così:

```
{IF foto}
<img src = '{foto}' width='100' height='200' />
{ENDIF foto}
```

Quando Sar-At crea la pagina web di un prodotto, rimuoverà tutto lo HTML contenuto nel blocco IF se la informazione indicata ("foto", nel caso dell'esempio) è nulla oppure zero. È anche possibile creare un blocco "altrimenti", così:

```
{IF foto}
<img src = '{foto}' width='100' height='200' />
{ELSE foto}
<img src = 'logoSocieta.gif' width='100' height='200' />
{ENDIF foto}
```

In questo ultimo esempio, la foto verrà sostituita da una immagine fissa quando una foto non è disponibile.

### IF campo = valore

È possibile testare il valore di un campo. Per esempio possiamo scrivere {IF timbro = tenore}. Usate spazi bianchi attorno al segno di uguale. I test per valore minore o maggiore non sono per ora supportati.

### IF fuori dalle viste

Il costrutto IF-ELSE-ENDIF può venire utilizzato in congiunzione con il nome di un campo, come nell'esempio qui sopra, o con il nome di una vista. Quando viene usato con una vista, il ramo IF verrà inserito nella pagina web se la vista seleziona almeno un record e il ramo ELSE se la vista non seleziona alcun record.

Per esempio, vediamo come creare un semplice motore di ricerca interno al nostro sito, che seleziona tutte le pagine di una sezione il cui titolo contiene una parola scelta dal visitatore.

1. Creiamo una nuova Vista. Le diamo un nome a piacere e per etichetta scegliamo "motore\_ricerca".

Impostiamo la Vista perché selezioni in soli record il cui titolo contiene un valore scelto dal visitatore, che la pagina HTML passerà nel parametro *search*.

2. Creiamo la interfaccia utente del motore di ricerca usando lo HTML. Ecco una proposta minimalista:

```
<form action="showPage.php" method="post">
  <input type="hidden" name="template" value="articoli" />
  <input type="hidden" name="masterPage" value="trova.html" />
  <input type="text" name="search" size="30" />
  <input type="submit" name="Trova" />
</form>
```

3. Creiamo la pagina mastro "trova.html" destinata ad ospitare l'elenco di pagine trovate. Ecco una possibile implementazione:

```
{IF motore_ricerca}
  <p>Elenco dei prodotti il cui nome contiene la parola {search}:</p>
  {BEGIN motore_ricerca}
  <p>{title}</p>
  {END motore_ricerca}
{ELSE motore_ricerca}
  <p>Spiacente, nessun prodotto contiene la parola {search} nel nome.</p>
{ENDIF motore_ricerca}
```

Se vogliamo inserire un link a ciascuna pagina trovata, la prima parte del nostro esempio diventa:

```
{IF motore_ricerca}
  <p>Elenco dei prodotti il cui nome contiene la parola {search}:</p>
  {BEGIN motore_ricerca}
  <p><a href='showPage.php?template=prodotti&id={id}'>{title}</a></p>
  {END motore_ricerca}
{ELSE motore_ricerca}
eccetera.
```

Se vogliamo preparare una pagina che cerca la parola digitata dal visitatore in più sezioni del sito possiamo ottenere il risultato creando (al punto “1” di questo paragrafo) più viste. Ogni vista lavorerà su una sezione distinta: per esempio avremo una vista “motore\_ricerca1” che opera sulla sezione “prodotti”, una vista “motore\_ricerca2” che lavora sulla sezione “risposte” e una vista “motore\_ricerca3” che lavora sulla sezione “recall”.

Al punto 3, nella pagina che mostra i risultati della ricerca, concateneremo le Viste.

```
{IF motore_ricerca1}
  <p>Elenco dei prodotti il cui nome contiene la parola {search}:</p>
  {BEGIN motore_ricerca1}
  <p><a href='showPage.php?template=prodotti&id={id}'>{title}</a></p>
  {END motore_ricerca1}
{ENDIF motore_ricerca1}
{IF motore_ricerca2}
  <p>Elenco delle risposte il cui titolo contiene la parola {search}:</p>
  {BEGIN motore_ricerca2}
  <p><a href='showPage.php?template=risposte&id={id}'>{title}</a></p>
  {END motore_ricerca2}
{ENDIF motore_ricerca2}
{IF motore_ricerca3}
  <p>Elenco dei recall il cui nome contiene la parola {search}:</p>
  {BEGIN motore_ricerca3}
  <p><a href='showPage.php?template=recall&id={id}'>{title}</a></p>
  {END motore_ricerca3}
{ENDIF motore_ricerca3}
```

## IF dentro alle viste

L'istruzione IF può anche venire usata dentro alle Viste, e in questo caso fa riferimento ai campi dei record ripescati dalla Vista. Per esempio, se stiamo guardando via showPage una pagina web della sezione film, dunque relativa a un DVD, e se vi usiamo una Vista per ripescare i dati sugli interpreti, possiamo scrivere:

```
Titolo del disco: {title}
{BEGIN vista_che_trova_interpreti}
{IF sesso}Attrice{ELSE sesso}Attore{ENDIF sesso} {nome} {cognome}<br />
{END vista_che_trova_interpreti}
```

## LOCATION

Seguito da un indirizzo http, viene utilizzato per inviare il visitatore ad un'altra pagina. Per esempio, un tipico codice piazzato nelle pagine web che impaginano il risultato di una ricerca è:

```
<h1>Risultato della ricerca</h1>
{IF ricerca_sul_sito}
  [...mostra risultati...]
{ELSE ricerca_sul_sito}
Spiacente, non abbiamo trovato nulla
{ENDIF ricerca_sul_sito}
```

Se si desidera portare il visitatore che ha chiesto una ricerca senza successo ad una pagina completamente richiesta è possibile modificare il brano come segue:

```
<h1>Risultato della ricerca</h1>
{IF ricerca_sul_sito}
    [...mostra risultati...]
{ELSE ricerca_sul_sito}
{LOCATION mappa-del-sito.html}
{ENDIF ricerca_sul_sito}
```

## Costrutti condizionali sui moduli

Quattro istruzioni sono definite esclusivamente dentro showForm, il motore dei moduli. Sono presentate nel capitolo sui moduli di questo stesso manuale.

### ONERR..ENDERR

Lo HTML contenuto tra ONERR e ENDERR viene lasciato se c'è un errore di compilazione relativo al campo citato, altrimenti rimosso. Esempio;

```
{ONERR codice_fiscale}Codice fiscale assente o non valido!{ENDERR
codice_fiscale}
```

### CHECKBOX

Genera una casella di controllo (*checkbox*) selezionata se il campo indicato ha valore "on". Esempio:

```
{CHECKBOX consenso_legge_privacy}
```

Se state usando Sar-At versione 7.5 o superiore esiste una versione più potente di questo tag, che è indicata per i checkbox multipli (esempio: come si compone il vostro pasto ideale? Seguono checkbox per antipasti, primi, secondi, contorni, dolce e frutta)

La sintassi è analoga a quella di SELECTEDIF, ma riflette il fatto che stiamo trattando una serie di valori:

```
{CHECKBOX nomecampo[ ] valore}
```

Quindi avremo per esempio:

```
<label>{CHECKBOX pietanza[ ] primi} Primi piatti</label>
<label>{CHECKBOX pietanza[ ] secondi} Secondi piatti</label>
<label>{CHECKBOX pietanza[ ] contorni} Contorni</label>
```

CHECKBOX può venire usato all'interno di una Vista. Per esempio avremmo:

```
{BEGIN tutte_pietanze}
<label>{CHECKBOX pietanza [ ] {stringa}} {title}</label><br />{END tutte_pietanze }
```

Le pagine mastro automatiche generate da Sar-At per le sezioni modulo contengono una struttura altamente ottimizzata come quest'ultima, generata automaticamente, a fronte dei campi di tipo "caselle di spunta" inserite nel modulo. Consigliamo dunque di generare i moduli in Sar-At e poi incollare lo HTML risultante dentro un impaginato.

### RADIOBUTTON

Genera un pulsante radio selezionato se il campo menzionato al secondo parametro ha valore uguale al terzo parametro. Esempio:

```
{RADIOBUTTON personaggio_disney Pippo}
```

### SELECTEDIF

Genera l'attributo XHTML selected se il campo indicato come secondo parametro ha valore uguale al terzo parametro. Esempio:

```
<select name='personaggio'>
    <option {SELECTEDIF personaggio Pippo}>Pippo</option>
    [...]
```

### PHP

Dentro una pagina mastro è possibile inserire codice in linguaggio PHP. Il brano di codice deve venire prefisso da <?php e terminato con ?>

Non è sufficiente la notazione abbreviata <?, neppure se il server è configurato per accettarla normalmente.



Sar-At non considera la presenza di codice PHP nella pagina mastro come limitativa dell'uso di cache, quindi il codice non viene eseguito quando della pagina viene spedita al visitatore la copia archiviata. Il codice PHP viene eseguito da Sar-At e non direttamente da Apache. Il risultato degli *echo* viene visualizzato nel punto in cui si trova il codice PHP.

### Variabili predefinite per le Viste

Sono dettagliate nel capitolo sulle Viste. Includono i parametri passati nella URL e i valori dei campi del record corrente.