Sviluppo Web con Rails

Esploriamo le caratteristiche del framework MVC Ruby on Rails

Ruby on Rails (anche detto RoR o semplicemente Rails, http://rubyonrails.org) e' stato uno dei fautori dell'enorme crescita e diffusione del giovanissimo linguaggio di programmazione Ruby.
Rails e' stato un vero e proprio terremoto nell'ambito del web development: la sua semplicita' d'uso ha reso questo settore accessibile (anche troppo...) a tutti e con poca difficolta'. RoR permette con estrema semplicita' di recuperare o inserire dati all'interno di database, di creare facilmente delle pagine (X)HTML, di integrare funzionalita' che caratterizzano applicazioni web moderne, come ad esempio **AJAX**.
Per capire quanto Rails sia stato apprezzato dalla comunita' informatica, basta pensare a quanti framework analoghi sono nati in breve tempo: Cake per PHP, Trails per Java, Turbogears e Subway per Python e molti altri.
Nonostante cio', Rails non introduce comunque nulla di innovativo, ma semplicemente racchiude tutte le tecniche di programmazione gia' sperimentate in passato all'interno di un nuovo modello di sviluppo. Tutto cio' ha ridotto drasticamente i tempi di sviluppo di un'applicazione e parallelamente anche le righe di codice necessarie a realizzarla (anche grazie alla potenza del linguaggio su cui si basa: ruby).

Cio' che non abbiamo ancora detto di Rails e' che si tratta di un framework **MVC** (*Model View Controller*). L'MVC e' un pattern architetturale introdotto originariamente da Smalltalk e implementato oggi non solo da Rails, ma anche da moltissimi altri framework per il web development (ad esempio Django, un framework Python, o Structs, scritto in Java).
L'acronimo MVC suggerisce quali sono i componenti principali di un'applicazione che segue questo pattern: i modelli, le viste ed i controller.
I **modelli** (che implementano la business logic) definiscono i dati e le operazioni che si possono compiere su questi, ed espone al controller le funzionalita' necessarie per il loro aggiornamento. Le **viste** (logica di presentazione) si occupano della costruzione della GUI (Graphical User Interface), ovvero dell'interfaccia grafica dell'applicazione che rappresenta il mezzo mediante il quale gli utenti interagiranno con il sistema. I **controller** invece si occupano di trasformare le operazioni compiute dall'utente con la vista in azioni eseguite dal modello. Il controller realizza quindi il mapping fra l'input dell'utente e le azioni di business logic, rappresentando quindi la vera e propria logica di controllo.

Installazione del framework

Il framework Rails puo' comodamente installato dal gestore di pacchetti ruby **gem**. Questo non e' l'unico metodo di installazione possibile, ad esempio potremmo sfruttare il gestore di pacchetti della nostra distribuzione o ancora ricorrere agli archivi .zip o .tgz disponibili su http://rubyforge.org.

Cartella	Scopo
app	Contiene il cuore vero e proprio dell'applicazione. All'interno delle apposite sottocartelle troveremo i modelli, i controller, le viste e gli helper.
config	Contiene i file di configurazione di 3 aspetti: le rotte, l'accesso al database e gli environments
db	In questa cartella vengono mantenute informazioni sui database e dei particolari file ruby (detti migration) che permettono modifiche alle loro tabelle.
doc	Contiene la documentazione del progetto
lib	Destinata a librerie non prettamente legate al lato web dell'applicazione
log	Contiene i log del web server
public	E' la web root, il posto in cui inseriamo eventuali file HTML statici, immagini, file javascript e fogli di stile.
script	Contiene utili script, ad esempio: • *console:* un'istanza di irb in cui vengono messi a disposizione tutti i modelli e i controller della nostra web application • *generator:* genera gli scheletri di modelli, controller e viste, migration • *server:* e' un piccolo web server scritto in ruby (Webrick) che consente di testare al volo il nostro software. In alternativa a Webrick e' possibile utilizzare Mongrel (presente nei repository di rubygems) o qualsiasi altro server Web come ad esempio Apache o Lighttpd. • *plugin:* per l'installazione di librerie di terze parti
test	E' qui che vengono inseriti i test sui vari componenti dell'applicazione
tmp	E' una cartella temporanea
vendor	Contiene librerie di terze parti e plugin

In figura e' possibile vedere la struttura di un'applicazione Rails.

Una volta installato il framework e' sufficiente lanciare il comando:

$ rails nome_applicazione

per far si che rails generi lo scheletro dell'applicazione.

Nella prossima puntata vedremo come creare una piccola applicazione con rails

Stay tuned!