Introduzione a Rails

Rails è un framework per lo sviluppo di applicazioni web basate sul pattern **MVC**. Rails favorisce la metodologia di sviluppo detta **convenzione sopra configurazione**, in altre parole il framework fa delle assunzioni su come vogliamo sviluppare e su come le vogliamo sviluppare, piuttosto che costringerci a definire numerosi file di configurazione per descrivere il comportamento di ogni singolo meccanismo del framework.

Rails ha inoltre una visione del pattern MVC leggermente differente rispetto altri framework come ad esempio Struts. In particolare per quanto riguarda il MODEL lo strato del CONTROLLER lavora sul MODEL tuttavia nella maggioranza dei casi delle classi Active Record, invece per quanto riguarda lo strato del CONTROLLER, per rimanendo un layer di connessione tra MODEL e VIEW, esplicita gran parte della business-logic che normalmente implementiamo nell'applicazione.

Componenti fondamentali

Rails viene distribuito in una libreria gem, che è costituita da quattro componenti fondamentali:

1)ActionPack
   (a) ActionController
   (b) ActionDispatch
   (d) ActionView

2)ActiveRecord

3)ActiveModel

4)ActiveRecord

5)ActiveResource

6)ActiveSupport

In questa piccola guida non farò altro che spiegare il ruolo di questi componenti, sviluppando una serie di esempi, dedicati durante lo sviluppo di un'applicazione di social-networking (twitter-clone) seguendo una guida ufficiale : http://www.railstutorial.org

MODEL

Nello strato del MODEL, vanno a finire tutte le classi del nostro dominio di business, le cosiddette classi entity, mentre il più delle volte la business-logic viene scritta nello strato del controllore. Nessuno comunque ci impedisce di creare un layer business-logic per i casi d'uso più complessi che coinvolgono diverse classi entity.

In ogni caso lo strato del MODEL è fortemente influenzato dal **pattern Active Record** e dalle **Migrations**. Rails favorisce il progetto del dominio di business e del modello dei dati come un processo unico, supportato dalle Migrations, che ci permettono di definire lo schema dei dati senza conoscere DDL-SQL, e di gestire l'evoluzione dello schema senza ricorrere a istruzioni SQL, ma solo grazie agli script Ruby.