Guida Ruby on Rails: l'ambiente di lavoro e la nostra prima app

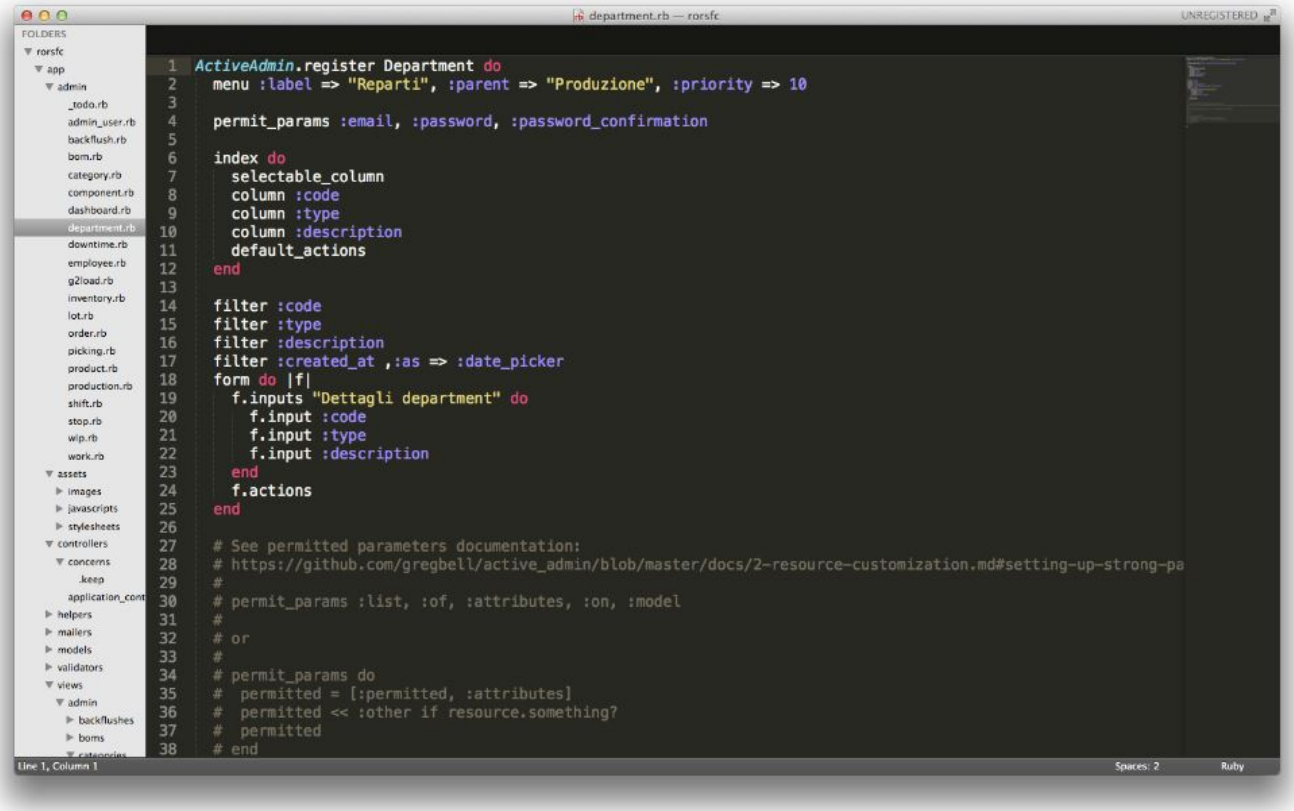
Nel corso del secondo articolo di questa guida, ci occuperemo della configurazione dell'ambiente di lavoro scegliendo, nel dettaglio, un editor e un sistema di versioning. Inoltre, per i più desiderosi di azione, daremo vita alla nostra prima semplicissima app. Allora bando alle ciance e mettiamoci al lavoro!

Scegliere l'editor

La scelta dell'editor è senz'altro un passo fondamentale nella configurazione del nostro ambiente di lavoro. Necessitiamo, infatti, di un software che ci permetta di concentrarci sulla scrittura del codice, facilitandoci compiti ingrati come quello dell'organizzazione dei file e del rispetto della sintassi. Per questo motivo, pur ritenendo la scelta dell'editor un fattore strettamente personale, trovo fondamentale scegliere uno strumento che rispetti, possibilmente, questi parametri:

• **Identare e reidentare automaticamente il codice Ruby**, in quanto risulta essere il metodo migliore per individuare testo erroneamente nidificato.
• **Navigare tra i file**, in quanto, come vedremo, un'applicazione Rails può arrivare a contenere tantissimi file in almeno 34 directory differenti; quindi, potersi muovere agevolmente da una cartella all'altra è basilare.
• **Evidenziare la sintassi Ruby**, in quanto molti dei file che utilizzeremo avranno l'estensione .erb (incorporano all'interno dell'HTML snippet di codice Ruby).
• **Completare automaticamente i nomi**, in quanto alcuni nomi possono risultare ostici o lunghi, quindi digitare solo le prime lettere per ottenere un suggerimento è spesso di grande aiuto.

Detto questo, mi sento di raccomandarti due strumenti che ho avuto modo di sperimentare a lungo: **TextMate** e **SublimeText**. Puoi scaricare entrambi gratuitamente dai rispettivi siti web linkati poc'anzi. Ecco come appare SublimeText al lavoro su un progetto RoR:



Version Control

Il Version Control è un sistema ampiamente utilizzato in informatica per gestire diverse versioni dello stesso progetto. In Rails è importantissimo controllare ogni modifica attraverso i test: per questo motivo, nel corso della guida, faremo ampio utilizzo di **Git, il quale ci permette di spostarci agevolmente tra le varie versioni modificate che caricheremo su di esso**. Ti consiglio, nel caso non li conoscessi già, di imparare i comandi base di Git come init, clone, add ed altri; per far ciò puoi basarti sull'ottima documentazione ufficiale. I più esperti di voi potrebbero apprezzare un sistema di project managment come **Redmine**, configurabile con svariati tipi di version control tra i quali Git, il quale permette, oltre a creare nuove copie ad ogni cambiamento portando al minimo la possibilità di danni accidentali, di condividere alcune versioni con i collaboratori e altre direttamente con i clienti che potranno "giocare" con le funzioni appena implementate nel progetto.

Database

I database compatibili con Rails sono numerosi. Puoi, infatti, scegliere di utilizzare MySQL, Oracle,

Postgres, DB2 e altri ancora, tuttavia nel corso di questa guida faremo riferimento unicamente a SQLite 3. Qualora decidessi di optare per altre soluzioni, ricorda che nei forum di supporto troverai numerose e dettagliate guide su come installare e configurare database differenti da SQLite!

Creiamo la nostra prima app

Iniziare un nuovo progetto in Rails è veramente facile!

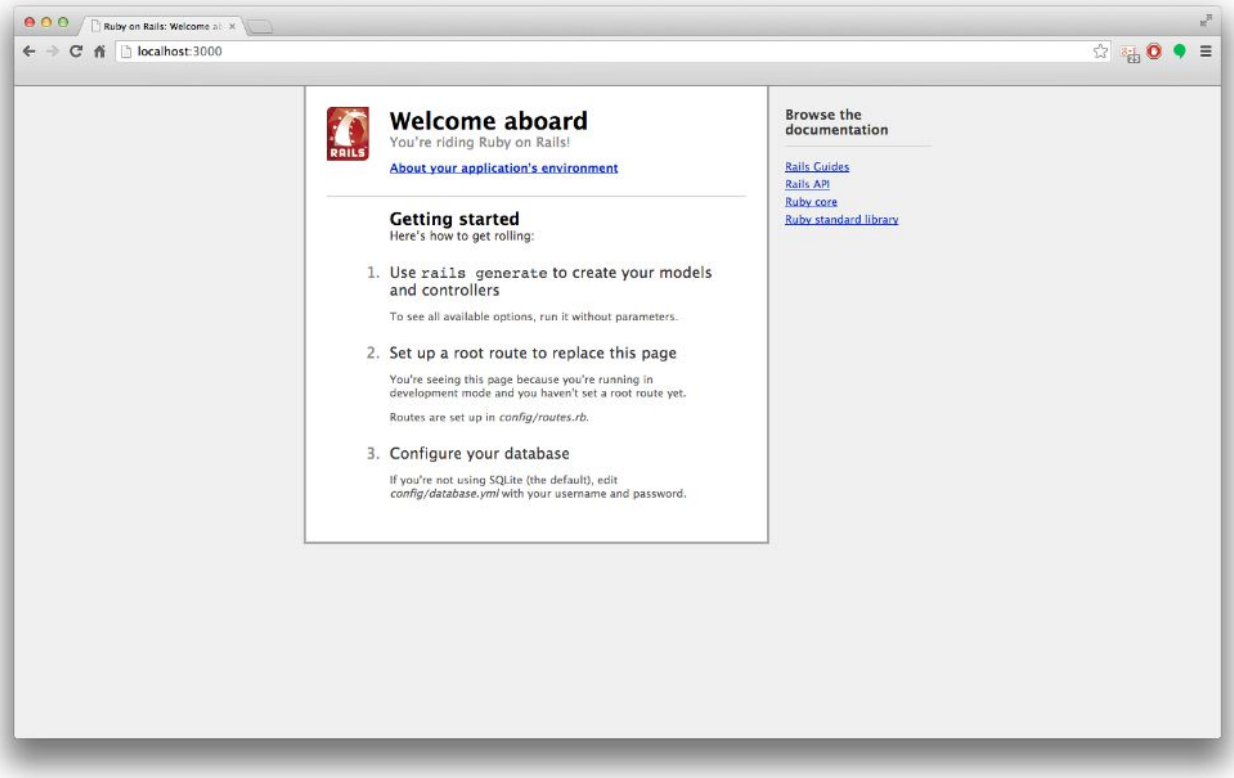
Come prima cosa creiamo una nuova cartella (in questo esempio denominata “prova rails”) nella posizione che preferiamo e dirigiamoci al suo interno dal terminale digitando:

A questo punto non ci resta che digitare:

per creare ufficialmente la nostra prima applicazione, denominata per l’appunto app1, all’interno della directory appena creata. Come possiamo vedere, attraverso il comando:

il numero di directory che contiene un progetto Rails appena creato può scoraggiare e confondere i neofiti. Non aver paura, al momento lavoreremo esclusivamente nella cartella app per poi approfondire, progressivamente, i diversi compiti delle altre. Sempre da terminale, all’interno della nostra nuova cartella app1, digitiamo:

per far partire il server all’indirizzo http://localhost:3000 e visualizzare la seguente pagina, pronta a indicarci che tutto funziona a dovere.



Adesso la nostra applicazione è funzionate e attiva ma, nonostante ciò, non contiene alcun codice! Provvediamo subito creando un banale e sempre di moda "*Hello, World!*".

Come approfondiremo nel corso delle prossime lezioni e come abbiamo accennato nell'articolo introduttivo, Rails è un framework Model-View-Controller, quindi, una volta acquisite le richieste dal browser, Rails le decodifica in cerca del relativo controller e richiama un'azione al suo interno per poi invocare una particolare vista che mostri i risultati all'utente. Può sembrare a prima vista un meccanismo complesso ma, in realtà, Rails si occupa automaticamente di tutto ciò, lasciando a noi l'unico compito di definire controller e view.

Cominciamo creando il controller. La procedura, anche in questo caso, è semplice. Assicurandoci di trovarci nella cartella dell'applicazione, basta digitare all'interno del terminale:

Analizziamo passo passo il comando appena inserito. Abbiamo detto a Rails di generare un controller denominato *Say*, che contiene due azioni: *hello* e *goodbye*. Possiamo visualizzare il controller appena creato aprendo, nel nostro editor, il file app/controllers/say_controller.rb. All'interno troveremo:

ovvero la definizione della classe *SayController* popolata da due azioni al momento vuote: *hello* e *goodbye*. Insomma, proprio quello che ci aspettavamo e che abbiamo ottenuto digitando un singolo comando!

Conclusioni

Ricapitolando, nel corso di questo articolo abbiamo elencato tutti gli strumenti utili alla gestione dei nostri progetti Rails e abbiamo creato la nostra prima applicazione che, al momento però, non svolge alcuna azione. Nel corso della lezione successiva, impareremo come aggiungere un'azione rendendo la pagina dinamica alle richieste dell'utente.

Come sempre, puoi usare questo link per scaricare il codice completo di questa lezione.

GUIDA RUBY ON RAILS: INDICE LEZIONI