Ruby on Rails

Wechseln zu:Navigation, Suche

Inhaltsverzeichnis

1 Definition

(Ruby on) Rails ist ein freies, leichtgewichtiges Web-Framework auf Basis der Programmiersprache Ruby. Es erlaubt die schnelle Entwicklung von datenbankgestützten Web-Applikationen nach dem Model-View-Controller-Paradigma.

2 Philosophie

Convention over Configuration: Es muss nur das konfiguriert werden, was nicht den konventionen enspricht.

Don't repeat yourself: Redundanz soll vermieden werden. So ist es zum Beispiel möglich das Aussehen der Seite in einem sog. Layout zu definieren. Dies geschieht dann an nur einer Stelle.

3 Bemerkungen

Ruby on Rails wurde von David Heinemeier Hansson bei der Firma 37signals entwickelt. Viele Web 2.0 Sites bzw. Anwendungen basieren auf Ruby on Rails.

4 Projekt erzeugen:

```
rails --database <DBTYPE> ApplicationName
```

Erzeugte Verzeichnisse und Dateien:

app - Enthält Komponenten der Application
- controllers - Controller-Klassen, bearbeitet Benutzer-Anfragen
- models - Datenmodell und Schnittstelle zur Datenbank
- views - Templates, für die Erzeugung von Ergebnis-Dokumenten
- helpers - Hilfsklassen

config - Konfigurationsdateien
db - Datenbankk-Kassen
public - Ordner statische Dateien
test - Ordner für automatische Tests
lib - Ordner für Rails-Bibliotheken
vendor - Ordner für Bibliotheken von Dritt-Anbietern
doc - Zielordner für die von RubyDoc generierte Dokumentation
log - Logging-Dateien
script - Tools für rails (z.B. Server-Start)
tmp - Temporäre Dateien
README
Rakefile - Ruby Make, ein Werkzeug zum Erstellen, Packen und Testen von Rails-Code

5 Das MVC-Paradigma

5.1 Controller und View anlegen

Um einen neuen Controller und View zu erstellen kann das `generate`-Skript verwendet werden.

```
ruby script/generate controller ControllerName ActionName
```

Nun werden die Dateien für den Controller und die dazugehörigen View erzeugt.

5.2 Model anlegen

Für die Erstellung eines Models wird ebenfalls das `generate`-Skript verwendet.

```
ruby script/generate model ModelName
```

Mit diesem Befehl wird eine Datei für das Model und eine „Migration“-Datei zu Erstellung der zugehörigen Datenbank erzeugt.

5.3 View

Die View ist die Schnittstelle zum Benutzer. Sie erzeugt i. Allg. HTML-Code, der in einem Webbrowser angezeigt wird.

Alle Views befinden sich im Verzeichnis `app/view`. Dort liegen die zu den einzelnen Actions gehörenden Views in Unterordnern mit dem Namen des zugehörigen Controllers.

Vor der Version 2.0 war die Dateiendung von Views `.rhtml`, aktuell ist allerdings `.html.erb`.

Um ein allgemeines Design zu ermöglichen, kann mit so genannten Layouts gearbeitet werden. So können Elemente wie Footer, Header und Navigation, die seitenübergreifend gleich bleiben, ausgelagert werden.

In den View-Dateien kann der Entwickler mit Ruby dynamisch erzeugte Elemente einbinden. Die Tags **<% %>** und **<%= %>** werden verwendet um Ruby-Code ausführen zu lassen. Die erste Variante wertet nur aus, die Zweite wertet aus und fügt den zurückgegebenen Text in die zugehörige HTML-Seite ein.

Darüber hinaus gibt es von Rails bereitgestellte Ruby-Funktionen um Tags zu erzeugen. Das einfachste Beispiel ist die Erstellung eines Links :

```
<%= link_to "Login" , :action => :login %>
```

Das Ergebnis dieses Aufrufs im HTML sieht wie folgt aus:

```
<a href="/webshop/login">Login</a>
```

5.4 Model

Das Model dient im Allgemeinen zum Zugriff auf externe Daten, wie z.B. Datenbanken und Dateien, um sie in einem Ruby on Rails Projekt verwenden zu können. Alle Models sind in dem Verzeichnis */app/model* abgelegt.

Der Standardfall ist, dass der Zugriff auf eine Tabelle in einer Datenbank durch ein Model realisiert wird. Bei der Erstellung eines solchen Models ist es Konvention, dass die Tabelle im Plural und das Model und die Klasse für die Tabelle im Singular erzeugt werden. Es werden mehrere Datenbanken unterstützt, eine genaue Liste ist auf der Wikipedia Seite zu finden. Die Daten für den Datenbankzugriff sind in der Datei *config/database.yml* gespeichert. Mit dem Rakebefehl

```
rake db::create
```

kann die Datenbank erstellt werden.

In der Migration-Datei kann das Create-Skript für die Tabelle erstellt werden. Ein Beispiel für solch eine Migration-Datei wäre:

```ruby
class CreateCarts < ActiveRecord::Migration
  def self.up
    create_table :carts do |t|
      t.column :customer_id, :integer
      t.column :product_id, :integer
      t.column :quantity, :integer
      t.timestamps
    end
  end
  def self.down
    drop_table :carts
  end
end
```

Um die Create-Skripts in SQL umzusetzen und auf der Tabelle auszuführen dient der Rake-Befehl

```
rake db::migrate
```

Die Klasse einer Tabelle wird von der Klasse `ActiveRecord::Base` abgeleitet. Diese `ActiveRecord::Base`-Klasse enthält und erzeugt automatisch Funktionen, um mit der Tabelle ähnlich arbeiten zu können wie mit einem Objekt. Die erstellten Funktionen sind:

Für jede Spalte eine Funktion zum Zugriff und Setzen des Wertes z.B. `Cart.quantity`.
Eine save-/create-, update- und destroy-, delete-/drop-Funktion für die Tabelle.
`find(ID)` und `find(.all)` zum Suchen in der Tabelle.

Mit Hilfe der Parameter `belongs_to :<Tabelle_im_Singular>`, `has_one :<Tabelle_im_Singular>` (1:1), `has_many :<Tabelle_im_Plural>` (1:n) und `has_and_belongs_to_many :<Tabelle_im_Plural>` (m:n) können die Relationen (Fremdschlüsselbeziehungen) in einer Datenbank auf das Model abgebildet werden. Belongs_to muss in das Model der Tabelle eingefügt werden, in der der Fremdschlüssels steht. Die anderen zwei Befehle `has_one` und has_many geben in dem Model, auf das sich der Fremdschlüssel bezieht, an, wie oft es in der Relation vorkommen darf. `has_and_belongs_to_many` muss in beide Models gleich eingetragen werden. Wichig hierbei ist hierbei zu beachten, wann die Tabelle im Singular oder wann im Plural geschrieben werden muss.

6 Module

Für Ruby on Rails gibt es eine Reihe von Modulen, die die Funktionalität erweitern: Eine PostgreSQL-Anbindung, eine Apache-Anbindung etc.

7 Beispiele

Eine einfache Web-Anwendung: Die Händler-Anwendung, ein Ruby-on-Rails-Beispiel.

8 Quellen

Ruby on Rails Website
Agiles Web Development with Rails
Ruby on Rails 2

9 Siehe auch

Wikipedia: Ruby on Rails
http://www.rubyenterpriseedition.com/

Seiten die mit Ruby on Rails entwickelt wurden:

http://www.twitter.com
http://www.xing.com
http://www.spickmich.de

Dieser Artikel ist GlossarWiki-konform.

In diesem Artikel sollten die Quellenangaben überarbeitet werden.
Bitte die Regeln der GlossarWiki-Quellenformatierung beachten.

Kategorien:
Content-Management-System
Web-Programmierung