Ruby on Rails

Introduction to Rails

- Rails is a server-side web application development framework written in Ruby language.

- It allows you to write less code than other languages and frameworks.

- It includes everything needed to create database-backed web applications according to MVC pattern.

Rails Scripts

- Rails provides some excellent tools that are used to develop Rails application.
- These tools are packaged as scripts for commandline.
- Following are the most useful Rails scripts used in Rails application.
  - WEBrick Web Server
  - Generators
  - Migrations

WEBrick Web Server

- Rails is configured to automatically use WEBrick server. This server is written in pure ruby and supports almost all platforms like Windows, Mac or Linux.

- Rails also supports either of these servers Mongrel or Lighttpd.

Generators

- The Rails include code generation scripts, which are used to automatically generate model and controller classes for an applications.

- By running generator command, skeleton files for all your model and controller classes will be generated.

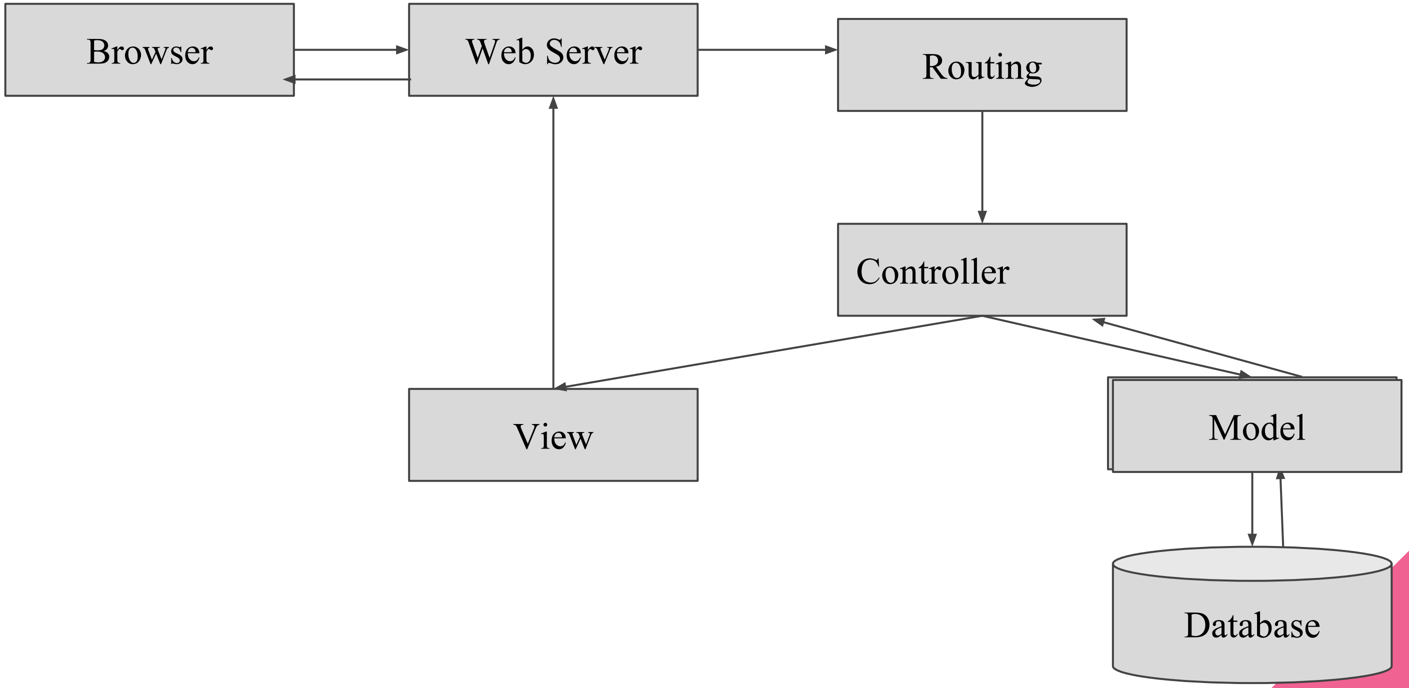
- It also generates, database migration files for each model it generates.

Migrations

- It is a pure ruby code that defines the structure of a database.

- The changes you make to database scheme is isolated in a separate migration file, which has a method to implement or reverse the change.

Rails Architecture



Routing in RoR

- The Rails router recognizes URLs and dispatches them to corresponding controller. It also generates paths and URLs.
  - Basically a router is a way to redirect incoming requests to controllers.
  - Rails handles routing via config/routes.rb file rather than relying on the webserver to control URL routing.

- Request comes to the controller.

- Controller finds an appropriate view and interacts with model which in turn interacts with database and send response to controller.

- The controller gives output to the view based on the response.

Model

- The models are classes in Rails. They interact with database, store data, handles validation, transaction etc.

- This subsystem is implemented in **ActiveRecord** library. This library provide interface between databases tables and ruby program code that implements database records.

- Ruby models method names are automatically generated from database tables field names.

View

- View represent data in a particular format in an application for the users.
- It handles HTML, CSS, JavaScript and XML in an application. They do what controller tell them.
- This subsystem is implemented in **ActionView** library. This library is an **Embedded Ruby(erb)** based system which defines presentation templates for data presentation.

Controller

- Controller directs traffic to views and models.

- It interacts with models for data from the database and display the desired result with the help of view in an application.

- This subsystem is implemented in **ActionController** library. This library is a data broker sitting between **ActionRecord** and **ActionView.**