Ruby on Rails

Arquitetura MVC

Elder Crul
Aramis Fernades

Conteúdo

- O que é Ruby?
- O que é Rails?
- Arquitetura MVC em Ruby on Rails
- Arquitetura dos módulos Rails

O que é Ruby ?

O que é Ruby ?

- É uma linguagem de script interpretada
- Programação orientada a objetos com uma filosofia e sintaxe muito limpa
- Recursos de tratamento de exceções
- Todo dado em Ruby é um objeto

O que é Ruby ?

- Ruby tem um garbage collector que realmente é do tipo marca-e-limpa. Atua em todos os objetos do Ruby
- Não precisa de declaração de variáveis
- Usa a convenção de nomenclatura para delimitar o escopo das variáveis
- Ruby tem um sistema de threading independente do sistema operacional

O que é Rails?

O que é Rails?

- **E um framework livre que promete aumentar velocidade e facilidade no desenvolvimento de sites orientados a banco de dados (*database-driven web sites*)**
- **sendo possível criar aplicações com base em estruturas pré-definidas.**
- **É um projeto de código aberto escrito na linguagem de programação Ruby.**
- **As aplicações criadas utilizando o framework Rails são desenvolvidas com base no padrão MVC**

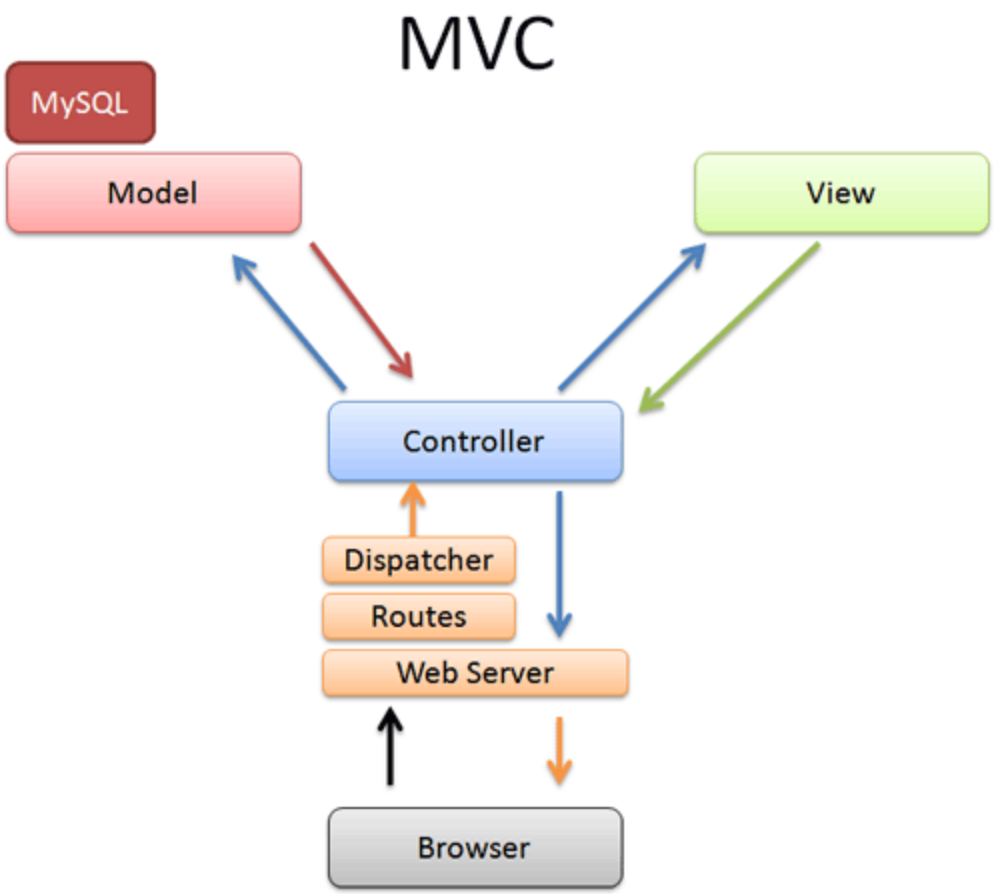
O que é Rails?

- Permite a escrita de código de forma elegante, favorecendo a convenção ao invés da configuração
- O Rails foi criado com o intuito de permitir o desenvolvimento ágil

Arquitetura MVC em Ruby on Rails

- Model
  - Tratamento do dados e lógica do negócio
- View
  - Manipular objetos gráficos da interface do usuário e lógica de apresentação
- Controller
  - Mediador entre a interface do usuário e lógica da aplicação

Arquitetura MVC em Ruby on Rails



Arquitetura MVC em Ruby on Rails

- O **Browser** faz uma requisição: www.inf.ufpr.br/matricula/mostrar/20061127
- O **Web Server** (Ex Apache) recebe a requisição e usa o **Routes** para decidir qual Controller usar. O modelo padrão de rota é controller/action/id. O web server usa o **dispatcher** para instanciar o controlador desejado, chama a action e passar o paramêtro.
- O **Controller** faz o parse dos dados submetidos, e chama a classe do **Model** responsável pela action recebida.
- O **Model** irá executar a lógica de negócio, e envia a resposta ao controller.
- O Controller envia a resposta do **Model** para o **View**, que gera a saída html, esta saída é enviada para Web Server pelo controller.

Arquitetura Ruby on Rails

- Ruby on Rails é separado em varios pacotes:
  - Action Pack
    - Action Controller
    - Action View
  - Active Record
  - Active Model
  - Active Resource
  - Active Mailer

Action Pack

- Action Pack é um a única gem que contém o Action Controller e o Action View. O “VC” do “MVC”.
- Action Controller
  - É o componente que gerencia o Controller em uma aplicação Rails. O Action Controller processa as requisições feitas para uma aplicação Rails, extrai parametros, and envia para a Action desejada.
- Action View
  - Action View gerencia as Views da aplicação Rails, criando saídas HTML e XML.

Active Record

- **O Active Record é uma camada de mapeamento objeto-relacional (*ORM*) equivalente ao Linq, Hibernate, JPA...**
- **É uma camada base para os modelos em uma aplicação Rails.**
- **Garante independencia de banco de dados, operações basicas CRUD.**

Active Model

- **É a camada de lógica de negócio**
- **É chamada pelo Controller, pode utilizar dados da camada de persistência e retorna os resultados ao Controller.**

Active Resource

- O Active Resource fornece um framework para utilização de webservices no modelo REST.
- Usado para comunicação entre aplicações

Action Mailer

- **O Action Mailer é um pacote responsável pelo serviço de entrega e recebimento de e-mails. É relativamente pequeno e simples, porém poderoso.**