Язык программирования Ruby, занятие 5

CS 212

20 ноября 2018 г.

План

1. Sinatra
2. Как отдать верстку?
3. Архитектура Web приложений
4. Ruby on Rails

Sinatra

```
1  #in myapp.rb
2  require 'sinatra'
3
4  get '/' do
5  'Hello world!'
6  end
7
8  #instanll gem
9  gem install sinatra
10
11 ruby myapp.rb
12 #http://localhost:4567
```

Листинг 1: Простейшее web приложение

Routes

```ruby
get '/hello/:name' do |n|
# matches "GET /hello/foo" and "GET /hello/bar"
# params['name'] is 'foo' or 'bar'
# n stores params['name']
"Hello #{n}!"
end
```

Листинг 2: Routes

Routes + Params

```
1 get '/say/*/to/*' do
2 # matches /say/hello/to/world
3 params['splat'] # => ["hello", "world"]
4 end
5
6 get '/download/*.*' do
7 # matches /download/path/to/file.xml
8 params['splat'] # => ["path/to/file", "xml"]
9 end
```

Листинг 3: Routes + params

Params

```
1 get '/posts' do
2 # matches "GET /posts?title=foo&author=bar"
3 title = params['title']
4 author = params['author']
5 # uses title and author variables; query is
      optional to the /posts route
6 end
```

Листинг 4: Params

Ответы Sinatra

```
1
2 [status (Fixnum), headers (Hash), response body
      (responds to #each)]
3 [status (Fixnum), response body (responds to
      #each)]
4 Object.each
5 Fixnum
```

Листинг 5: Ответы

Как отдавать верстку?

1. Отдавать статические файлы
2. Генерировать верстку на сервере
3. Генерировать верстку на клиенте
4. Совмещать все подходы (нет!)

Шаблонизаторы Ruby

1. ERB
2. HAML
3. SLIM

Просто верстка

```
1  <HTML>
2    <HEAD>
3      <TITLE>Your Title Here</TITLE>
4    </HEAD>
5    <BODY BGCOLOR="FFFFFF">
6      <IMG SRC="clouds.jpg" ALIGN="BOTTOM">
7      <H1>TODAY IS 16.11.2018</H1>
8      <P> This is a new paragraph!</P>
9      <HR>
10   </BODY>
11 </HTML>
```

Листинг 6: HTML

Шаблонизатор ERB

```
1  <% require 'date' %>
2  <HTML>
3    <HEAD>
4      <TITLE>Your Title Here</TITLE>
5    </HEAD>
6    <BODY BGCOLOR="FFFFFF">
7      <IMG SRC="clouds.jpg" ALIGN="BOTTOM">
8      <H1><%= DateTime.now.strftime('%d.%m.%Y')
           %></H1>
9      <P> This is a new paragraph!</P>
10     <HR>
11   </BODY>
12 </HTML>
```

Листинг 7: ERB

HAML

```
1 =require 'date'
2 %html
3   %head
4     %title Your Title Here
5   %body{bgcolor: "FFFFFF"}
6     %img{align: "BOTTOM", src: "clouds.jpg"}/
7     %h1 =DateTime.now.strftime('%d.%m.%Y')
8     %p This is a new paragraph!
9     %hr/
```

Листинг 8: HAML

SLIM

```
1 =require 'date'
2 html
3   head
4     title Your Title Here
5   body bgcolor: "FFFFFF"
6     img align: "BOTTOM", src: "clouds.jpg" /
7     h1 =DateTime.now.strftime('%d.%m.%Y')
8     p This is a new paragraph!
9     hr/
```

Листинг 9: SLIM

Известные архитектуры/Паттерны

1. MVC
2. MVVM
3. MVP
4. SPA

MVC

Model - View - Controller
Модель - Представление - Контроллер
Данные - UI - Управление

Model

Отвечает за данные

В большинстве случаев осуществляет отображение из БД в объекты.

Ничего не знает про View

View

Представляет данные пользователю.
Ничего не знает про Model.
Отвечает за взаимодействие с пользователем

Controller

Формирует представления
Обрабатывает действия пользователя, полученные от представления или через запрос
Инициирует изменение моделей.
Стремительно превращается в FSUC (Fat Stupid Ugly Controller)

Active Record

Паттерн и антипаттерн в одном флаконе.
Представлена Мартином Фаулером.
Нарушает принципы SRP

1. Каждый экземпляр данного класса соответствует одной записи таблицы;
2. При создании нового экземпляра класса (и заполнении соответствующих полей) в таблицу добавляется новая запись;
3. При чтении полей объекта считываются соответствующие значения записи таблицы баз данных;
4. При изменении (удалении) какого-либо объекта изменяется (удаляется) соответствующая ему запись.

Rails

Rails - это гем.

```
1 gem install rails
2 rails new blog
3 rails new -h
```

Листинг 10: Инициализация проекта в Rails

Структура проекта

1. app - код приложения
2. bin - обертки для гемов
3. config - конфигурационные файлы
4. db - описание структуры БД
5. lib - внешние/переносимые файлы приложения
6. log - логи
7. public - статические файлы
8. test - тесты
9. tmp - временные файлы
10. vendor - модули и плагины от сторонних разработчиков

Содержимое App

1. assets - config, images, javascript, stylesheets
2. channels - web-сокеты
3. controllers - C
4. helpers - костыли
5. jobs - задачи для сервера
6. mailers -рассылка писем
7. models - M
8. views - V

Окружения

1. development
2. test
3. production
4. stage