

УДК 004.438

РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЙ НА RUBY ON RAILS

Федотов М.А., студент гр. ИТБ-161, IV курс
Алексеева Г.А., старший преподаватель кафедры ИИАПС
Научный руководитель: Чичерин И. В., к.т.н., доцент
Кузбасский государственный технический университет
имени Т.Ф. Горбачева
г. Кемерово

На сегодняшний день существует две возможности реализации приложений. Можно написать приложения для клиентской машины, специально адаптированное к ней, но это влечет за собой определенные проблемы, например – совместимость с различными архитектурами операционных систем и спецификациями оборудования (центральный процессор, графическая карта, ПО и проще). Приложение, созданное в Visual Studio на Windows, необязательно будет работать на Linux, Ubuntu или MacOS. Это ограничивает возможности в ситуации, когда рабочие станции (например, на предприятии) не обладают одной операционной системой, вынуждая «переизобретать колесо» для разных архитектур операционных систем. Но даже в этой ситуации, можно найти решение, которое позволит уменьшить объем работ, требуемый для выполнения поставленной цели. Второй вариант – веб-разработка.

Веб-приложения – это компьютерные программы с архитектурой «клиент-сервер», которую клиент запускает в своем веб-браузере. Наиболее распространенными примерами этих приложений являются почтовый ящик или даже онлайн-банк. Из конкретных примеров – Kickstarter, Github (рисунок 1), Soundcloud, Twitter. Веб-приложения создаются на основе языка HTML, с привязыванием таких языков, как CSS и Java для расширения функционала. В отличие от веб-сайтов, веб-приложение может динамично изменять информацию на странице и позволяет пользователю вносить изменения прямо на странице и увидеть их результат на лету.

The screenshot shows a GitHub repository page for 'Crowbar'. At the top, there's a navigation bar with links for 'Why GitHub?', 'Enterprise', 'Explore', 'Marketplace', 'Pricing', 'Search', 'Sign in', and 'Sign up'. Below the header, the repository name 'ZeqMacaw / Crowbar' is displayed, along with 'Watch' (13), 'Star' (110), 'Fork' (6) buttons. A tabs menu includes 'Code' (selected), 'Issues' (2), 'Pull requests' (0), 'Actions', 'Security', and 'Insights'. The main content area is titled 'Crowbar - GoldSource and Source Engine Modding Tool'. It shows statistics: 256 commits, 11 branches, 0 packages, 7 releases, 2 contributors, and a 'View license' link. A dropdown shows 'Branch: master' and a 'New pull request' button. On the right, there are 'Find file' and 'Clone or download' buttons. The commit history lists several commits from 'ZeqMacaw' and others, with dates ranging from '3 days ago' to '8 months ago'. Below the commit history is a section for 'README.md' containing the text: 'Crowbar - GoldSource and Source Engine Modding Tool' and a 'Website' link. A note at the bottom of this section says: 'IMPORTANT: When providing link to Crowbar, please only use the above website link, because that page provides info and links to everything related to Crowbar.'

Рисунок 1 – Github, пример сайта на Ruby on Rails

Одним из фреймворков для создания веб-приложений является Ruby on Rails. Он является популярным фреймворком MVC (Model-View-Controller), позволяя такие возможности, как безшовное создание таблиц базы данных, миграции, возможность работы с JSON и XML для передачи информации, и HTML, CSS и JavaScript для интерфейса. Ruby on Rails построен на принципах соглашения по конфигурации (Convention over Configuration), Don't repeat yourself и архитектуре Active Record. Версия 1.0 была выпущена 13-го декабря 2005, с последней версией Ruby on Rails, 6.0, вышедшей 16 августа прошлого года.

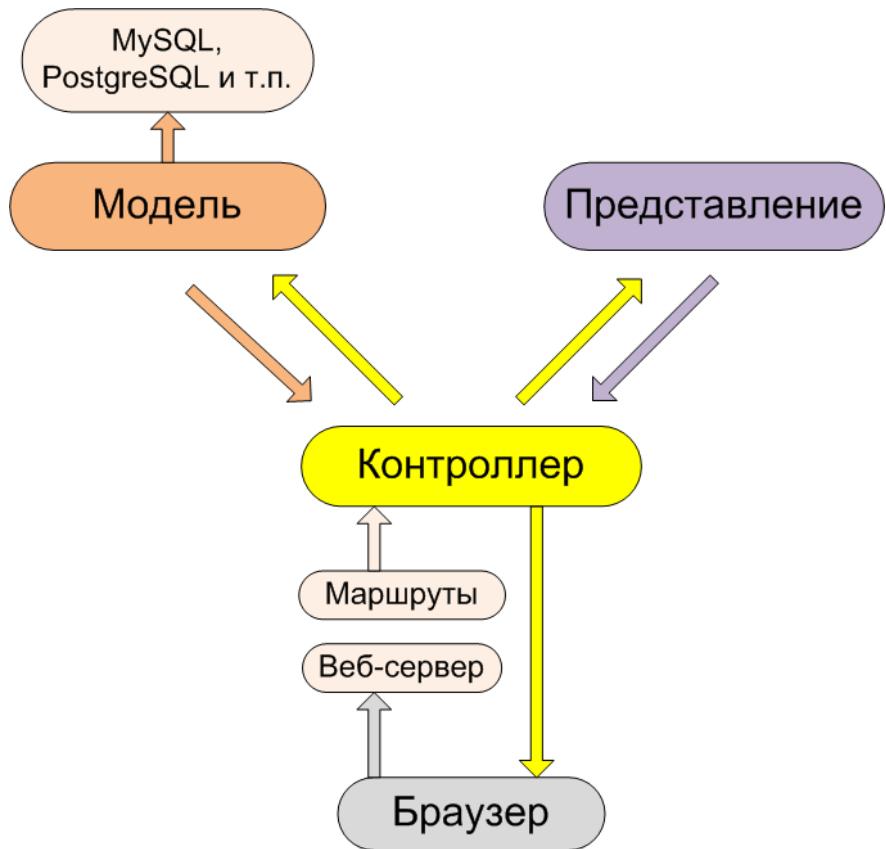


Рисунок 2 – Архитектура фреймворка и взаимодействие с компонентами

Структура проекта Ruby on Rails приведена на рисунке 3.

📁 app	7/1/2015 5:06 PM	File folder
📁 bin	7/1/2015 5:06 PM	File folder
📁 config	7/1/2015 5:06 PM	File folder
📁 db	7/1/2015 5:06 PM	File folder
📁 lib	7/1/2015 5:06 PM	File folder
📁 log	7/1/2015 5:06 PM	File folder
📁 public	7/1/2015 5:06 PM	File folder
📁 test	7/1/2015 5:06 PM	File folder
📁 tmp	7/1/2015 5:06 PM	File folder
📁 vendor	7/1/2015 5:06 PM	File folder
📄 .gitignore	7/1/2015 5:06 PM	GITIGNORE File 1 KB
config.ru	7/1/2015 5:06 PM	RU File 1 KB
Gemfile	7/1/2015 5:06 PM	File 2 KB
Gemfile.lock	7/1/2015 5:07 PM	LOCK File 4 KB
Rakefile	7/1/2015 5:06 PM	File 1 KB
README.rdoc	7/1/2015 5:06 PM	RDOC File 1 KB

Рисунок 3 – Структура проекта Ruby on Rails

Основными элементами структуры приложений Ruby on Rails являются:

1. app – организовывает компоненты приложения. Содержит подпапки, которые содержат в себе контроллеры, модели и представления;
2. bin – скрипты для старта, настройки, обновления, запуска и развертывания;
3. config – содержит конфигурационный код, необходимый для приложения, связанный с конфигурацией БД, структуры приложения и перенаправления входящих веб-запросов;
4. db – в ней присутствуют файл базы данных и миграции, создаваемые в процессе работы над созданием моделей;
5. lib – сюда можно вложить сторонние библиотеки для включения их в проект;
6. log – здесь можно найти логи ошибок, которые генерируются Ruby on Rails;
7. public – содержит файлы, которые не изменяются, выполняя функцию наподобие тех, что есть на веб-серверах;
8. test – сюда попадают тесты, которые пишутся разработчиком в консоли и создаваемые Ruby on Rails;
9. tmp – временные файлы;
10. vendor – библиотеки, предоставленные третьими источниками;
11. Gemfile – файл с так называемыми «джемами», компонентами, которые можно установить из консоли, может быть дописан разработчиком и обработан по запросу;
12. Gemfile.lock – содержит все джемы, которые были установлены в текущем проекте;
13. Rakefile – подобен Makefile Unix'а, помогает с строительством, упаковыванием и тестированием кода Ruby;
14. README – файл «прочти меня», содержит основные заметки о приложении.

Рассмотрим создание приложения на примере небольшой страницы. Для ее создания необходимо выполнить следующие действия:

1. Создать проект Ruby on Rails

```
C:\RubyProjects>rails new Diploma_project_RoR
```

2. Перейти в папку проекта и сгенерировать контроллер StaticPages с ветками «home» и «about»

```
C:\RubyProjects\Diploma_project_RoR>rails generate controller StaticPages home about_-
```

3. Отредактировать файлы routes.rb (для создания «маршрутов» к страницам приложения), home.html.erb (главная страница), а также создать _header.html.erb (навигационная полоса сверху страницы). Листинги файлов приведены на рисунках 4-6.

```
Rails.application.routes.draw do
    root 'static_pages#home'
    get '/about', to: 'static_pages#about'
    get '/contact', to: 'static_pages#contact'

    # For details on the DSL available within :end
```

Рисунок 4 – Файл routes.rb

```
<div class="center jumbotron">
    <h1>Gotta go fast</h1>

    <h2>
        Так много осталось сделать... И так мало времени...
    </h2>

</div>
```

Рисунок 5 – Файл home.html.erb

```
<header class="navbar navbar-fixed-top navbar-inverse">
    <div class="container">
        <%= link_to "Ruby on Rails", root_path, id: "logo" %>
        <nav>
            <ul class="nav navbar-nav navbar-right">
                <li><%= link_to "Домой", root_path %></li>
                <li><%= link_to "О сайте", about_path %></li>
                <% if false %>
                <% if logged_in? %>
                    <li><%= link_to "Пользователи", users_path %></li>
                    <li class="dropdown">
                        <a href="#" class="dropdown-toggle" data-toggle="dropdown">
                            Аккаунт <b class="caret"></b>
                        </a>
                        <ul class="dropdown-menu">
                            <li><%= link_to "Профиль", current_user %></li>
                            <li><%= link_to "Настройки", edit_user_path(current_user) %></li>
                            <li class="divider"></li>
                            <li>
                                <%= link_to "Выйти", logout_path, method: :delete %>
                            </li>
                        </ul>
                    </li>
                <% else %>
                    <li><%= link_to "Войти", login_path %></li>
                <% end %>
                <% end %>
            </ul>
        </nav>
    </div>
</header>
```

Рисунок 4 – Файл _header.html.erb

4. Запустить локальный сервер приложения Puma
- ```
C:\RubyProjects\Diploma_project_RoR>rails server
```

5. Запустить веб-браузер и зайти по адресу localhost:3000  
Результат работы представлен на рисунке 7.

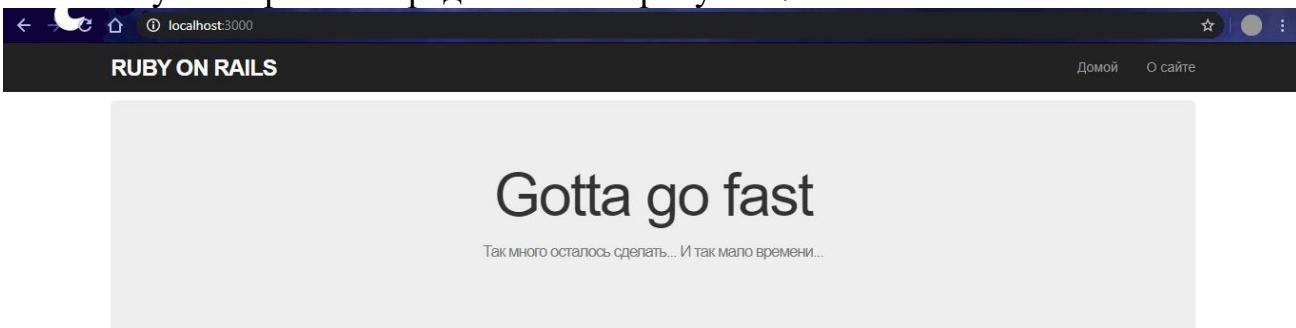


Рисунок 7 – Стартовая страница нашего приложения

В заключение, отметим преимущества разработки на фреймворке Ruby on Rails:

1. Архитектура MVC, которая централизует бизнес-логику приложения и правила манипуляций с данными, улучшает обслуживаемость и создания направленных тестов
2. Парадигма соглашения по конфигурации (Convention over Configuration), которая позволяет тратить меньшее количество времени и сил на учет параметров всех файлов конфигурации, так как фреймворк «предполагает» ваши действия, распространяя конфигурацию на весь проект
3. Принцип Don't repeat yourself, который упрощает чтение кода, минимизирует затраты времени на разработку на 25-50% по сравнению с другими популярными веб-фреймворками и подходит для всех бизнес индустрий

#### Список литературы:

1. Web application – Wikipedia [Электронный ресурс]. – URL: – [https://en.wikipedia.org/wiki/Web\\_application](https://en.wikipedia.org/wiki/Web_application) (дата обращения: 05.03.2020)
2. Ruby on Rails – Wikipedia [Электронный ресурс]. – URL: – [https://en.wikipedia.org/wiki/Ruby\\_on\\_Rails](https://en.wikipedia.org/wiki/Ruby_on_Rails) (дата обращения: 05.03.2020)
3. Difference between Website and Web Application [Электронный ресурс]. – URL: – <https://www.guru99.com/difference-web-application-website.html> (дата обращения: 05.03.2020)
4. Advantages of Web Development with Ruby on Rails [Электронный ресурс]. – URL: – <https://onix-systems.com/blog/advantages-of-web-development-with-Ruby-on-Rails> (дата обращения: 05.03.2020)