

УДК 004

СРАВНЕНИЕ ВЕБ-ФРЕЙМВОРКОВ DJANGO И RUBY ON RAILS

В.А. Жилин, студент гр. ПИ-121, III курс
Научный руководитель: И.А. Соколов, к.т.н., доцент
Кузбасский государственный технический университет
имени Т.Ф. Горбачева, г. Кемерово

В наши дни молодому программисту довольно трудно определиться с направлением и технологией, с которой стоит связать своё будущее. Эта же проблема встала передо мной в прошлом. Как и большинство начинающих программистов я не знал с чего начать, но я понимал, что свяжу свою жизнь с языками программирования высокого уровня. В связи со спецификой своего обучения, мой выбор пал на разработку back-end'а различных веб-платформ.

После поисков на сайте <https://www.google.ru/> мой выбор пал на 2 самых распространенных фреймворка: Ruby on Rails и Django, написанные на Ruby и Python соответственно. С технологической точки зрения среди них нет явного лидера, они оба используют в своей основе языки программирования высокого уровня и обладают поистине безграничным функционалом.

Если перед вами встает вопрос выбора технологии для разработки, то ваши личные предпочтения в плане логики используемого инструмента так же важны, как и ваши конечные цели. Давайте посмотрим на них поближе:

Общее:

- Фреймворки работают на сервере, не на клиентской части, что очень позитивно сказывается на производительности.
- Одной из основных концепций является DRY (Don't Repeat Yourself – стремление исключить все повторяющиеся участки кода).
- Богатая база документации, плагинов и расширений, активно поддерживаемых пользователями и разработчиками (что крайне позитивно сказывается на скорости развёртывания веб-проектов).

Рассмотрим более подробно Ruby on Rails. У этого фреймворка открытый исходный код, а сам он был разработан в 2005 году.

Особенности:

- Изначально в новых проектах используется подход, получивший название «Convention over Configuration» На практике это означает, что нет необходимости в специфичной настройке какой-либо составляющей веб-проекта, если это не удовлетворяет некоторой спецификации (как следствие – быстрое и лёгкое развёртывание новых приложений).

- Довольно удобная система назначений функционала ключевых слов специальным символам (позволяет существенно сократить объёмы программного кода).
- Основные цели, реализуемые этим проектом – скорость разработки и адаптации к этому инструменту.
- Большая база знаний по фреймворку как на специальных сайтах, так и во встроенной справке.
- Часто используется стартап разработчиками.

Примерами сайтов на нём являются [Twitter](#) и [Github](#).

Теперь взглянем на Django. Этот фреймворк написан на Python и так же был выпущен в 2005 году.

Особенности:

- Разработчик имеет большой простор в настройке конфигурации и внешнего вида.
- Логика взаимодействия составляющих частей довольно прозрачна и минималистична.
- По сравнению с RoR переходы на новые версии не вызывают затруднений и выходят не слишком часто.
- Так как используется Python, синтаксис довольно просто и напоминает английский язык.
- Помимо этого, Python проще управляется с манипулированием данными, аналитикой, системным администрированием и программированием сложных информационных систем.
- Часто используется в научных целях.

Примерами сайтов на нём являются [The Washington Times](#) и [Instagram](#).

В целом главное различие заключается в том, что Ruby on Rails требует меньше времени для изначального развёртывания, в то время как Django позволяет более детально настраивать свой продукт. Поэтому начав изучение Django, мне впоследствии пришлось перейти на Ruby on Rails, так как он лучше подошёл для конвейерного создания небольших сайтов.

Сама его философия (настройки по умолчанию) предполагает меньшее количество трудозатрат. Помимо этого, поддержка сообщества этого фреймворка мне показалась более широкой (у разработчиков фреймворка даже есть специальная горячая линия для консультирования).

Список литературы:

1. <http://rubyonrails.org/> - официальный сайт Ruby on Rails
2. <https://www.djangoproject.com/> - официальный сайт Django