УДК 004.4 DOI: 10.47813/dnit.2021.2.289-293

# Возможности веб фреймворка FastAPI для реализация серверной части веб систем

**Р.Е. Богачёв[1,2,*], Н.В. Зариковская[1,2]**

[1]ООО «АльдераСофт», ул. Мокрушина, 9, Томск, 634045, Россия
[2]Томский государственный университет систем управления и радиоэлектроники, пр. Ленина, 40, Томск, 634050, Россия

*E-mail: bogachyov.mail@gmail.com

**Аннотация.** В работе приведен обзор возможностей веб фреймворка FastAPI для backend - разработки веб систем. Произведено его сравнение с популярными аналогами такими как Django и Flask, описаны основные достоинства и преимущества при разработке на примере разработки веб системы для терминалов.

**Ключевые слова:** веб фреймворк, веб система, серверная разработка, FastAPI, Django, Flask

# Overview of the capabilities of the FastAPI web framework for implementing the backend of web systems

**R.E. Bogachev[1,2,*], N.V. Zarikovskaya[1,2]**

[1]«AlderaSoft» LLC, 9, Mokrushina st., Tomsk, 634045, Russia
[2]Tomsk State University of Control Systems and Radioelectronics, 40, Lenina pr., Tomsk, 634050, Russia

*E-mail: bogachyov.mail@gmail.com

**Abstract.** This article provides an overview of the FastAPI web framework capabilities for web systems backend development. The comparison with popular frameworks such as Django and Flask is described and the main advantages of using FastAPI framework are presented with the terminals web system development as an example.

**Keywords:** web framework, web systems, backend development, FastAPI, Django, Flask

## 1. Введение

В настоящее время разработка информационных систем разделяется на две большие составляющие: frontend-разработка и backend-разработка. Для backend-разработки наиболее популярными являются следующие языки программирования: C++, C#, Java, Python, PHP. Однако на «чистых» языках сейчас никто не пишет, в основном, используются вспомогательные средства, такие как библиотеки и веб-фреймворки. Данная работа посвящена обзору наиболее перспективного веб фреймворка для языка программирования Python – FastAPI.

## 2. Python и его фреймворки

Python [1] является очень популярным языком программирования, который мало того, что является мощным инструментом разработки, но и одним из самых любимых языков разработчиков. К основным достоинствам языка можно отнести простой синтаксис, огромное количество сторонних библиотек [2] и хорошая совместимость на разных платформах. Python имеет огромное количество веб фреймворков, среди которых можно выделить, следующие наиболее популярные: Django и Flask.

Django фреймворк – это один из популярнейших на данный момент фреймворков, имеющий поддержку многих «тех гигантов», таких как Googe и NASA, огромный набор инструментов «из коробки», такие как встроенная ORM, библиотека «шаблонов» HTML и строго заданная структура проектов. Из недостатков можно выделить следующее: отсутствие поддержки асинхронного режима работа «из коробки» (доступно из сторонних библиотек), огромное количество встроенного функционала, что может повлечь дополнительные трудозатраты для проектов малого и среднего масштаба, медленный процесс разработки фреймворка (новые технологии приходят в него с «опозданием»).

Flask фреймворк – это так называемый микро-веб фреймворк. По сравнению с Django имеет минимальный набор инструментов, что делает его проще в освоении и понимании. Он позволяет значительно облегчить разработку для проектов небольшого масштаба, имеет инструменты легкой интеграции со сторонними технологиями и огромное количество расширений. К недостаткам относятся: отсутствие поддержка асинхронного режима работы («из коробки»), сложная поддержка и сопровождение для проектов большого масштаба.

Помимо этого, ни Django, ни Flask не поддерживают из «коробки» в полной мере архитектурный стиль REST.

До недавнего времени если речь шла о создании веб-системы на Python выбор стоял в основном между Django и Flask, однако недавно появился достойный конкурент для них – FastAPI.

FastAPI – это новый веб фреймворк, активно набирающий популярность. На данный момент, по результатам различных тестов, он является одним из самых быстрых веб-фреймворков для Python. Системы, написанные с использованием данного фреймворка, значительно превосходят по скорости обработки запросов аналоги, написанные на Django или Flask [3]. К остальным достоинствам можно отнести следующие: поддержка асинхронного режима работа, легкость в понимании и использовании, встроенная и автоматическая генерация документации API Swagger и Redoc (ни Django, ни Flask не имеет альтернатив, которые бы смогли так же быстро и качественно её генерировать), отличная документация самого фреймворка.

## 3. Возможности фреймворка FastAPI

Для обзора практического использования FastAPI, рассмотрим его применение в проекте по реализации веб системы для терминалов, а конкретнее автоматическую генерацию документации API Swagger, а также реализацию системы разделения пользовательских прав.

### 3.1. Документация API Swagger

Как было сказано ранее, FastAPI позволяет автоматически генерировать документация Swagger и Redoc. Фреймворк изначально разрабатывался с целью их поддержки, поэтому качество документации и простота её создания находятся на высоком уровне. Серверной части веб системы терминала, необходима интеграция со многими сторонними приложениями на других платформах, например, desktop, web и mobile. Наличие API документации значительно упрощает процесс интеграции, поскольку предоставляет весь необходимый набор данных, который может понадобиться разработчику, работающему с данным API. Djanog и Flask генерируют документацию Swagger с помощью сторонних библиотек или расширений, которые разрабатываются сторонними разработчиками, из-за этого возможно появление ошибок в документации как показано на рисунке 1, где вместо одной API точки доступа, сгенерировалось четыре. В FastAPI такие ошибки возможно только по вине самого разработчика.

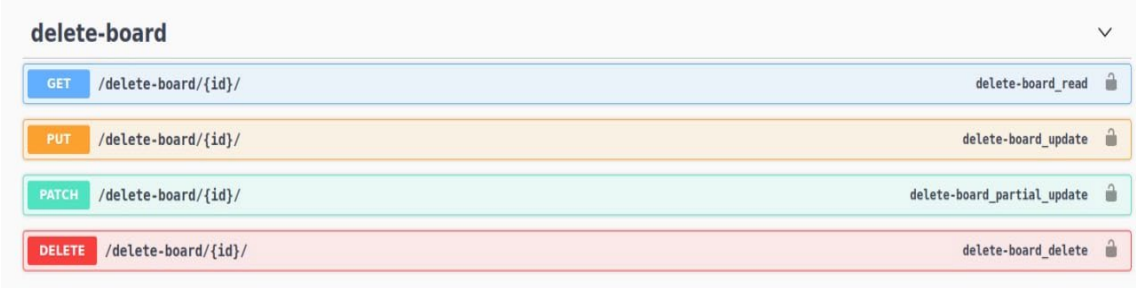


**Рисунок 1.** Ошибка генерации Swagger для Django.

### 3.2. Система разделение пользовательских прав

Данная система предполагает, что на каждую API точку доступа есть некое ограничение. В данном случае – булево выражение, которое определяет условия, при которых пользователь будет иметь доступ к данной точке доступа. На рисунке 2 графически представлен пример такого подхода.

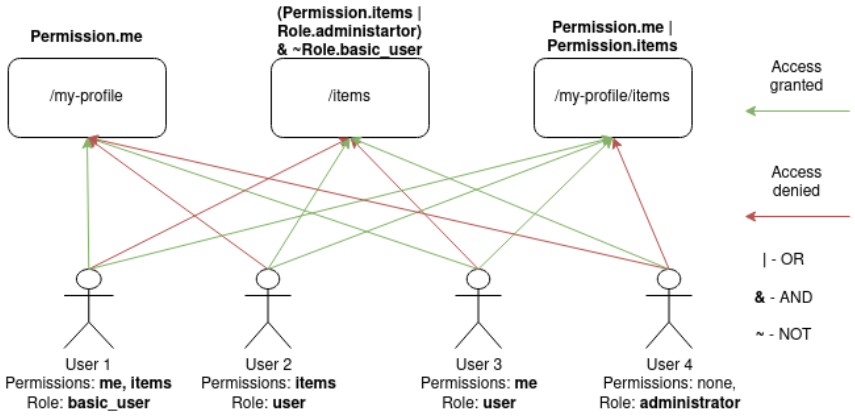


**Рисунок 2.** Схема системы пользовательских прав.

Такой механизм позволяет очень гибко настраивать, а точнее, ограничивать доступ к определенному функционалу системы. Веб система терминала предполагает наличие множества ролей (администратор, оператор, техник и так далее), а также большое количество прав (доступ к сессиям, доступ к панели пользователей и др.). Имея такую системы разделения ролей можно легко, в плане написания кода, и в очень понятной форме для человека разделить функционал всей системы для пользователя с конкретной ролью или правами. Для интеграции данной системы непосредственно с самим FastAPI, был использован удобный и гибкий инструмент, входящий в состав самого фреймворка и называемый Dependency Injection system [4]. Его особенность – простота в использовании, а также легкость в интеграции сторонних компонентов с FastAPI.

## 4. Выводы

FastAPI доказал, что является отличным фреймворком, содержащим в себе наличие многих современных технологий и тенденций в разработке. Дополненный собственными уникальными механизмами, а также показывающий великолепную производительность написанных на нём систем, он является очень мощным и перспективным инструментом разработки, который, вполне возможно вскоре станет более популярным, чем Django и Flask.

## Список литературы

1. Stackoverflow Developer Survey [Электронный ресурс]. URL: https://insights.stackoverflow.com/survey/2021#most-loved-dreaded-and-wanted-language-love-dread (дата обращения: 29.11.2021)
2. Pypi [Электронный ресурс]. URL: https://pypi.org/ (дата обращения: 29.11.2021)
3. FastAPI benchmarks [Электронный ресурс]. URL: https://fastapi.tiangolo.com/benchmarks/ (дата обращения: 29.11.2021)
4. FastAPI dependencies [Электронный ресурс]. URL: https://fastapi.tiangolo.com/tutorial/dependencies/ (дата обращения: 29.11.2021)