

УДК 004.423

DOI 10.51691/2541-8327\_2023\_6\_15

## ***ОСНОВЫ ФРЕЙМВОРКА FLASK ДЛЯ НАПИСАНИЯ WEB-СТРАНИЦ В PYTHON***

***Халевин Т.А.***

*студент направления подготовки информатики и вычислительной техники,  
Хакасский государственный университет имени Н.Ф. Катанова,  
г. Абакан, Россия <sup>1</sup>*

**Аннотация:** Данная статья охватывает тему создания веб-приложений с использованием Flask – популярного фреймворка на языке Python. В статье описывается процесс установки Flask и создания первого приложения, а также рассматривается механизм роутинга для связывания URL-адресов с функциями-обработчиками запросов. Статья предназначена для разработчиков, которые хотят овладеть основами работы с Flask и создать веб-приложения на языке Python.

**Ключевые слова:** Python, Flask, фреймворк, веб-приложение, роутинг, функции-обработчики

## ***FLASK FRAMEWORK BASICS FOR WRITING WEB PAGES IN PYTHON***

***Khalevin T.A.***

*student of computer science and computer engineering department,  
N.F. Katanov Khakass State University,  
Abakan, Russia*

---

<sup>1</sup> Научный руководитель: Голубничий А.А. старший преподаватель кафедры ПОВТиАС, Хакасский государственный университет имени Н.Ф. Катанова, г. Абакан, Россия  
Дневник науки | [www.dnevnika.ru](http://www.dnevnika.ru) | СМИ ЭЛ № ФС 77-68405 ISSN 2541-8327

**Abstract:** This article covers the topic of creating web applications using Flask, a popular Python framework. The article describes the process of installing Flask and creating the first application, and also discusses the routing mechanism for linking URLs to request handler functions. The article is intended for developers who want to master the basics of Flask and create web applications in Python.

**Keywords:** Python, Flask, framework, web applications, routing, handler functions

В данной статье мы рассмотрим фреймворк Flask и его функциональность. Flask является одним из самых популярных фреймворков для создания веб-приложений на языке Python. Он обладает множеством инструментов для создания высококачественных веб-приложений и может быть использован как для небольших проектов, так и для крупномасштабных систем [1].

Flask – это легковесный фреймворк для разработки веб-приложений на языке программирования Python. Он обеспечивает минимальный набор инструментов и библиотек, необходимых для создания веб-приложений, но при этом обладает достаточной гибкостью для расширения и адаптации под конкретные потребности разработчика [2].

Flask не имеет определенного способа работы с базами данных, аутентификацией и другими аспектами, что позволяет разработчикам использовать любые сторонние библиотеки и решения для создания приложений. Такой подход делает Flask универсальным инструментом для разработки веб-приложений любого уровня сложности.

Функциональность Flask включает в себя следующие основные компоненты:

1. Роутинг и обработка запросов: Flask предоставляет удобный способ определения URL-адресов и связывания их с функциями, обрабатывающими запросы.

2. Шаблонизация: Flask использует библиотеку Jinja2 для генерации HTML-кода, что упрощает создание динамических веб-страниц.

3. Работа с формами: Flask позволяет легко обрабатывать отправленные пользователем данные через формы на веб-страницах.

4. Взаимодействие с базами данных: Flask не имеет своего встроенного инструмента для работы с базами данных, но с помощью сторонних библиотек, таких как SQLAlchemy, разработчики могут легко интегрировать базы данных в свои приложения.

5. Аутентификация: Flask позволяет создавать системы аутентификации для веб-приложений. Flask-Login - это одно из расширений Flask, предназначенных для работы с системами аутентификации.

6. Расширяемость: Flask позволяет использовать сторонние библиотеки и пакеты для создания приложений любой сложности.

Flask является одним из самых популярных фреймворков для разработки веб-приложений на Python, благодаря своей гибкости, простоте и легковесности [3].

Установка Flask довольно проста. Для начала, необходимо убедиться, что у вас установлен Python на вашем компьютере. Flask поддерживает Python версии 3.6 и выше.

В первую очередь, для установки фреймворка Flask, необходимо выполнить следующую команду (Рис. 1).

```
$ pip install flask
```

Рисунок 1 – Установка фреймворка [разработано автором]

Теперь, когда установлен Flask, можно приступить к созданию первого Flask-приложения.

Для создания простого Flask-приложения необходимо:

1. Создание экземпляра класса Flask;
2. Функция-обработчик HTTP запроса;

### 3. Запуск проекта.

Рассмотрим каждый пункт по отдельности.

Создадим простое приложение, которое будет выводить "Это домашняя страница" при обращении к корневому URL в веб-браузере [3].

Первым шагом необходимо создать экземпляр класса Flask для текущего файла.

Для этого нужно импортировать модуль Flask и создать экземпляр класса Flask (Рис. 2).

```
from flask import Flask
app = Flask(__name__)
```

Рисунок 2 – Импорт модуля и создание экземпляра [разработано автором]

Здесь `__name__` – это имя текущего модуля, которое Flask использует для поиска других файлов в вашем приложении.

Вторым шагом определяем функцию-обработчик, которая будет вызываться при обращении к корневому URL [4]. Эта функция будет возвращать строку "Это домашняя страница" (Рис. 3).

```
@app.route('/')
def home():
    return 'Это домашняя страница.'
```

Рисунок 3 – Функция обработчик [разработано автором]

Декоратор `@app.route('/')` указывает, что функция `home()` должна вызываться при обращении к корневому URL-адресу.

Последним пунктом запускаем Flask-приложение (Рис. 4).

```
if __name__ == '__main__':
    app.run()
```

Рисунок 4 – Запуск Flask приложения [разработано автором]

Здесь `if __name__ == '__main__':` проверяет, является ли текущий файл главным, и только тогда запускает сервер Flask с помощью метода `app.run()`.

Необходимо сохранить файл и запустить его.

После запуска сервера, по адресу `http://127.0.0.1:5000` на экране будет отображаться текст "Это домашняя страница." (Рис. 5).

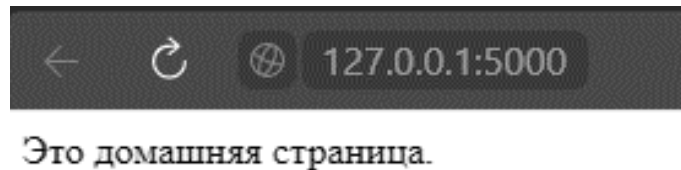


Рисунок 5 – Вывод строки в веб-браузере [разработано автором]

Первое Flask приложение создано. При необходимости можно изменять этот код и добавлять новые функции, чтобы создавать более сложные веб-приложения. Теперь рассмотрим, как использовать роутинг в Flask для настройки обработки запросов на разных URL-адресах. В статье уже использовался декоратор с роутингом, но остановимся немного подробнее на данной технологии.

Роутинг (routing) в Flask – это механизм, который позволяет настраивать обработчики запросов для разных URL-адресов. Flask использует декораторы Python для определения роутинга. Для настройки роутинга в Flask используется декоратор `@app.route`. Декоратор принимает URL-адрес в качестве аргумента и связывает его с функцией-обработчиком, которая будет вызываться при обращении к этому URL-адресу. В качестве примера добавим функцию-обработчик `about()` которая будет выводить текст "Это страница о чём-то.". Для этого перед функцией используем декоратор `app.route('/about')` (Рис.6).

```
from flask import Flask

app = Flask(__name__)

@app.route('/')
def home():
    return 'Это домашняя страница.'

@app.route('/about')
def about():
    return 'Эта страница о чём-то'

if __name__ == '__main__':
    app.run()
```

Рисунок 6 – Полный код приложения [разработано автором]

Необходимо сохранить файл и запустить его. После перехода по адресу <http://127.0.0.1:5000> в веб-браузере будет отображаться сообщение "Это домашняя страница.". При переходе по адресу <http://127.0.0.1:5000/about>, будет отображаться сообщение "Эта страница о чём-то." (Рис. 7).



Рисунок 7 – Страница about [разработано автором]

## Заключение

В целом, Flask представляет собой легковесный и гибкий фреймворк для создания веб-приложений на языке Python, который обладает широкими возможностями для разработки и отладки приложений.

## Библиографический список:

1. Доусон М. Програмируем на Python [Текст] / Доусон М. – СПб.: Издательский Дом ПИТЕР, 2022. – 416 с.
2. Гэддис Т. Начинаем программировать на Python [Текст] / Т. Гэддис. – СПб.: БХВ-Петербург, 2021. – 768 с.
3. Васильев А.Н. Программирование на Python в примерах и задачах [Текст] / Васильев А.Н. – М.: Бомбора, 2021, 2022.

4. Гуриков, С.Р. Основы алгоритмизации и программирования на Python [Текст] / С.Р. Гуриков. – М.: ИНФРА-М, 2023. – 343 с.

*Оригинальность 77%*