



8 Популярных фреймворков Python для создания API

Описание

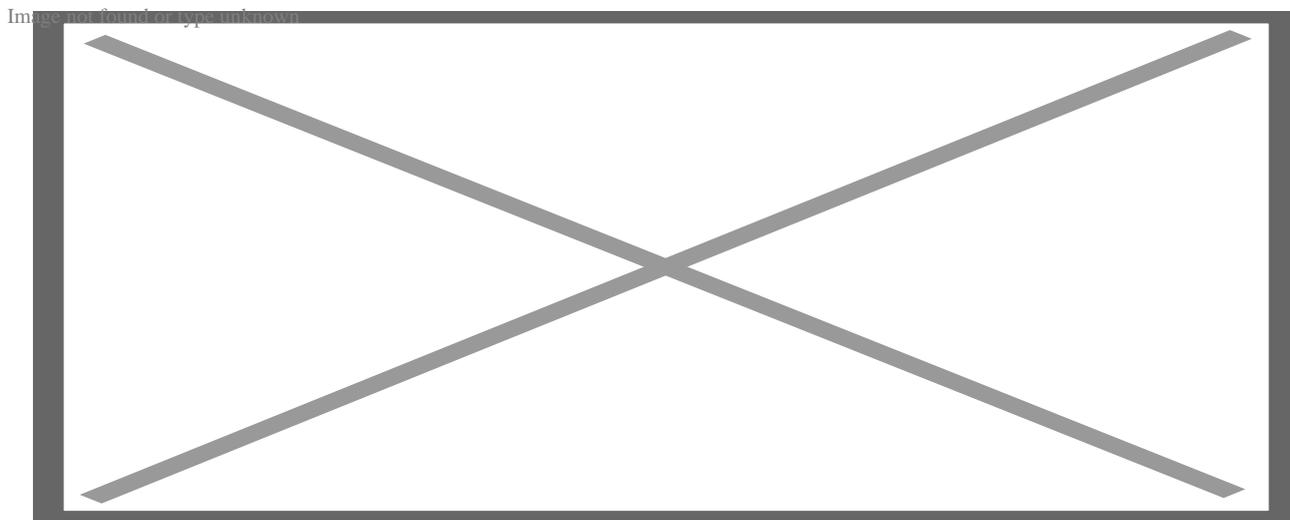
Создавайте мощные API с помощью Python. Давайте узнаем как! Python – один из самых универсальных языков программирования на сегодняшний день. Он завоевал свою популярность благодаря своей читабельности, менее сложному синтаксису и простоте изучения. За последние несколько десятилетий Python использовался во многих областях для решения различных задач. Он используется для автоматизации, веб-разработки, аналитики данных, разработки приложений, а в данной статье речь пойдет об интерфейсах прикладного программирования (API).

Пользовательские интерфейсы приложений используются для доступа к некоторой информации или функциональности через Интернет. Например, вы можете написать программу на Python для использования API Twitter для выполнения функции, которая может собирать твиты по определенному хэштегу. API облегчает разработчикам использование определенных технологий для создания приложений с использованием определенных предопределенных операций. Он помогает соединять различные точки. К счастью, существует множество фреймворков Python, но не все они предназначены для API. Давайте рассмотрим некоторые из популярных, которые вы можете использовать для разработки API.

Django REST

Django REST – это REST-фреймворк на основе Python, который часто используется разработчиками для создания веб-интерфейсов. Rest расшифровывается как representational state transfer, что используется в веб-архитектуре для передачи

данных. Он использует HTTP для осуществления вызовов между машинами. GET, PUT, POST и DELETE – это несколько стандартных методов, используемых в Django rest framework.



Django rest framework – это просто способ создания REST API с помощью Django, так что это просто набор инструментов, с которыми мы можем работать. Он дает нам много дополнительных функций. Используя pip, вы можете установить Django REST framework одной командой на вашу систему. Но перед установкой Django REST необходимо иметь Python версии 3.5+ и Django в вашей системе.

Особенности Django REST:

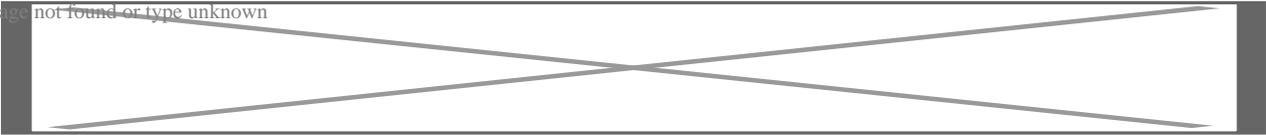
- API с возможностью веб-просмотра и огромным удобством для разработчиков
- Множество встроенных политик аутентификации
- Сериализация, которая поддерживает как ORM, так и не ORM источники данных
- Обширная и хорошая документация для ознакомления и изучения
- Очень активная поддержка сообщества
- Доверяют такие организации, как Red Hat, Mozilla, Heroku
- Поддерживает автоматическую маршрутизацию URL в Django
- Поддерживает тестирование, кэширование, дросселирование и т.д.

Flask Restful

Flask Restful – это расширение фреймворка Flask, используемое разработчиками для быстрого создания rest API. Он известен для создания микро веб-сервисов, поскольку имеет очень легкий модуль. Он быстр и требует всего несколько строк

кода, чтобы инициализировать Flask и начать использовать его для создания API.

Image not found or type unknown



Он прост в настройке и поощряет лучшие практики создания API. За последние несколько лет во Flask было добавлено множество функций, и теперь он не уступает полнофункциональному фреймворку. Это предпочтительный выбор веб-разработчиков для создания веб-приложений и их API.

Особенности Flask Restful:

- Очень легкий и простой в установке
- Предоставляет ресурсы, которые являются строительным блоком flask restful. Он используется для доступа к нескольким методам HTTP.
- Поставляется с модулем fields и декоратором marshal_with() для форматирования данных.
- Позволяет контролировать, какие поля будут показаны в ответе после рендеринга.
- Позволяет добавлять пользовательские поля и входы на основе вашего типа данных
- Предоставляет декоратор representation() для поддержки других представлений, таких как XML, CSV, HTML.

Falcon

Falcon – это веб-фреймворк, который используется для построения надежных и высокопроизводительных бэкендов крупномасштабных приложений и микросервисов. Falcon определяет свои ресурсы API с помощью объектно-ориентированного и основанного на классах интерфейса. Он совместим с различными серверами и платформами, поскольку совместим с WSGI.



Falcon

Falcon принимает стиль архитектуры HTTP и REST, который используется для создания чистых дизайнов. Девиз Falcon – делать как можно меньше, но при этом быть высокоэффективным. Этот REST-фреймворк ориентирован на контроль качества.

Особенности Falcon:

- Он поставляется со встроенным сервером и отладчиком для разработки
- Легкий вес с минимальной настройкой
- В основном используется для повышения производительности
- Сопоставление URI с ресурсами происходит по принципу REST
- Поддерживает маршрутизацию, обработку медиафайлов по протоколу HTTP, cookies и утилиты URI
- Легко интегрируется с NoSQL
- Поддерживает встроенные модульные тесты
- Обеспечивает поддержку безопасных файлов cookie
- Поддерживает Python 2.7, 3.5+ и совместим с Pyru.

Connexion

Connexion обрабатывает HTTP-запросы автоматически, и он основан на спецификации OpenAPI. Спецификация OpenAPI описана в формате YAML. В отличие от других инструментов, где вы получаете сгенерированную спецификацию на основе кода Python, Connexion позволяет вам написать спецификацию OpenAPI и отобразить соответствующие конечные точки функций Python. Как только вы подробно опишете свой REST API, Connexion убедится, что он работает.

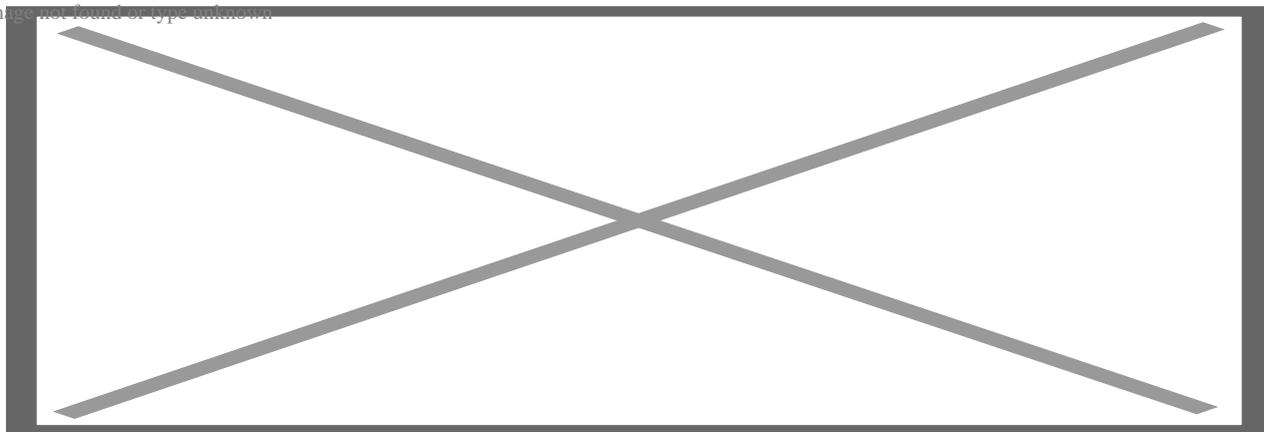
Возможности Connexion:

- На основе предоставленной вами спецификации он проверяет запросы и конечные точки.
- Обрабатывает аутентификацию на основе токенов с помощью OAuth 2.
- Поддерживает версионирование API
- Поставляется с Web Swagger Console UI, который также позволяет вызывать конечные точки API.
- Автоматически сериализует полезную нагрузку

FastAPI

Как следует из названия, FastAPI является одним из самых быстрых и высокопроизводительных Python-фреймворков для создания API. Он представляет собой полный пакет полноценного веб-фреймворка. Используя FastAPI, вы можете разрабатывать код в 200-300% раз быстрее. Популярность FastAPI значительно растет в сообществе Python: 200+ контрибьюторов и 28,6К запусков на данный момент.

Image not found of type unknown



FastAPI основан на спецификации ASGI и в основном используется для создания асинхронных веб-приложений. Ниже перечислены некоторые из его особенностей.

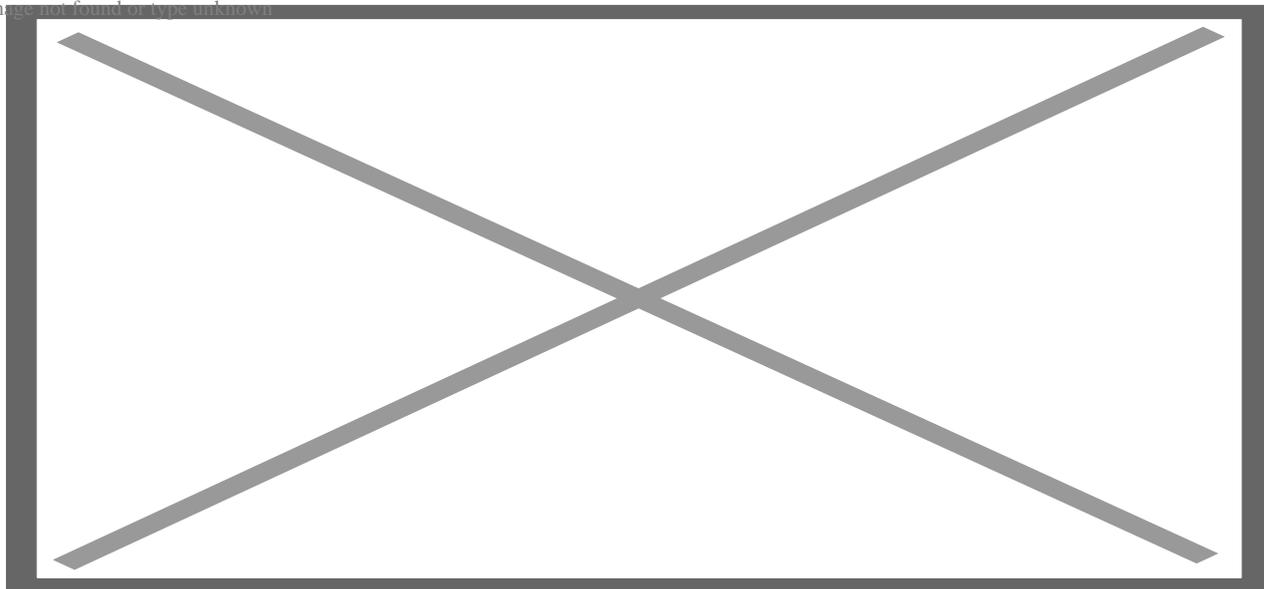
- Основан на открытых стандартах – OpenAPI, схема JSON, автоматическая генерация кода клиента
- Пользовательский интерфейс Swagger для вызова и тестирования API из браузера
- Интерактивная документация по API

- Снижает 40% ошибок, возникающих из-за ошибок разработчика в коде
- Минимизирует дублирование кода
- Поддержка интуитивного редактора (VSCode/PyCharm) с автозавершением.
- Поставляется с интегрированной системой безопасности и аутентификации
- Система инъекции зависимостей для автоматической обработки всех зависимостей
- Полная совместимость с Starlette и Pydantic
- Неограниченная поддержка плагинов
- Поддерживает тестирование приложений с помощью PyTest
- Поддерживает базы данных SQL, базы данных NoSQL, GraphQL

Hug

Hug – это очень хороший микрофреймворк, который позволяет создавать API, CLI и локальные пакеты. Он позволяет создать API один раз и использовать этот API в любом месте своего кода. Hug использует Cython для компиляции и потребляет ресурсы только тогда, когда это необходимо; это помогает Hug обеспечивать очень высокую производительность.

Image not found or type unknown



Вы можете установить Hug с помощью pip одной командой. Это один из самых чистых способов создания HTTP REST API на Python 3. Для создания API Hug отделяет зависимости версий и интерфейс от бизнес-логики API. Он предлагает множество интерфейсов для упрощения разработки API для разработчиков.

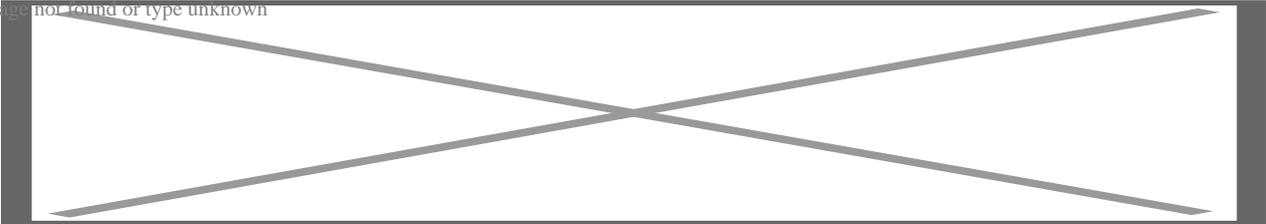
Особенности Hug:

- Придерживается философии write-once use anywhere.
- Предоставляет логику API через несколько интерфейсов (CLI, HTTP или локальную функцию).
- Предоставляет модуль hug.test для тестирования всего стека Python вашего API.
- Чистая и простая для понимания документация

Eve

Eve построен на базе Flask и Cerberus. Цель этого python-фреймворка – сделать разработку API простой и быстрой. Он без особых усилий создает RESTful веб-сервисы, которые хорошо поддаются настройке. Этот фреймворк предпочтителен для проектов среднего размера.

Image not found of type unknown



Особенности Eve:

- Поддерживает полный спектр CRUD-операций
- Поддерживает настраиваемые конечные точки, пагинацию, сортировку, фильтрацию
- Предоставляет расширенные возможности, такие как аутентификация, ограничение скорости, кэширование
- Легко интегрируется с базами данных SQL, MongoDB, Elasticsearch и Neo4js.
- Встроенная поддержка валидации данных
- Поддерживает CORS, кросс-оригинальный обмен ресурсами
- Доказано несколько событий для расширенного протоколирования

Cornice

Cornice – это REST-фреймворк для Pyramid, который предоставляет помощников для создания и документирования REST-сервисов. Pyramid разработан для масштабирования вместе с приложением. Он автоматически следует спецификации

HTTP везде, где это возможно.

Особенности Cornice:

- Один путь в Cornice равен одному сервису.
- Автоматически выбрасывает ошибку 400 после проверки содержимого, если данные недействительны
- Добавляет поддержку CORS (Cross-Origin Resource Sharing) к вашим сервисам
- Использует списки контроля доступа (ACL) Pyramid для авторизации
- Возвращает 406 Not Acceptable для недопустимого типа медиа с сообщением об ошибке, содержащим все поддерживаемые типы контента
- Поддерживает префиксацию URL для всех ваших маршрутов
- Использует Tох, инструмент командной строки на Python для автоматизированного тестирования.

Заключение

Это был список некоторых популярных фреймворков Python для создания API. Я надеюсь, что вы нашли этот список интересным для начала работы. Чтобы оценить все возможности этих Python-фреймворков, вам нужно попробовать их. Итак, чего же вы ждете? Выберите любой из этих REST-фреймворков и начните создавать API на Python.

Дата Создания

29.06.2023