



13 лучших фреймворков Python для создания приложений для малых и крупных предприятий

Описание

Python – это одновременно и забавная игрушка, и пугающий огнемет. То же самое можно сказать и о том, что вы можете делать с Python. Python любят и любители, и ученые, и архитекторы. С ним чертовски легко начать работу, он обладает абстракциями высшего порядка и возможностями метапрограммирования для создания больших и сложных систем, а также имеет множество библиотек для выполнения практически любых задач. Конечно, есть ограничения, когда дело доходит до параллелизма и сильной типизации, но их можно обойти. В этой статье мы рассмотрим некоторые из лучших фреймворков Python для создания больших и малых веб-приложений.

Django

Фреймворк Django выдержал испытание временем и стал основным веб-фреймворком для сообщества Python. Если бы вы напали на Python-разработчика во сне и заставили его под дулом пистолета создавать веб-приложение, он бы автоматически потянулся к Django, как рубинист потянется к Rails.

The web framework for
perfectionists with deadlines.[OVERVIEW](#) [DOWNLOAD](#) [DOCUMENTATION](#) [NEWS](#) [COMMUNITY](#) [CODE](#) [ABOUT](#) [DONATE](#)

Django makes it easier to build better
Web apps more quickly and with less
code.

[Get started with Django](#)

И на это есть веская причина. Django – это, как гласит слоган, “веб-фреймворк для перфекционистов со сроками”. Это так называемый “фреймворк с батарейками” (подобно тому, как Python является языком с батарейками), который предоставляет все общие функции из коробки. Благодаря этим функциям Django значительно сокращает время разработки:

- Удобный и приятный ORM, с миграциями, создаваемыми и применяемыми фреймворком автоматически.
- Подмостки для автоматической генерации панели администратора на основе ваших моделей.
- Поддержка cookies, сессий, промежуточного ПО, шаблонов и т. д.
- Такие функции безопасности, как предотвращение XSS, предотвращение CSRF и т. д., применяются автоматически.
- Работает практически со всеми существующими базами данных (легко найти адаптеры там, где нет официальной поддержки).
- Первоклассная поддержка географических данных и пространственных запросов с помощью GeoDjango

И многое, многое другое. Достаточно сказать, что Django – это полноценный, дружелюбный веб-фреймворк.

Подходит ли вам Django? Безусловно, да. Django отлично подходит для всех случаев использования, будь то быстрое создание прототипа или планирование корпоративного приложения. Единственная неровность, с которой вы столкнетесь, – это структура фреймворка. Поскольку Django изгибается, чтобы сделать разработку быстрой и легкой для вас, он навязывает разработчику свою структуру (концепция, называемая “конвенция над конфигурацией”), с которой вы можете не

согласиться. Например, если вы захотите заменить Django ORM на что-то другое (скажем, SQL Alchemy), будьте готовы к сюрпризам. Хотите стать полнофункциональным разработчиком на Django и Python? – Ознакомьтесь с этим фантастическим онлайн-курсом.

Flask

Хотя Django занимает почти все позиции в веб-разработке на Python, Flask является сильным соперником.



В отличие от Django, Flask – это “микрофреймворк”, что означает, что он фокусируется на выполнении нескольких минимальных вещей, а остальное оставляет на ваше усмотрение. Это “остальное зависит от вас” может стать источником разочарования или радости, в зависимости от ваших целей. Для тех, кто знает, что делает, и хочет с любовью создавать свои веб-приложения, выбирая компоненты по своему вкусу, Flask – просто находка. Flask предлагает следующие возможности:

- Маршрутизация, шаблонизация, управление сессиями и другие полезные функции.
- Полная поддержка юнит-тестирования
- Минимальная, подключаемая архитектура
- Первоклассная поддержка REST
- Поддержка Blueprints, уникального подхода Flask к архитектуре миниатюрных веб-приложений
- Выберите пакеты для ORM, миграции и т. д.

- Гибкая структура приложения – размещайте файлы там, где они наиболее удобны для вас.
- Обслуживание статических файлов
- Соответствие требованиям РГСИ

Подходит ли вам Flask? Как уже говорилось, Flask – это минимальный веб-фреймворк, в котором все разбито на отдельные компоненты, которые можно менять местами. Если вы спешите создать прототип, вы потратите много времени на принятие тривиальных решений по базе данных, структуре папок, маршрутизации и т. д., что может оказаться непродуктивным. Flask лучше всего работает, когда вы переходите к стабильному, серьезному проекту среднего и крупного масштаба, особенно с REST API.

Bottle

Если вам показалось, что подход Flask недостаточно минимален, и вы все еще хотите больше контроля, поприветствуйте Bottle.



Bottle отсекает еще больше, вплоть до того, что единственной зависимостью является стандартная библиотека Python. Это означает, что не нужно устанавливать `pip install this` или `pip install that`, хотя, скорее всего, вам это понадобится в ближайшее время. Вот почему Bottle выделяется для некоторых людей:

- Однофайловое развертывание. Да, все ваше приложение живет в одном файле “.py”.
- Никаких внешних зависимостей. Если у вас установлена нужная версия Python, вы можете приступить к работе.

- Поставляет свой шаблонизатор, который может быть заменен на Jinja2, Mako или Cheetah.
- Поддержка форм, заголовков, файлов cookie и загрузки файлов.
- Встроенный веб-сервер, который можно легко заменить.

Бутылка для вас? Если вы создаете действительно небольшое приложение (скажем, менее 500 строк кода) без особых требований, Bottle может иметь для вас большой смысл. Это совершенно несерьезный подход к созданию веб-приложений, но на практике Bottle вам скорее помешает, чем поможет. Причина в том, что реальный мир постоянно меняется, и не успеете вы оглянуться. На вашу голову посыплются новые требования. В этот момент сведение всего в один файл превратится в рутину. Если вы думаете, что Bottle и Flask почти похожи, то вы правы. Предложения об их слиянии появились еще в 2012 году, и даже Армин, создатель Flask, согласен с этим. Однако Марсель, создатель Bottle, придерживается строгих идеологических различий из-за однофайлового подхода и настаивает на том, чтобы эти два проекта оставались отдельными.

Zope

Zope сложно объяснить, но я попытаюсь. Да, это веб-фреймворк, который можно использовать для создания больших и малых приложений, но это еще не все. Zope – это скорее платформа, состоящая из инструментов (в том числе и веб-фреймворков), созданных в рамках философии Zope и поддерживаемых Zope Corporation.



В Zope есть несколько интересных компонентов и функций, подходящих для разработки корпоративных приложений:

- Архитектура регистрации и обнаружения компонентов для настройки большого приложения.
- ZODB – (единственная) объектная база данных для Python, позволяющая хранить объекты нативно.
- Полноценная основа и стандарты для систем управления контентом
- Набор фреймворков для веб-приложений – канонический фреймворк по-прежнему называется Zope, хотя на его основе было создано несколько новых фреймворков (например, Grok).
- Строгие стандарты разработки, выпуска и сопровождения программного обеспечения.

Подходит ли вам Zope? Если вам нужна высокоструктурированная среда для создания действительно больших приложений, то Zope – отличный вариант. Тем не менее, вы столкнетесь и с достаточной долей проблем. Несмотря на то, что Zope продолжает развиваться, сообщество очень малочисленно, и многие разработчики Python даже не слышали о нем. Найти учебники и подробную документацию непросто, поэтому будьте готовы к тому, что вам придется немало покопаться (хотя сообщество очень полезно!). Кроме того, разработчики на Python, с которыми вы

столкнетесь, могут не захотеть изучать Zope и “разбавить” свой набор навыков.

TurboGears

TurboGears – это очень гибкий фреймворк с действительно композитной архитектурой. Это фреймворк, который масштабируется в зависимости от ваших потребностей, и вы можете использовать его для создания однофайлового приложения или многопользовательского гиганта.



TurboGears имеет несколько элегантных функций, некоторые из которых либо отсутствуют в популярных фреймворках (например, Django), либо их сложно создать:

- Первоклассная поддержка нескольких баз данных
- Транзакции с несколькими базами данных
- Высокая модульность – начните с одного файла и масштабируйте столько, сколько вам нужно.
- Мощный ORM (SQLAlchemy, который является более зрелым и способным, чем ORM в Django)
- Подключаемая архитектура, основанная на спецификации WSGI
- Встроенная поддержка шардинга баз данных
- Интерфейс, управляемый функциями, в отличие от глубоких, жестких объектно-ориентированных иерархий.

TurboGears для вас? Если вы хотите разрабатывать с удовольствием и хотите получить проверенный, зрелый и надежный фреймворк вдали от шума, создаваемого СМИ, “потрясающий, нового поколения” и все такое, TurboGears отлично подойдет. Он пользуется большим уважением в сообществе и имеет полную, обширную документацию. Конечно, TurboGears не имеет своего мнения, что означает, что время первоначальной установки и настройки может быть больше, но

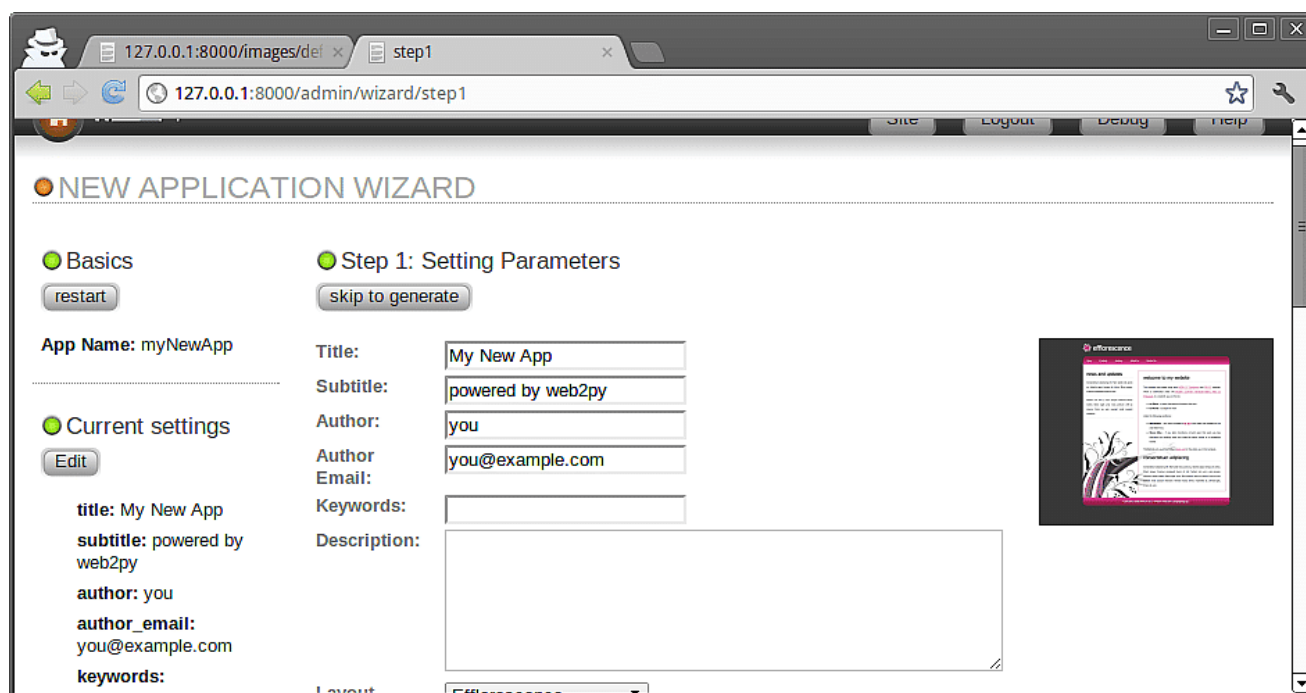
это идеальный фреймворк для разработки корпоративных приложений.

Web2py

Web2py начинался как хобби-проект и был выпущен в 2007 году. Его цель – сделать веб-разработку простой и доступной для всех.

WEB2PY

В результате Web2py доводит подход “нулевой зависимости” до крайности – у него нет требований, его не нужно устанавливать, и он включает в себя полнофункциональный веб-редактор для разработки, управления базами данных, а также развертывания.



Можно считать, что это почти Android Studio, которая представляет собой скорее полноценную среду, чем просто фреймворк. Некоторые приятные особенности, которыми обладает Web2py, таковы:

- Практически никакого обучения.

- Минимальное ядро (всего 12 предметов), которое можно даже запомнить!
- Шаблонирование на Pure-Python
- Защита от XSS, CSRF и других атак
- Приятный и последовательный API

Подходит ли вам Web2py? Web2py – стабильный и интересный фреймворк, но его трудно рекомендовать на фоне других вариантов, таких как Django, Flask и т. д. Рабочих мест почти нет, а история с юнит-тестированием не очень хороша. Тем не менее, вам может понравиться API-код и общий опыт, который предлагает фреймворк, особенно если вы создаете REST API.

CherryPy

CherryPy – еще один микрофреймворк, цель которого – обеспечить базовую функциональность фреймворка и быстро уйти с дороги.



Хотя его можно сравнить с другими микрофреймворками, такими как Flask, CherryPy может похвастаться некоторыми отличиями:

- Он содержит встроенный многопоточный сервер (то, что остается в списке желаний Flask).

- На одном веб-сервере может быть размещено несколько приложений!
- Представляйте свое приложение как WSGI-приложение (для взаимодействия с другими WSGI-приложениями) или как обычный HTTP-сервер (что лучше работает).
- Первоклассная поддержка профилирования и юнит-тестирования
- Работает на PyPy (для настоящих ценителей производительности), Jython и даже Android

CherryPy делает все это, а также все то, что вы ожидаете от веб-фреймворка.

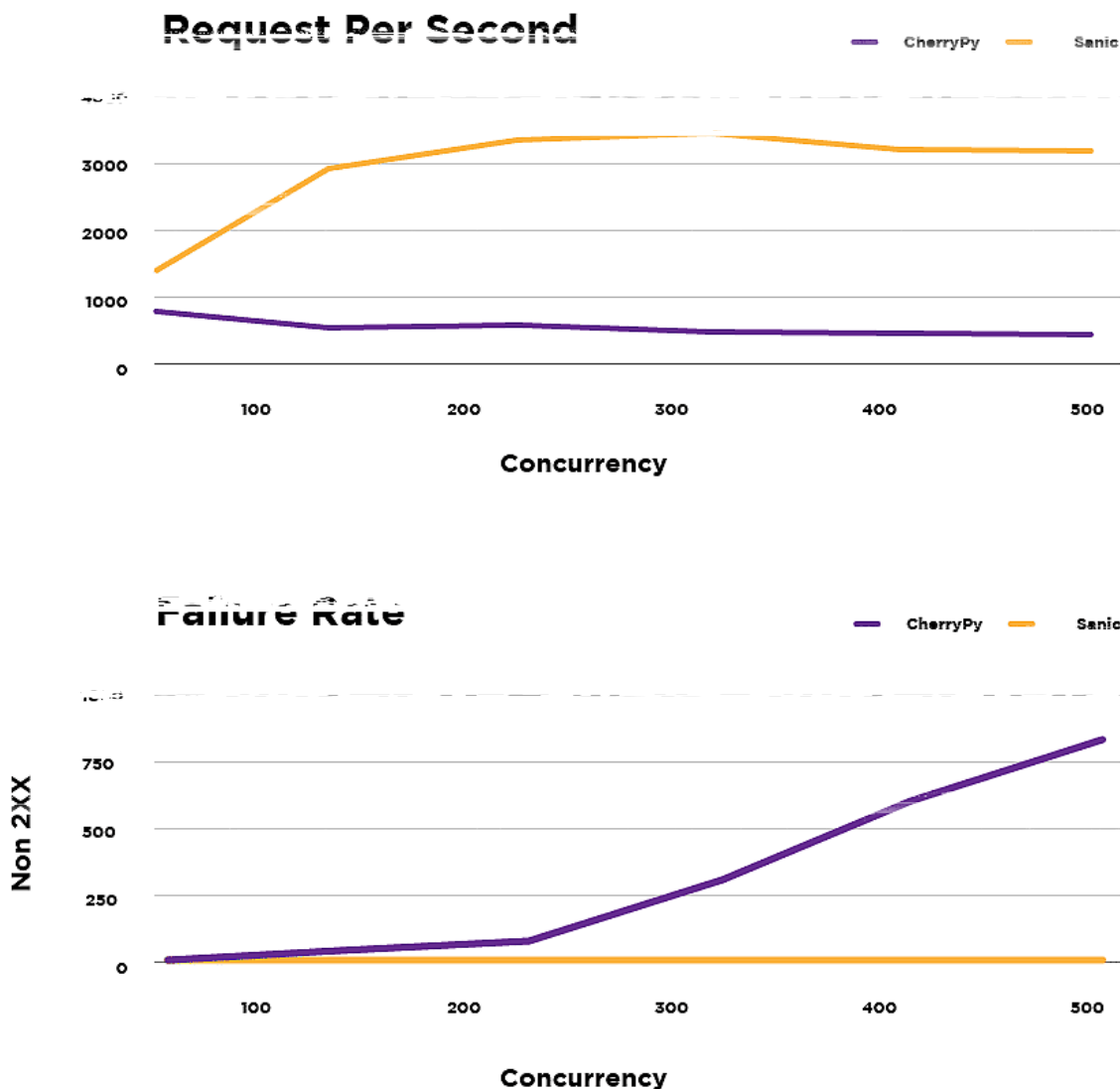
Подходит ли вам CherryPy? Если вы в основном создаете RESTful-сервисы, CherryPy – более серьезный соперник, чем Flask. Это фреймворк десятилетней давности, который хорошо развился и подходит как для небольших, так и для крупных приложений.

Sanic

С появлением Node и его асинхронной модели программирования многие сообщества почувствовали себя отстающими, в том числе и Python. В ответ на это на радаре появился шквал асинхронных веб-фреймворков, и Sanic – один из них.



Sanic сильно вдохновлен Flask, до такой степени, что позаимствовал декораторы маршрутов, Blueprints и другие основы. И им не стыдно в этом признаться. Если вы поклонник Flask, то Sanic предлагает настоящий неблокируемый ввод-вывод, чтобы соответствовать уровню производительности Node-приложений. Другими словами, Sanic – это Flask с поддержкой `async/await`! По сравнению с CherryPy, Sanic обладает невероятным преимуществом в производительности (только представьте, как он справится с Flask!). Посмотрите на следующие результаты, протестированные DataWeave:



Как видите, как только число параллельных операций начинает превышать 50 в секунду, CherryPy практически задыхается и выдает высокий процент отказов.

Вам подходит Sanic? Хотя характеристики производительности Sanic превосходят все остальные, он может оказаться не лучшим выбором для вашего следующего проекта. Основная причина – отсутствие асинхронных библиотек. Большинство существующих инструментов и библиотек для Python были написаны для однопоточной версии CPython, без учета высокого параллелизма или асинхронных операций. Если, например, ваш любимый ORM не поддерживает асинхронные операции, весь смысл использования Sanic сводится на нет. По причине зрелости и доступности мы больше не будем рассматривать асинхронные фреймворки в Python.

Masonite

Я познакомился с этим фреймворком некоторое время назад и подумал, что это шаг в правильном направлении. С тех пор была выпущена версия 2.0, и я чувствую, что наконец-то наступило время, чтобы подарить Masonite немного любви.



Masonite

Проще говоря, Masonite – это Python-версия Laravel (знаменитого PHP-фреймворка, если вы не знали). Почему это важно? Это важно, потому что Laravel был построен

на принципах Ruby on Rails, и вместе эти два фреймворка позволяют разработчикам, не использующим Ruby, испытать “путь Rails”. Разработчики Laravel (и в некоторой степени Rails) будут чувствовать себя как дома и смогут приступить к работе буквально в мгновение ока. Когда я попробовал Masonite (а я отправил пару проблем, включая ошибку!), я смог создавать REST API, не задумываясь, потому что все делала моя мышечная память Laravel. Будучи фреймворком с полным стеком батарей, Masonite предлагает несколько интересных вещей:

- ORM в стиле Active-record
- Миграции баз данных (которые, в отличие от Django, должны быть созданы разработчиком)
- Мощный IoC-контейнер для инъекции зависимостей
- Собственный CLI (называемый “craft”) для создания лесов и выполнения задач
- Первоклассная поддержка модульного тестирования

Самым большим “конкурентом” Masonite является Django, поскольку сообщество делает все возможное, чтобы представить этот фреймворк как простой, восхитительный и следующий за ним. Превзойдет ли он Django, покажет время (как по мне, у него есть все шансы).

Вам подходит Masonite? Masonite все еще ребенок по сравнению с Django, поэтому его никак нельзя рекомендовать вместо Django. Тем не менее, если вы любите Rails (или Laravel), вы оцените то, что предлагает Masonite. Он идеально подходит для быстрого создания прототипов, в которых все должно быть предварительно настроено и легко переключаться.

Falcon

Falcon – это веб-фреймворк на языке Python, предназначенный для работы с крупномасштабными микросервисами. В библиотеке есть полная система с шаблонами, бесплатными пакетами и дополнениями, которые вы можете использовать в своем проекте. Falcon – это отличная отправная точка для получения всех необходимых ресурсов.



Проект также имеет сильное сообщество, в котором пользователи обсуждают дизайн и развитие этого фреймворка. Falcon предлагает следующие возможности;

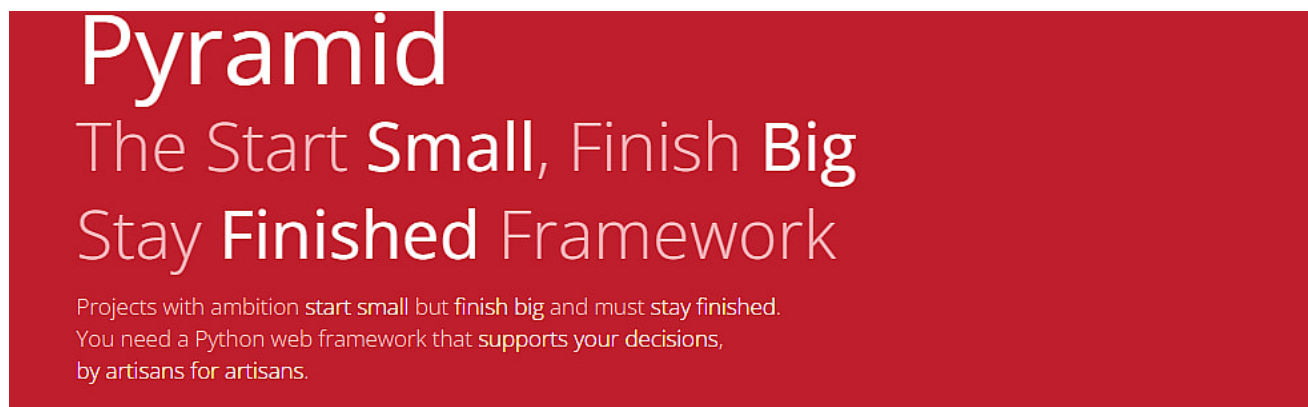
- **Возможность отладки:** Вы можете легко отладить приложение Falcon, поскольку оно сообщает вам, какие входы приводят к каким выходам. К счастью, все необработанные исключения в Falcon никогда не инкапсулируются и не маскируются.
- **Гибкость:** Falcon дает вам контроль над деталями реализации, позволяя настраивать приложение по своему усмотрению. Минималистичный подход облегчает отладку и настройку приложения по своему усмотрению.
- **Быстрота:** Было доказано, что Falcon выполняет запросы быстрее, чем другие фреймворки. Этот фреймворк отлично работает с PyPy.
- **Поддержка ASGI и WSGI:** Вы можете использовать Falcon независимо от того, являетесь ли вы поклонником асинхронного интерфейса шлюза сервера (ASGI) или интерфейса шлюза веб-сервера (WSGI). WSGI – синхронный, то есть он обрабатывает запросы один за другим (последовательно). ASGI – асинхронный, в котором запросы обрабатываются одновременно.

Подходит ли вам Falcon?

Falcon – отличный выбор для создания микросервисов и критически важных REST API. Тот факт, что этот фреймворк создан для быстрой работы, делает его отличным выбором для приложений, обрабатывающих множество транзакций за один раз.

Pyramid

Pyramid представлена как “Начать с малого, закончить большим, остаться законченным”. Если вы любите учиться, вы найдете Pyramid легкой для начала.



Фреймворк позволяет выбрать подход к безопасности, шаблонам, базе данных и многому другому. Вы даже можете начать со строительного леса. Pyramid предлагает следующие возможности;

- **Масштабируемость:** Тот факт, что вы можете начать с малого, не означает, что ваше приложение останется таким навсегда. В Pyramid есть система дополнений, расширений и конфигураций, которая поможет вам масштабировать ваше приложение.
- **Аутентификация и авторизация:** Вы можете защитить свое приложение от несанкционированного доступа с помощью встроенных функций авторизации и аутентификации.
- **Расширяемость:** Pyramid разработана как полный пакет для всего жизненного цикла приложения. Платформа стремится к стабильности API, чтобы вы всегда могли доверять API.
- **Высокая настраиваемость:** Вы можете настраивать или переопределять основной код без использования форков. Такой подход позволяет создавать приложения, соответствующие вашим потребностям.

Пирамида для вас? Pyramid следует использовать при создании контент-ориентированных и масштабных веб-приложений. Модульная и гибкая архитектура позволяет легко масштабировать приложения. Pyramid предлагает простую интеграцию с базами данных и имеет встроенную поддержку гибкой маршрутизации, что делает его идеальным выбором для контентных сайтов, таких как новостные платформы и блоги.

Tornado

Помимо того, что Tornado является фреймворком Python, это еще и асинхронная сетевая библиотека. В отличие от большинства фреймворков Python, основанных на WSGI, Tornado работает только с одним потоком на процесс. Вы все еще можете получить поддержку WSGI через `tornado.wsgi`. Однако большинство разработок базируется на `tornado.web`, собственном интерфейсе Tornado. Tornado предлагает следующие возможности;

- **Неблокируемый сетевой ввод-вывод:** Эта библиотека построена на базе библиотеки `asyncio`. Это позволяет ей обрабатывать тысячи соединений за раз без особых накладных расходов.
- **Безопасность и аутентификация:** Tornado обладает такими функциями безопасности, как защита от подделки межсайтовых запросов, HTTPS и cookies. Эта же библиотека имеет встроенную поддержку аутентификации.
- **Масштабируемость:** Используя Tornado, вы можете увеличивать мощность или добавлять новые серверы по мере необходимости.
- **Поддержка WebSocket:** Связь между сервером и клиентом в реальном времени осуществляется с помощью встроенной реализации WebSocket.

Торнадо для вас? Tornado станет отличным вариантом, если вы хотите создавать высокопроизводительные веб-сервисы. Этот Python-фреймворк отличается высокой масштабируемостью, а функция неблокирующего ввода-вывода делает его идеальным вариантом, если вы создаете сервис, который будет обрабатывать большое количество одновременных соединений. Tornado также подходит для приложений, выполняющих большое количество операций ввода-вывода, таких как обработка больших файлов или получение данных из внешних API.

CubicWeb

CubicWeb – это семантический фреймворк для веб-приложений, который позволяет разработчикам создавать приложения, следуя принципам объектно-ориентированного проектирования.

CubicWeb

— Define your knowledge
CubicWeb builds your
application

CubicWeb is an open-source (LGPL) Python Web framework built with Semantic Web principles in mind. It promotes reusability through the usage of components (called cubes). It is the most appropriate choice for data-centric applications.

Фреймворк содержит набор библиотек и инструментов, которые разработчики могут использовать для создания мощных приложений.

CubicWeb оснащен следующими функциями;

- **Многократно используемые компоненты:** Вы можете сэкономить время разработки, повторно используя компоненты во всех приложениях.
- **Использует язык запросов:** CubicWeb использует RQL, язык запросов в фоновом режиме.
- **Совместимость с несколькими базами данных:** Воспользуйтесь совместимостью CubicWeb с Mercurial, каталогами LDAP и базами данных SQL.
- **Настраиваемость:** CubicWeb отличается гибкостью и настраиваемостью, что позволяет вам контролировать работу вашего приложения.

Подходит ли вам CubicWeb? CubicWeb – отличный выбор, если вы хотите создать приложения, требующие гибких возможностей по формированию запросов и моделированию данных. Таким образом, с помощью CubicWeb можно создавать приложения для социальных сетей, управления знаниями и научными данными.

Заключение

Не существует недостатка в больших и малых фреймворках Python. Хотя для небольшого проекта можно подобрать практически все, что угодно, корпоративное приложение предъявляет требования, которые не многие из этих фреймворков могут удовлетворить. Если вы спросите меня, то для корпоративной разработки на ум приходят Django (в некоторой степени), Zope и TurboGears. И даже среди них я склоняюсь к TurboGears. При этом любой архитектор, стоящий своей соли, может взять микрофреймворк и развернуть свою архитектуру. Именно это и происходит на практике, что объясняет успех Flask и подобных идей.

Дата Создания

29.03.2024