

ПРОГРАММИРОВАНИЕ БЭКЕНДА НА Python

Практическое
руководство

Тим Питерс

Оглавление

Предисловие	20
Пролог	23
Предварительные требования	25
Начальные навыки.....	25
Использование кодов.....	25
Благодарности	27
ГЛАВА 1. Основы разработки внутренних компонентов	28
Введение.....	28
Описание разработки внутренних компонентов.....	29
Основные внутренние компоненты.....	29
Сервер.....	29
База данных.....	31
Интерфейсы прикладного программирования (API).....	32
Веб-фреймворк.....	33
Middleware (связующее или промежуточное программное обеспечение).....	35
Кеширование.....	36
Функционирование внутренних компонентов.....	36
Роль Python в разработке внутренних компонентов.....	38
Универсальность и читабельность.....	38
Универсальная стандартная библиотека.....	39
Фреймворки и инструменты для разработки внутренних компонентов.....	39
Поддержка асинхронной работы.....	40
Возможность интеграции.....	40
Возможность подключения к базам данных.....	41
Сообщество и ресурсы.....	41
Искусственный интеллект и машинное обучение.....	41

Настройка среды разработки: Python, VS Code и Linux	42
Установка Linux	42
Установка Python	42
Установка Visual Studio Code (VS Code)	43
Настройка VS Code для Python	43
Настройка виртуальной среды	44
Заключительные этапы и тестирование	44
Знакомство с виртуальными средами	45
Что такое виртуальная среда?	45
Зачем нужна виртуальная среда?	46
Как настроить и использовать виртуальную среду?	46
Установка	46
Создание виртуальной среды	46
Активация виртуальной среды	47
Установка пакетов	47
Деактивация	47
Управление зависимостями	47
Передовой опыт	48
Основные принципы работы с интерфейсом командной строки (CLI)	49
Описание интерфейса командной строки	49
Основные команды интерфейса командной строки	49
Советы по использованию интерфейса командной строки	52
Введение в управление версиями с помощью Git	53
Знакомство с управлением версиями и Git	53
Основные операции Git для разработки внутренних компонентов	54
Установка Git	54
Конфигурирование Git	54
Инициализация репозитория	54
Клонирование репозитория	55
Добавление и фиксация файлов	55
Ветвление и слияние	55
Размещение изменений	56
Извлечение обновлений	56
Обработка конфликтов при слиянии	56
Использование журнала Git	56
Библиотека Python Refresher: синтаксис, структуры данных и функции	57
Синтаксис Python	57
Отступы	57
Переменные	57
Комментарии	58
Структуры данных	58
Списки	58
Кортежи	58

Множества	59
Словари	59
Функции	59
Определение функции	59
Вызов функции	60
Параметры	60
Аргументы ключевых слов	60
Произвольные аргументы	61
Лямбда-функции	61
Обработка ошибок	61
Модули и пакеты	62
Импорт модулей	62
Импорт с помощью псевдонимов	62
Оператор импорта <i>from</i>	62
Передовой опыт программирования на Python	63
Поддержка принципов «Дзен Python»	63
Соблюдение стандарта PEP 8	63
Написание документальных строк	64
Использование встроенных функций и библиотек Python	64
Использование списков и генераторов выражений	65
Обработка ошибок с помощью исключений	65
Использование менеджеров контекста для управления ресурсами	66
Функция должна выполнять одну задачу и иметь минимально возможный размер	66
Избежание преждевременной оптимизации	66
Использование контроля версий	67
Тестирование своего кода	67
Резюме	67

ГЛАВА 2. Создание первого веб-приложения с помощью

Flask	69
Введение	69
Основы Flask	70
Что представляет собой Flask?	70
Основные возможности Flask	71
Простота	71
Гибкость	71
Сервер разработки и отладчик	72
Основан на Юникоде	72
Документация	72
Запуск и работа с Flask	72
Маршрутизация	74
Шаблоны	74

Настройка окружения Flask	76
Создание проекта	76
Настройка Flask	77
Установка Flask	77
Создание приложения Flask	77
Создание простого представления	77
Управление зависимостями с помощью Pip	77
Настройка приложения Flask	78
Создание файла конфигурации	78
Загрузка конфигурации	78
Запуск приложения Flask	78
Командная строка Flask	78
Использование скрипта	79
Маршрутизация и представления	79
Определение маршрутов	80
Динамические маршруты	80
Методы HTTP	81
Создание URL-адресов	81
Обработка ошибок	82
Эффективное сочетание маршрутов и представлений	83
Шаблоны и статические файлы	83
Описание шаблонов	84
Управление статическими файлами	85
Организация статических файлов	86
Обслуживание статических файлов	86
Работа с формами и загрузка файлов	88
Flask — работа с формами	88
Создание HTML-формы	88
Создание маршрута для отображения формы	88
Обработка данных формы	88
Загрузка файлов	89
Модификация HTML-формы для загрузки файлов	89
Обработка загрузки файлов в Flask	89
Основы интеграции баз данных	90
Настройка SQLAlchemy с помощью Flask	92
Установка Flask-SQLAlchemy	92
Настройка приложения	92
Определение моделей	92
Создание базы данных	93
Взаимодействие с базой данных	93
Вставка данных	93
Запрос данных	93
Обновление данных	94
Удаление данных	94

Работа с отношениями.....	94
Извлечение связанных данных.....	95
Использование опций запросов.....	95
Введение в расширения Flask.....	96
Описание расширений Flask.....	96
Flask-WTF.....	97
Flask-SQLAlchemy.....	97
Flask-Migrate.....	97
Flask-Login.....	98
Flask-Mail.....	98
Flask-RESTful.....	98
Использование расширений Flask.....	99
Развертывание приложения Flask.....	100
Подготовка приложения Flask к развертыванию.....	100
Выбор хостинга.....	101
Настройка веб-сервера и сервера приложений WSGI.....	101
Использование Heroku.....	102
Подготовка приложения.....	102
Настройка Git-репозитория.....	102
Создание приложения Heroku.....	102
Выполнение проверки развертывания.....	103
Резюме.....	103
ГЛАВА 3. Дополнительные возможности Flask.....	105
Введение.....	105
Создание модульной структуры больших приложений с помощью Flask Blueprints.....	106
Для чего предназначен Flask Blueprints?.....	106
Создание и регистрация эскизов.....	107
Определение эскиза.....	107
Регистрация эскиза.....	107
Структурирование представлений в эскизах.....	108
Использование эскизов.....	108
Шаблон Application Factory во Flask.....	109
Описание шаблона Application Factory.....	109
Использование шаблона Application Factory.....	110
Создание функции <i>Factory</i>	110
Настройка параметров конфигурации.....	110
Регистрация сценариев и расширений.....	111
Реализация RESTful-сервисов с помощью Flask-RESTful.....	112
Введение в Flask-RESTful.....	112
Функционирование Flask-RESTful.....	113

Создание REST API с помощью Flask-RESTful	113
Тестирование API.....	115
Аутентификация и авторизация пользователей.....	115
Значение аутентификации, авторизации и управления сессиями	115
Аутентификация	115
Авторизация.....	116
Управление сессиями.....	116
Реализация аутентификации в Flask	116
Установка Flask-Login.....	116
Настройка Flask-Login	116
Функция загрузчика пользователя	117
Определение модели пользователя.....	117
Создание маршрутов аутентификации	117
Выполнение авторизации	118
Управление доступом на основе ролей (RBAC).....	118
Проверка прав доступа	119
Обработка ошибок и ведение журнала.....	119
Flask — обработка ошибок.....	120
Обработка ошибок приложения.....	120
Обработка исключений.....	120
Flask — ведение журнала	121
Использование протоколирования для данных запроса	121
Пользовательские средства регистрации	122
Методы оптимизации производительности	122
Оптимизация баз данных.....	123
Оптимизация запросов.....	123
Пакетные вставки и обновления	123
Оптимизация обработки запросов	123
Эффективная последовательность данных	123
Асинхронные обработчики.....	124
Кеширование ответов	125
Настройка приложений и веб-серверов	125
Мониторинг и профилирование.....	126
Интеграция приложений Flask с Docker	127
Установка Docker.....	127
Создание файла Dockerfile.....	128
Создание файла .dockerignore.....	129
Сборка образа Docker.....	129
Запуск контейнера Docker	129
Тестирование контейнера Docker	129
Резюме.....	130

ГЛАВА 4. Введение в FastAPI	132
Введение.....	132
Переход на FastAPI.....	133
Основные различия между FastAPI и Flask.....	133
Производительность	133
Аннотации типов данных и автоматическая валидация данных.....	134
Внедрение зависимостей	135
Встроенная интерактивная документация по API	136
Современные возможности Python	136
Создание RESTful API с помощью FastAPI	136
Определение конечных точек RESTful	137
Параметры пути и строки запросов.....	138
Параметры пути.....	138
Параметры запросов	138
Использование тела запроса.....	139
Обработка ответа	139
Пользовательские коды состояния	139
Заголовки ответа	139
Обработка ошибок.....	140
Внедрение зависимостей	140
Описание внедрения зависимостей в FastAPI.....	141
Как работает внедрение зависимостей в FastAPI?	141
Определение зависимостей	141
Использование зависимостей в обработчиках маршрутов	142
Работа с зависимостями в больших приложениях	143
Расширенная интеграция баз данных с SQLAlchemy.....	143
Настройка SQLAlchemy с помощью FastAPI.....	144
Установка необходимых пакетов.....	144
Настройка URL базы данных	144
Создание базы данных и таблиц	144
Интеграция SQLAlchemy с FastAPI	145
Зависимость от сеанса работы с базой данных.....	145
Выполнение операций с базой данных.....	145
Асинхронная обработка.....	146
Настройка асинхронного подключения к базе данных	146
Подключение и отключение событий	146
Использование асинхронных запросов	146
Реализация фоновых задач	147
Основы фоновых задач в FastAPI	147
Как определить фоновые задачи?.....	147
Реализация более сложных фоновых операций.....	148
Настройка Celery	148
Запуск задач Celery из FastAPI	149

Интеграция FastAPI и Docker	150
Установка Docker	150
Создание Docker-файла для FastAPI.....	150
Создание образа Docker.....	151
Запуск приложения FastAPI в контейнере Docker.....	152
Проверка работоспособности приложения.....	152
Резюме.....	152
ГЛАВА 5. Работа с базами данных.....	154
Введение.....	154
MySQL и PostgreSQL	155
MySQL	156
PostgreSQL	156
Выбор между MySQL и PostgreSQL.....	157
MongoDB.....	159
Описание MongoDB и модели документов.....	159
Документы	159
Коллекции	159
Преимущества MongoDB	160
Ключевые особенности MongoDB	160
Интеграция MongoDB	161
Установка.....	161
Подключение к MongoDB	161
Операции.....	161
Принципы проектирования баз данных	162
Основные принципы проектирования баз данных.....	162
Нормализация	162
Модель отношений между сущностями.....	163
Пример разработки схемы базы данных	163
SQL-код для создания таблиц	164
Операции CRUD.....	165
Создание записей (CREATE)	165
Чтение записей (READ).....	166
Обновление записей (UPDATE).....	166
Удаление записей (DELETE).....	167
Рекомендации по выполнению операций CRUD	167
Расширенные методы обработки запросов	167
Подзапросы.....	168
Пример программы: «Поиск курсов, на которые не зарегистрировался ни один студент».....	168
Объединения.....	168
Пример программы: «Список студентов с информацией о курсах»	169

Агрегатные функции SQL	169
Пример программы: «Подсчет количества студентов, записанных на каждый курс»	169
Расширенная фильтрация с помощью оператора <i>HAVING</i>	170
Пример программы: «Курсы с более чем 5 студентами».....	170
Оконные функции	170
Пример программы: «Распределение студентов по дате зачисления на каждый курс»	170
Миграция баз данных и контроль версий.....	171
Основные сведения о миграции баз данных	172
Настройка Alembic.....	172
Установка Alembic	172
Инициализация Alembic.....	172
Настройка Alembic	173
Создание и применение миграций.....	173
Создание миграции	173
Редактирование сценария миграции.....	173
Применение миграций	173
Управление изменениями схемы базы данных.....	174
Создание версий	174
Совместная работа	174
Развертывание	174
Интеграция баз данных с ORM на Python.....	175
Обзор ORM для Python	175
Обзор SQLAlchemy	175
SQLAlchemy Core	175
SQLAlchemy ORM.....	176
Интеграция SQLAlchemy ORM с Python.....	176
Установка.....	176
Определение моделей	176
Создание сессии	177
Выполнение операций с базой данных	177
Стратегии кеширования для оптимизации баз данных	178
Типы кеширования.....	178
Кеширование результатов	178
Кеширование объектов	178
Кеширование планов запросов.....	179
Реализация кеширования в веб-приложениях	180
Redis в качестве кеша.....	180
Стратегия кеширования.....	180
Рекомендации по эффективному кешированию.....	181
Резюме.....	181

ГЛАВА 6. Асинхронное программирование в Python.....	184
Введение.....	184
Общее описание асинхронного программирования.....	186
Что такое асинхронное программирование?	186
Асинхронное программирование для разработки внутренних компонентов	186
Примеры использования асинхронного программирования.....	187
Веб-серверы	187
Обработка данных	187
Работа приложения в режиме реального времени.....	187
Архитектура микросервисов	188
Как работает асинхронное программирование?.....	188
Цикл событий	188
Корутины (сопрограммы или асинхронные функции).....	188
Задачи и фьючерсы	188
Пример программы: «Асинхронное выполнение».....	189
Основы <code>asyncio</code>	189
Синтаксис <code>async</code>	190
Синтаксис <code>await</code>	190
Структура программы <code>asyncio</code>	190
Пример программы: «Интеграция библиотеки <code>asyncio</code> в приложение для университета».....	191
Разработка асинхронных веб-приложений.....	192
Пример программы: «Асинхронное университетское приложение».....	193
Настройка и установка.....	193
Определение конечных точек асинхронной связи.....	193
Асинхронный доступ к базам данных	194
Описание асинхронного доступа к базам данных	194
Настройка и установка.....	195
Интеграция асинхронных операций с базами данных в веб-фреймворке <code>FastAPI</code>	195
Конфигурация	195
Функции асинхронной базы данных.....	196
Использование асинхронных функций в маршрутах	197
Внедрение веб-сокетов	197
<code>WebSocket</code> — что это?.....	197
Реализация <code>WebSocket</code> в <code>FastAPI</code>	198
Настройка и базовая конечная точка <code>WebSocket</code>	198
Установка <code>FastAPI</code> и <code>Uvicorn</code>	199
Определение конечной точки <code>WebSocket</code>	199
Обновление записей в режиме реального времени	199

Сохранение соединений WebSocket	200
Интеграция с процессом регистрации	200
Передовой опыт и шаблоны Async	201
Обработка ошибок в асинхронном коде	201
Пример программы: «Корректная обработка ошибок базы данных»	201
Управление параллельными процессами и отмена задач	202
Пример программы: «Отмена устаревших запросов к базе данных»	202
Использование менеджеров контекста для управления ресурсами	203
Пример программы: «Асинхронный менеджер контекста для подключения к базе данных»	203
Разделение и модульное построение кода	203
Пример программы: «Модульная обработка WebSocket»	203
Отладка асинхронных приложений	204
Инструменты и методы отладки	205
Пример программы: «Ведение подробного журнала»	205
Пример программы: «Использование сообщений <i>print</i> для получения немедленной обратной связи»	205
Пример программы: «Включение режима отладки <i>asyncio</i> »	206
Резюме	207

ГЛАВА 7. Организация работы с пользователями и их безопасность	209
Введение	209
Проектирование систем аутентификации пользователей	210
Определение требований	210
Компоненты системы аутентификации пользователей	211
Пример программы: «Подключение аутентификации пользователей»	212
Реализация стандартов OAuth и JWT	213
Введение в OAuth	213
Введение в веб-токены JSON (JWT)	214
Пример программы: «Реализация OAuth и JWT»	214
Настройка OAuth с помощью провайдера	214
Интеграция OAuth в приложение	215
Реализация JWT для управления сеансами	215
Контроль доступа на основе ролей (RBAC)	216
Определение эффективности RBAC	216
Пример программы: «Внедрение RBAC»	217
Определение ролей и разрешений	217
Настройка среды	217

Определение моделей пользователей и ролей	218
Создание ролей	218
Назначение ролей пользователям	218
Обеспечение проверки ролей	219
Обеспечение безопасности REST API	219
Что необходимо для обеспечения безопасности REST API?	219
Стратегии обеспечения безопасности REST API	220
Использование HTTPS	220
Аутентификация и авторизация	220
Валидация и дезинфекция входных данных	221
Внедрение ограничения скорости	222
Управление сеансами пользователей	222
Основные понятия	222
Реализация безопасного управления сеансами	223
Создание и обработка сеансов	223
Сохранение сеансов	224
Истечение срока действия сеанса	224
Обслуживание и безопасность сеансов	224
Внедрение двухфакторной аутентификации	225
Описание двухфакторной аутентификации	225
Шаги по внедрению 2FA	226
Генерация секретного ключа для пользователя	226
Связывание секретного ключа с приложением Authenticator	227
Проверка TOTP во время входа в систему	227
Интеграция 2FA в процесс входа в систему	227
Резюме	228

ГЛАВА 8. Развертывание внутренних приложений, написанных на языке Python 230

Введение	230
Обзор Docker и контейнеров	231
Описание контейнерной технологии	231
Роль Docker в контейнеризации	231
Доминирующее положение Docker	232
Пример программы: «Использование Docker»	233
Использование Docker для приложений, написанных на языке Python	234
Установка Docker	234
Контейнеризация университетского приложения	236
Подготовка приложения	236
Создание Dockerfile	236
Сборка образа Docker	237
Запуск контейнера Docker	237

Kubernetes для управления приложениями	237
Описание Kubernetes	238
Установка и настройка Kubernetes	238
Установка Minikube	238
Установка kubectl	239
Развертывание университетского приложения на Kubernetes	239
Создание конфигурации развертывания	239
Развертывание приложения	240
Открытие приложения	240
Доступ к приложению	240
CI/CD для внутренних приложений на Python	241
Основные принципы работы CI/CD	241
Использование CI/CD для университетского приложения	241
Настройка контроля версий	241
Выбор инструмента CI/CD	242
Использование Nginx в качестве обратного прокси-сервера	244
Функция Nginx	244
Установка и настройка Nginx	245
Установка Nginx	245
Настройка Nginx в качестве обратного прокси-сервера	245
SSL-сертификаты и настройка HTTPS	247
Описание SSL/TLS и HTTPS	247
Генерация SSL-сертификатов	247
Установка Certbot	248
Получение сертификата	248
Проверка установки SSL-сертификата	248
Конфигурация HTTPS в Nginx	248
Масштабирование приложений на Python	249
Необходимость масштабирования	249
Горизонтальное и вертикальное масштабирование	250
Горизонтальное масштабирование (масштабирование наружу/внутри)	250
Вертикальное масштабирование (масштабирование вверх/вниз)	250
Реализация масштабирования в Kubernetes	250
Определение запросов и лимитов ресурсов	251
Настройка горизонтального автоматического масштабирования подсистем	252
Контроль масштабирования	253
Резюме	253
ГЛАВА 9. Микросервисы и интеграция с облаком	255
Введение	255
Проектирование и разработка микросервисов с помощью Python	256
Разбиение приложения на микросервисы	257

Определение границ сервисов	257
Создание независимых сред.....	257
Разработка API для межсервисного взаимодействия.....	258
Упаковка сервисов в контейнеры	258
Управление микросервисами с помощью Docker и Kubernetes	259
Контейнеризация с помощью Docker.....	259
Упаковка каждого микросервиса в контейнер.....	260
Организация работы с помощью Kubernetes	260
Создание развертываний Kubernetes	261
Управление сервисами с помощью Kubernetes Services	261
Развертывание приложений Python на AWS.....	262
Настройка AWS для университетского приложения	262
Создание учетной записи AWS	262
Настройка IAM (управление идентификацией и доступом).....	263
Настройка AWS CLI.....	263
Развертывание приложения на AWS	264
Вариант 1: использование Elastic Beanstalk	264
Вариант 2: использование EC2	264
Использование бессерверных архитектур с AWS Lambda	265
Описание AWS Lambda	265
Установка AWS Lambda.....	266
Подготовка приложения	266
Создание функции Lambda в AWS	267
Реализация gRPC для взаимодействия микросервисов.....	268
Зачем нужен gRPC	268
Реализация gRPC в микросервисах на Python	269
Определение сервиса с помощью буферов протокола.....	269
Создание кода сервера и клиента.....	270
Реализация сервиса на Python	270
Создание клиента	271
Резюме.....	271

ГЛАВА 10. Брокеры сообщений и асинхронная обработка

задач	273
Введение.....	273
Обзор брокеров сообщений.....	274
Роль брокеров сообщений во внутренних приложениях	274
Redis как брокер сообщений	275
Каким образом Redis обеспечивает обмен сообщениями?.....	275
Интеграция Kafka для обработки данных в реальном времени.....	277
Описание Apache Kafka	277
Основные компоненты Kafka.....	277

Интеграция Kafka для обработки данных в режиме реального времени	278
Установка и настройка Kafka	278
Создание тем.....	278
Реализация производителей и потребителей Kafka.....	278
Асинхронная обработка задач с помощью Celery	280
Знакомство с асинхронной обработкой задач	280
Использование Celery для асинхронной обработки задач	280
Как работает Celery?	281
Использование Celery	281
Установка Celery и Redis	281
Настройка Celery	281
Запуск Celery Worker.....	282
Постановка задач в очередь.....	282
RabbitMQ как альтернативный брокер сообщений	282
Описание RabbitMQ и принцип его работы.....	283
Интеграция RabbitMQ	283
Установка RabbitMQ.....	284
Настройка RabbitMQ в приложении.....	284
Отправка сообщений.....	285
Резюме.....	285
Эпилог.....	287