

Тестирование Django проекта

№ урока: 11 **Курс:** Django Базовый

Средства обучения: Pycharm Community, GIT, аккаунт в GitHub, Python 3.6 и выше, Django проект, Docker образ.

Обзор, цель и назначение урока

Научиться тестировать Django-проект. Рассмотреть подходы тестирования API. Получить знания о абстрактных тестах, как их создавать и использовать. Изучить как писать юнит-тесты для REST API и Graphene. А также рассмотреть pytest и способы генерации документации.

Изучив материал данного занятия, учащийся сможет:

- Создавать полезные тесты.
- Протестировать свой Django-проект.
- Создавать абстрактные тесты.
- Протестировать API.
- Писать тесты с помощью pytest.
- Сгенерировать документацию.

Содержание урока

1. Несколько слов о тестах и их пользе
2. Рассмотрение фреймворка для тестирования unittest
3. Создание mock
4. Рефакторинг тестов и создание абстрактных тестов
5. Написание тестов для REST API
6. Написание тестов для Graphene
7. Тестирование с помощью Postman
8. Рассмотрение пользы от pytest
9. Подключаем к API Swagger

Резюме

- Юнит тесты служат не только для того, чтобы удовлетворить клиента, который запросил покрытие кода в 95%. Юнит тесты это верный помощник программиста. Они нужны для:
 - уверенности в коде - после тестов программист может быть уверен, что он избежал простых ошибок, например IndexError или ValueError.
 - документации - тесты это лучшая документация, они более точно описывают поведение кода.
 - увеличения поддержки кода - другими словами проще обновлять или добавлять что-то в код с тестами.
 - экономии времени - если при тестировании была найдена ошибка, её легче локализовать в коде, для которого написаны тесты.
- **Unittest** — это фреймворк, который позволяет писать модульные тесты. Иными словами, это набор функций, классов и методов для тестирования кода.
- Юнит тесты в Django дают возможность программисту создавать модули в тестовой базе данных, что дает возможность тестировать обработку данных.

- **Mock** — это объект “пустышка”, которым заменяется реальный объект в юнит тестах. Это делается для моделирования окружения программы и для реализации её инкапсуляции при модульном тестировании.
- **Абстрактные тесты** — это тесты, которые только описывают тесты, но сами не запускаются, подобно абстрактным классам. Они нужны для решения DRY принципа, то есть в них можно вынести весь одинаковый функционал.
- Как для REST так и для Graphene для тестирования используется Client. В него передаются запрашиваемые данные, а он в свою очередь должен возвращать необходимый ответ.
- Для тестирования API можно использовать библиотеку snapshottest, которая дает возможность тестировать проект без лишних затрат ресурсов.
- **Postman** - система для написания автоматических тестов под API.
- **Pytest**. Основные преимущества:
 - возможность использовать тестовую базу между тестами;
 - отключать миграции;
 - моделировать настройки базы данных;
 - блокировать базу данных.
- Установка инструмента для документации: `pip install django-rest-swagger`

Закрепление материала

- Для чего нужно тестировать проект?
- Какая польза от абстрактных тестов?
- Что такое Postman?
- Чем pytest отличается от unittest?
- Какие есть подходы к тестированию API?

Дополнительное задание

Задание

Создать тесты для Django-проекта. Добавить тестирование Graphene и REST API.

Самостоятельная деятельность учащегося

Задание 1

Выучить и понять, как эффективно тестировать Django-проект.

Задание 2

Поэкспериментировать с моками. Покрыть несколько модулей тестами на 95%. Рассмотреть, как работает coverage библиотека.

Задание 3

Найти ресурсы для тестирования API (аналоги Postman). Подключить проект к одному из них.

Рекомендуемые ресурсы

Документация:

- <https://www.django-rest-framework.org/topics/documenting-your-api/#third-party-packages>
- <https://docs.graphene-python.org/en/latest/testing/>
- <https://docs.djangoproject.com/en/3.0/topics/testing/advanced/>
- <https://www.django-rest-framework.org/api-guide/testing/>
- <https://learning.getpostman.com/>
- <https://docs.graphene-python.org/projects/django/en/latest/testing/>

- <https://www.soapui.org/>