Тестирование с помощью Pytest

№ урока: 7 **Курс:** Flask

Средства обучения: Python3 и любимая среда разработки, например, PyCharm.

Обзор, цель и назначение урока

Рассмотрим, какие бывают виды тестирования веб-приложений, узнаем, какие инструменты существуют для тестирования веб-приложений в Python и автоматизируем запуск тестов с помощью Continuous Integrations tools.

Изучив материал данного занятия, учащийся сможет:

- Писать определенные виды тестов (unittests, end to end tests) в конкретных случаях.
- Использовать инструмент pytest для тестирования веб-приложений.
- Настраивать CI pipelines с помощью Travis CI для автоматизации запуска тестов.

Содержание урока

1. Виды тестирования веб-приложений
2. Инструменты тестирования веб-приложений в Python
3. Pytest
4. Что такое Continuous Integration
5. Travis CI

Резюме

- **Функциональное тестирование** – процесс оценки поведения приложения, позволяющий определить, все ли разработанные функции ведут себя так, как нужно. Для корректной работы продукта все процессы должны работать так, как это предусмотрено в требованиях: от разграничения прав доступа при авторизации до корректного выхода из системы.
- **Тестирование «белого ящика» (white box)** – тестирование на соответствие программного продукта требованиям со знанием внутренней структуры реализации системы (есть в наличии исходный код и технические спецификации).
- **Тестирование «черного ящика» (black box)** – тестирование на соответствие программного продукта требованиям без знания внутренней структуры реализации системы.
- **Модульное тестирование (unit testing)** – каждая сложная программная система состоит из отдельных частей - модулей, выполняющих ту или иную функцию в составе системы. Для того, чтобы удостовериться в корректной работе всей системы, необходимо вначале протестировать каждый модуль системы по отдельности. В случае возникновения проблем при тестировании системы в целом это позволяет проще выявить модули, вызвавшие проблему, и устранить соответствующие дефекты в них. Такое тестирование модулей по отдельности получило называние модульного тестирования.
- **Регрессионное тестирование (regression testing)** – регрессионное тестирование проводится с целью проверить, не влияют ли новые функции, улучшения и исправленные дефекты на существующую функциональность продукта и не возникают ли старые дефекты.

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

- **Тестирование производительности** – тестирование, которое проводится с целью определения, как быстро работает вычислительная система или её часть под определённой нагрузкой.
- **Нагрузочное тестирование (Load Testing)** – нагрузочное тестирование обычно проводится для того, чтобы оценить поведение приложения под заданной ожидаемой нагрузкой.
- **Стресс-тестирование (Stress Testing)** – стресс-тестирование обычно используется для понимания пределов пропускной способности приложения. Этот тип тестирования проводится для определения надёжности системы во время экстремальных или диспропорциональных нагрузок.
- **Модульные (юнит) тесты против интеграционных тестов** – Разница в том, что интеграционный тест проверяет, что компоненты в вашем приложении правильно работают друг с другом, а модульный тест (unit test) проверяет отдельный компонент в вашем приложении.
- **Pytest** – это основанная на Python среда тестирования, которая используется для написания и выполнения тестовых кодов. В настоящее время службы REST pytest в основном используются для тестирования API, хотя мы можем использовать pytest для написания простых и сложных тестов, то есть мы можем писать коды для тестирования API, базы данных, пользовательского интерфейса и т. д.
- **Continuous Integration** – это практика разработки программного обеспечения, которая заключается в слиянии рабочих копий в общую основную ветвь разработки несколько раз в день и выполнении частых автоматизированных сборок проекта для скорейшего выявления потенциальных дефектов и решения интеграционных проблем. В обычном проекте, где над разными частями системы разработчики трудятся независимо, стадия интеграции является заключительной. Она может непредсказуемо задержать окончание работ. Переход к непрерывной интеграции позволяет снизить трудоёмкость интеграции и сделать её более предсказуемой за счет раннего обнаружения и устранения ошибок и противоречий. Основным преимуществом является сокращение стоимости исправления дефекта, за счёт раннего его выявления. Если вкратце, то это автоматизация запуска тестов.
- **Travis CI** – распределённый веб-сервис для сборки и тестирования программного обеспечения, использующий GitHub в качестве хостинга исходного кода.

Закрепление материала

- Что такое функциональное тестирование?
- Что такое модульное тестирование (unit test)?
- Чем отличается модульное тестирование от интеграционного?
- Что такое Continuous Integration?

Дополнительное задание

Задание

Напишите unit tests для оставшихся непокрытых тестами функций. Также, попробуйте создать интеграционный тест: протестируйте полностью логику получения конкретного фильма, проверьте, что сервер правильно валидирует данные, правильные данные отдает и status_code – OK.

Самостоятельная деятельность учащегося

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

Задание 1
Выучите основные понятия, рассмотренные на уроке.

Задание 2
Изучите материалы из рекомендуемых ресурсов.

Задание 3
Объедините все тесты, которые тестируют похожую логику. Например, все, что связано с фильмом, в специальном классе TestCase, а также попрактикуйте написание тестов с помощью других библиотек: unitests, nose, Hypothesis.

Рекомендуемые ресурсы

Pytest docs:
https://docs.pytest.org/en/stable/

Testing Flask application:
https://flask.palletsprojects.com/en/1.1.x/testing/

Тестирование в Python:
https://habr.com/ru/post/121162/

Юнит-тестирование:
https://habr.com/ru/post/169381/

Continuous Integration для новичков
https://habr.com/ru/post/352282/

Testing, CI/CD
https://www.youtube.com/watch?v=alMRNeRJKUE

Travis CI docs:
https://travis-ci.org/

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com