# Практическая работа №9. «Тестирование интеграции»

Цель: получить практические навыки в применение инструментов анализа качества программных продуктов, а также освоение метода тестирования интеграции.

**Практическое задание**

Цель практического задания: научиться интегрировать данные из стороннего API в локальную базу данных и тестировать корректность передачи данных.

*Необходимые инструменты*

1. Лёгкий API с открытым доступом (JSONPlaceholder).

2. SQLite (как простая база данных).

3. Python и библиотека requests для работы с API.

4. SQL-библиотека для работы с базой (например, sqlite3).

**Вводная часть**

**Интеграционное тестирование -** это процесс проверки взаимодействия и совместной работы различных компонентов или модулей системы для проверки их корректной интеграции и функционирования вместе.

**JSONPlaceholder** — это бесплатный и простой в использовании REST API, предоставляющий тестовые данные, такие как посты, комментарии, пользователи и альбомы. Он используется для обучения и тестирования запросов без необходимости создания собственного API. JSONPlaceholder позволяет работать с различными HTTP-методами (GET, POST, PUT, DELETE) для реализации полного цикла CRUD-операций.

**SQLite** — это встроенная реляционная база данных, которая не требует установки или настройки. Она идеально подходит для небольших проектов, где нужно хранить структурированные данные локально. SQLite легко интегрируется с Python и другими языками программирования, а также поддерживает стандартный SQL.

**Python —** это универсальный язык программирования, который часто используется для работы с API. Библиотека requests предоставляет простой и удобный интерфейс для отправки HTTP-запросов. С её помощью можно выполнять GET, POST, PUT и DELETE запросы, а также обрабатывать ответы от API.

**ЗАДАЧИ**

Задания на оценку «3»

***Задача 1.* Подготовка данных**

1. Подключитесь к API JSONPlaceholder (https://jsonplaceholder.typicode.com/posts) и получите список постов.
2. Создайте базу данных SQLite с таблицей для хранения постов:

Поля: id, title, body.

***Задача 2.* Реализация скрипта для интеграции**

1. Напишите Python-скрипт, который:
   * Делает запрос к API.
   * Получает данные (в формате JSON).
   * Записывает все данные в таблицу SQLite.

***Задача 3.* Тестирование интеграции**

1. Проверьте, что:
   * Все записи из API успешно добавляются в базу данных.
   * Нет дублирования записей при повторном запуске скрипта.
2. Смоделируйте ситуацию, когда API возвращает ошибку (например, замените URL на неправильный).
   * Как скрипт обрабатывает эту ошибку?
   * Выводится ли сообщение об ошибке?

Задания на оценку «4-5»

***Задание № 4.*** Решение ситуационных задач

1. **Вы интегрируете сторонний API для обработки платежей. Внезапно этот API начинает возвращать ошибки 500 (Internal Server Error). Как вы будете анализировать проблему? Какие меры вы предпримете, чтобы минимизировать влияние на ваших пользователей?**
2. **Ваш сервис использует несколько микросервисов для обработки данных. Один из микросервисов перестаёт отвечать из-за большой нагрузки. Как вы оптимизируете взаимодействие между сервисами, чтобы система оставалась стабильной?**
3. **В процессе тестирования вы обнаружили, что один из ваших API вызовов возвращает некорректные данные, если в запросе отсутствует обязательный параметр. Как вы будете исправлять эту проблему, чтобы API оставался предсказуемым для пользователей?**
4. **Во время тестирования интеграции между двумя системами вы заметили, что передаваемые данные периодически теряются. Как вы будете искать причину этой проблемы и какие решения предложите для её устранения?  
   Вы используете брокер сообщений (например, RabbitMQ), и он начинает замедлять обработку задач из-за увеличения объёма данных. Как вы будете определять узкое место и улучшать производительность системы?**
5. **Ваш API начинает возвращать более длительное время ответа, когда пользователь делает запрос с большим объёмом данных. Какие шаги вы предпримете для анализа и оптимизации обработки таких запросов?**
6. **Вы интегрировали сторонний API для загрузки изображений, но время отклика этого API значительно варьируется в зависимости от региона. Как вы будете решать проблему нестабильного времени ответа для глобальных пользователей?**
7. **Вы заметили, что некоторые запросы к API выполняются некорректно, если их отправляют с высокой частотой. Как вы реализуете защиту от подобных ситуаций?**
8. **Во время интеграции стороннего сервиса вы обнаружили, что его документация устарела, и фактическое поведение API не совпадает с описанным. Как вы будете адаптировать свою систему и уточнять детали у разработчиков API?**
9. **Пользователь сообщает, что при определённых условиях запрос к вашему API возвращает пустой ответ. Как вы будете искать причину этой проблемы и фиксировать её?**

**Контрольные вопросы**

Составить 10 тестовых вопросов с вариантами ответов.

**Рефлексия**

1. С какими трудностями столкнулись при выполнении практической работы?
2. Как оцениваете свои знания при выполнении практической работы?
3. Какие среды вы используете в своей практике?
4. Оцените ваш уровень практического применения от 0 до 10

**Домашнее задание**

1) Проанализировать видео ролик на тему: «Интеграционное тестирование» https://www.youtube.com/watch?v=BzDzFrhWEHE Оставьте комментарий под видео

## Форма сдачи:

1. Документ с описанием выполненных заданий и наблюдений.  
2. Скриншоты для подтверждения выполнения каждого задания.  
3. Ответы на вопросы и выводы по каждому заданию.

**Критерии оценивания для заданий 1-3**

| **Задание** | **Критерии оценивания** | **Оценка (1-5)** |
| --- | --- | --- |
| **Задание 1: Подготовка данных** | 1. Успешное подключение к API JSONPlaceholder и получение списка постов. |  |
|  | 2. Создание базы данных SQLite с таблицей, содержащей поля: id, title, body. |  |
| **Задание 2: Реализация скрипта** | 1. Написание Python-скрипта, который делает запрос к API. |  |
|  | 2. Получение данных в формате JSON. |  |
|  | 3. Запись всех данных в таблицу SQLite без дублирования записей. |  |
| **Задание 3: Тестирование интеграции** | 1. Проверка успешного добавления всех записей из API в базу данных. |  |
|  | 2. Проверка отсутствия дублирования записей при повторном запуске скрипта. |  |
|  | 3. Смоделированная ошибка API (неправильный URL) и корректная обработка ошибки скриптом. |  |
|  | 4. Вывод сообщения об ошибке пользователю при возникновении проблемы с API. |  |

**Список литературы**

1. <https://habr.com/ru/articles/736588/> – Как тестировать приложение c API интеграциями (дата обращения: 25.08.2024). – Текст : электронный.
2. https://logrocon.ru/news/intgration\_testing – Интеграционное тестирование: что это? Виды, примеры. (дата обращения: 25.08.2024). – Текст : электронный.
3. https://habr.com/ru/articles/843534/ – Интеграционное тестирование: виды, примеры и инструменты (дата обращения: 25.08.2024). – Текст : электронный.
4. https://www.zaptest.com/ru/%D1%87%D1%82%D0%BE-%D1%82%D0%B0%D0%BA%D0%BE%D0%B5-%D0%B8%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%BE%D0%B5-%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD – Что такое интеграционное тестирование? Глубокое погружение в типы, процесс и реализацию (дата обращения: 25.08.2024). – Текст : электронный.