

Лабораторная работа №4

Тестирование REST-сервисов с использованием SoapUI

Цель:

Тестирование бизнес-процесса получения информации о странах с использованием Fake Store API.

Базовый URL: <https://fakestoreapi.com>

Метод для получения всех товаров: GET /products

Метод для получения товара по ID: GET /products/{id}

Задача бизнес-процесса:

1. Получить список информацию о 10 первых товарах.
2. Проверить, что название (title), цена (price), категория (category), рейтинг (rating.rate, rating.count) присутствуют и имеют валидных тип данных.

Структура сюта-кейса:

TestSuite: Fake Store API

└─ TestCase: Fake Store TestCase

 └─ GetProducts (Request)

 └─ ExtractCountryCodes (Groovy Script)

 └─ LoopController (Groovy Script)

 └─ GetProductDetails (Request)

 └─ ValidateProduct (Groovy Script)

 └─ END (Groovy Script)

Этапы реализации:

1. GetProducts
 - 1.1. GET <https://fakestoreapi.com/products>
 - 1.2. Получает JSON-массив со всеми товарами
2. ExtractProductIds
 - 2.1. Парсит JSON и извлекает ID первых 10 товаров
 - 2.2. Сохраняет их в Custom Property ProductIds

- 2.3. Устанавливает ProductIndex = 0
3. LoopController
 - 3.1. Циклически перебирает ID
 - 3.2. Устанавливает CurrentProductId
 - 3.3. Переходит к следующему шагу, пока не достигнут конец списка
4. GetProductDetails
 - 4.1. GET <https://fakestoreapi.com/products/{CurrentProductId}>
 - 4.2. Получает подробную информацию о продукте
5. ValidateProduct
 - 5.1. Проверяет поля:
 - 5.1.1. id совпадает с запрашиваемым
 - 5.1.2. title не пустое
 - 5.1.3. price — число
 - 5.1.4. category — строка
 - 5.1.5. rating.rate, rating.count — числа
6. END
 - 6.1. Пустой шаг, завершающий TestCase

Результат:

- Проверено 10 товаров
- Успешно пройдены все проверки
- Ошибок в структуре и содержании JSON-ответов не обнаружено

Вывод:

За счёт Groovy скриптов можно реализовать гибкие сценарии с циклами и проверками. Подход можно легко расширить до проверки всех товаров, фильтрации по категории и т.д.

Такой подход особенно удобен для работы с иерархичными данными и вложенными вызовами сервисов.

Выполнил:

студент 2-го курса

Сибилев Антон Игоревич

Файл проекта приложен к отчёту.