**Лабораторная работа №8**

Ссылка на сайт (домен): BASE\_URL = <http://shop.qatl.ru/>

Реализовать API тесты, проверяющие функционал добавления/изменения/удаления товара в магазине. Необходимо учесть максимальное количество ситуаций при работе с товаром. Также нужно позаботиться о том, чтобы после тестирования в базе данных не остались тестовые товары, созданные в процессе тестирования.

Тестовые наборы данных нужно вынести в отдельный файл или атрибуты (зависит от framework). Например, в framework testng https://testng.org/doc/documentation-main.html раздел 3 - testng.xml все входные данные (параметры товара) берутся из xml файла.

Нужно по максимуму использовать Framework (не нужно писать свой). Запуск тестов и чтение тестовых данных в большинстве случаев может делаться стандартными методами, реализованными в Framework.

Для упрощения проверок, так как полей много и проверять нужно почти все поля, можно создать отдельный метод, который осуществит проверок всех полей и переиспользовать его в тесте.

**API:**

*Список всех товаров (GET):* BASE\_URL/api/products

*Удаление (GET):* BASE\_URL/api/deleteproduct?id=ID

*Добавление (POST):* BASE\_URL/api/addproduct

*Редактирование (POST):* BASE\_URL/api/editproduct

Тело запроса для добавления и редактирования:

{

"id": "число",

"category\_id": "число от 1 до 15",

"title": "текст",

"alias": "формируется из поля title через транслит на латиницу. Но если такой алиас существует, то добавляется префикс -0",

"content": "текст",

"price": "число",

"old\_price": "число",

"status": "число (0,1) ",

"keywords": "текст",

"description": "текст",

"hit": "число (0,1) "

}

Условия сдачи лабораторной работы:

1) Исходный код

2) Все входные данные и проверяемые значения должны быть вынесены в конфиг файлы.

3) Подробный и понятный вывод причины падения теста.