Университетская гимназия МГУ

«Роботизированная автоматизация IT процессов в логистике»

ФИО (полностью) учащегося,

Класс, профиль

Руководитель проекта:

Кривошеин Григорий Максимович

Москва 2020

**Введение**

  Разработка нового ПО Pegas 2.0 для бесперебойного функционирования и связи логистических узлов компании требовала постоянного тестирования всех возможных сценариев работы с веб-сервисами. Многообразие подобных сценариев делает невозможным их ежедневное ручное тестирование, что создает необходимость написания специального программного обеспечения, предназначенного для автоматизированная тестирования веб-сервисов.

Актуальность проблемы заключается в том, что Pegas 2.0 в данный момент находится в процессе активной разработки, что создает необходимость постоянного тестирования отдельных динамически изменяющихся веб-сервисов.

Значимость проблемы состоит в том, что система автоматизированного тестирования позволит в короткие периоды времени находить ошибки и уязвимости веб-сервиса Pegas 2.0, минимизируя потенциальный ущерб Pony Express.

**Анализ текущего состояния области исследования**

Команда тестировщиков Pony Express на момент начала деятельности нашей проектной группы уже успела покрыть часть сервисов Pegas 2.0 автоматизированными тестами. В нашем распоряжении оказалась еще не покрытая часть сервисов программного обеспечения: сервис географии и логистических центров, сервис регистрации событий “71. Прибыл на склад (без сортировки)”, сервис регистрации событий “79. Включен в консолидацию”, сервис по работе с группами пользователей (курьерами, сотрудниками складов и т. д.).

**Применяемые методы**

Всё созданное нами программное обеспечение, предназначенное для автоматизированного тестирования веб-сервисов, было написано с использованием стандартных библиотек языка программирования Python, а также нескольких дополнительных библиотек, предназначенных для работы с back-end (selenium) и front-end (requests и json) функционалом. Кроме того, само тестирование осуществлялось при помощи библиотеки pytest и дополнительного программного обеспечения Allure Framework.

Выбор языка программирования был определен, во-первых, целью внедрения написанных авто-тестировщиков в функционирующую сеть автоматизированная тестирования Pony Express, написанного при помощи языка программирования Python, а во-вторых, его многогранным функционалом при работе с веб-сервисами.

Библиотека Selenium Webdriver использовалась нами для тестирования front-end части Pegas 2.0, так как ее непосредственное предназначение - моделирование поведения пользователя веб-клиента, автоматизация рутинных действий в веб-браузере (нажатие на кликабельные объекты, ввод текста, перемещение между вкладками и т. п.). Selenium Webdriver широко применяется для тестирования пользовательского интерфейса сайтов.

Библиотека Requests была предназначена для создания авто-тестировщиков back-end части Pegas 2.0. Её методы позволяют отправлять запросы напрямую к сервисам Pegas 2.0 через специальное документированное api и получать протоколы результатов запросов в ответ, которые при помощи библиотеки Json можно конвертировать в формат списков, удобных для последующей обработки в Python.

Функционал библиотеки Assert предназначен для упрощения тестирования и связи его результатов с Allure Framework. Assert позволяет проводить сравнения некоторых данных и получать сообщения об ошибках. Allure Framework, получая на вход данные о проведенных тестах и полученных ошибках, переводит результаты в графический формат - графики и диаграммы, собирает вместе все ошибки и позволяет поддерживать статистику работы веб-сервиса на протяжении долгого времени.

Помимо всего перечисленного мы изучили и в процессе нашей работы активно использовали Git - распределенную систему управления версиями. При помощи данного сервиса мы могли координировать наши действия, а также обмениваться материалами.

**Результаты работы**

Результатом нашей работы являются написанное и внедренное во внутренние структуры Pony Express программное обеспечение, предназначенное для автоматизированного тестирования веб-сервисов. Нами было суммарно написано 34 программы-тестировщика: 14 для тестирования front-end сервиса при помощи библиотеки Selenium Webdriver и 20 для тестирования back-end сервиса при помощи библиотеки Requests.

**Выводы**

В ходе работы над проектом наша команда изучила язык программирования Python, а также библиотеки Requests, Selenium Webdriver, Json и программное обеспечение Allure Framework. Кроме того, для координации действий мы освоили систему Git. Практическим результатом нашей деятельности стали написанные авто-тестировщики, которые были проверены и внедрены в структуру тестирования программного обеспечения Pegas 2.0.

**Список литературы**

1). Python 3.8 Documentation - https://docs.python.org/3.8/

2). Requests Documentation - https://requests.readthedocs.io/en/master/

3). Selenium Documentation - https://www.selenium.dev/documentation/en/

4). JSON Documentation - https://www.json.org/json-en.html

5). Git Documentation - https://git-scm.com/doc