

АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ WEB-ПРИЛОЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ SELENIUM WEBDRIVER

Лось Н. А., Ярошенко А. Л., Алексеев В. Ф.

Научный руководитель: канд. техн. наук, доц. Алексеев В. Ф.

Белорусский государственный университет информатики и радиоэлектроники, Беларусь

E-mail: losnikita1995@gmail.com

Аннотация — Рассмотрены принципы построения архитектуры программного инструмента для автоматизированного тестирования web-приложений.

1. Введение

Web-приложения обладают сложной функциональной частью, что в значительной степени увеличивает процент ошибок в программном коде, а также время, затрачиваемое на тестирование. В условиях ограниченных человеческими, временными и финансовыми ресурсами необходима автоматизация этих процессов. Многочисленные изменения в web-приложении при разработке осложняют процесс автоматизации тестирования. Для решения этой проблемы необходимо создание инструмента тестирования с архитектурой, отвечающей требованиям масштабируемости и гибкости.

В работе рассматриваются принципы построения архитектуры инструмента автоматизированного тестирования на базе драйвера web-браузера Selenium WebDriver.

2. Основная часть

Selenium WebDriver, или просто WebDriver — это драйвер браузера, а также не имеющая пользовательского интерфейса программная библиотека. Этот инструмент напрямую вызывает команды браузера, используя его встроенный программный интерфейс (API). Назначение WebDriver состоит в предоставлении объектно-ориентированного API, который позволяет разрабатывать программы, управляющие поведением браузеров. WebDriver содержит драйвера всех распространённых браузеров. Он хранит библиотеки для обращения к API на разных языках программирования, а также является кроссплатформенным средством и поддерживает Microsoft Windows, Apple OS X, Linux [1].

Для обеспечения эффективной автоматизации кроссбраузерного тестирования архитектура программного средства должна быть построена так, чтобы логика создания объекта драйвера конкретного браузера была вынесена в отдельный класс. Так как все классы WebDriver, представляющие драйверы браузеров, имеют единый интерфейс, то в классах-тестах вызываются методы интерфейса, а не конкретной его реализации. Такой подход сведёт затраты на внесение изменений при тестировании web-приложения с использованием различных браузеров до минимума. Это возможно благодаря принципу полиморфизма, реализуемому в WebDriver.

Одной из главных проблем авто-тестов является их поддержка. Любые незначительные изменения в пользовательском интерфейсе, влекут за собой изменения в тесте. Page Object это шаблон, предназначенный для улучшения поддержки тестов и сокращения дублирования программного кода. Данный шаблон позволяет разделять логику выполнения тестов от их реализации. Page Object является классом, который моделирует web-страницы тестируемого web-приложения в качестве объектов в коде. Классы-тесты используют методы этих объектов

каждый раз, когда они должны взаимодействовать с пользовательским интерфейсом web-страницы. Преимущество шаблона заключается в том, что при изменении пользовательского интерфейса web-страницы классы-тесты не изменяются, а меняется только описание класса соответствующей web-страницы. Следовательно, все изменения, необходимые для поддержки нового интерфейса, инкапсулированы в одном месте [2, 3].

Для обращения к HTML-элементам web-приложения WebDriver использует локаторы элементов. Оптимальный способ хранения локаторов также важен для архитектуры программного средства. UI Map это механизм, который хранит все локаторы элементов в одном месте для оптимизации процесса изменения путей к элементам. Для реализации этого механизма необходимо создание properties файла, содержащего все локаторы в виде ключ-значение, что позволит инкапсулировать их в одном месте. UI Map будет использоваться автоматизированными тестами для определения локатора элемента.

3. Заключение

Таким образом основными элементами архитектуры программного инструмента автоматизированного тестирования на базе драйвера web-браузера Selenium WebDriver являются: web-страницы приложения, представленные объектами Page Object; классы-тесты, которые содержат логику тестового случая и обращаются к классам страниц; UI Map для инкапсуляции локаторов элементов; класс с логикой создания драйвера для конкретного браузера. Вышеперечисленные элементы должны быть спроектированы согласно принципам инкапсуляции и полиморфизма.

Построение такой архитектуры позволит создать инструмент, способный произвести эффективную автоматизацию тестирования web-приложения.

4. Список литературы

- [1] Test Design Considerations [Электронный ресурс]. — Режим доступа: <http://www.seleniumhq.org/docs>.
- [2] WebDriver. Обзор и принцип работы [Электронный ресурс] / Selenium WebDriver. — Режим доступа: <https://kreisfahrer.gitbooks.io/selenium-webdriver/content>.
- [3] Selenium и Page Object паттерн [Электронный ресурс]. — Режим доступа: <http://internetka.in.ua/selenium-page-object>.

AUTOMATED TESTING OF WEB APPLICATIONS USING SELENIUM WEBDRIVER

Los N. A., Yarashenka A. L., Alexeev V. F.

Scientific adviser: Alexeev V. F.

Belarusian State University of Informatics and
Radioelectronics, Belarus

Abstract — The principles of creation of architecture of the program instrument for the automated testing of web-applications is considered.