РАЗРАБОТКА ФРЕЙМВОРКА ДЛЯ АВТОМАТИЗИРОВАННОГО ТЕСТИРОВАНИЯ ВЕБ-ПРИЛОЖЕНИЙ

Андриевский П. В.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Осипович В. С. – к. т. н., доц. каф. ИПиЭ

В статье рассмотрены основные компоненты разработки фреймворка для тестирования веб-приложений. Продемонстрирован пример генерации отчета по результатам выполнения тестирования.

Цель разработки – создание и внедрение программного модуля, автоматизирующего процесс тестирования веб-приложений.

Программный модуль для автоматизированного тестирования веб-приложений реализуется в виде фреймворка, который обеспечивает снижение трудоемкости, временных затрат тестирования, а также исключение человеческого фактора. Предусмотрены возможности организации автоматизированного тестирования пользовательского интерфейса и API, генерация и предоставление удобочитаемых отчётов о результатах тестов.

Для создания и поддержки фреймворка необходимо реализовать взаимодействие драйвера браузера и компонентов на определенных страницах веб-приложения. Для достижения этой цели используется шаблон проектирования Page Object. В качестве драйвера браузера используется Slenium WebDriver – инструмент для проведения тестирования веб-приложений в различных браузерах и на различных платформах.

Программное обеспечение представляет собой код, написанный на языке программирования Java. Maven – фреймворк для автоматизации сборки проектов, специфицированных на XML-языке POM. Так же используется программный инструмент Cucumber, реализующий подход BDD. BDD – это методология разработки программного обеспечения, являющаяся ответвлением от методологии разработки через тестирование (TDD). Неотъемлемой составляющей использования Cucumber является инструмент Gherkin, который используется для описания поведения систем. Спецификации, созданные на нём, одновременно являются и проектной документацией, и автотестами. В качестве модуля для создания отчётов используется плагин инструмента Cucumber. Результаты тестирования оформляются в интуитивно понятный html-отчёт. Пример отчёта представлен на рисунке 1.

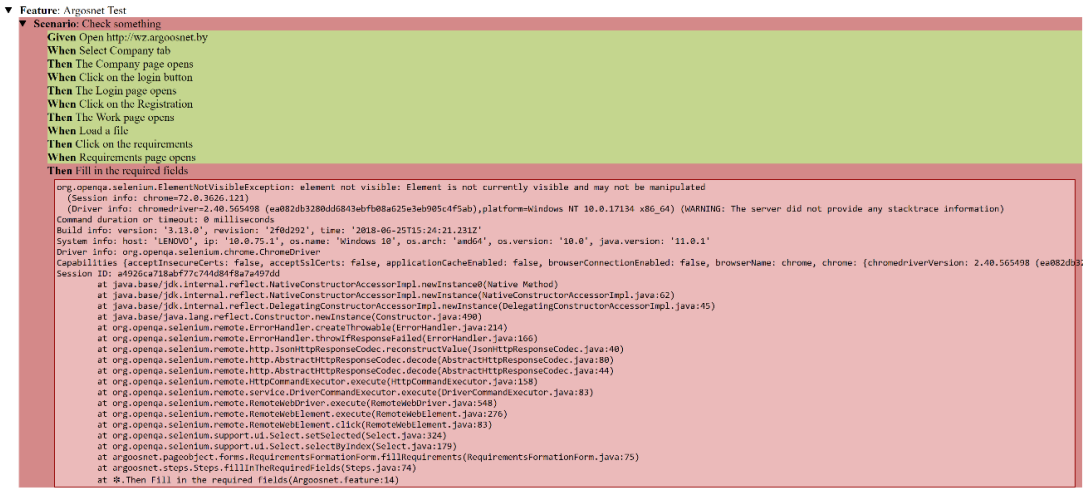

Рисунок 1 – Отчёт, сгенерированный с помощью плагина инструмента Cucumber

Для логирования сообщений использовалась библиотека Log4j. Логирование – это возможность следить за процессом выполнения бизнес-логики проекта.

Для организации непрерывной интеграции использовалась система Jenkins, поддерживающая используемую систему управления версиями Git. Сборка проекта может быть запущена разными способами, например, по событию фиксации изменений в системе управления версиями, по расписанию, по запросу на определенный URL или после завершения другой сборки в очереди.

Список использованных источников:

1. Савин Р. "Тестирование Дот Ком, или Пособие по жестокому обращению с багами в интернет-стартапах." — М.: Дело, 2007. — 312 с.

2. Куликов Святослав. Тестирование программного обеспечения. — Базовый курс, EPAM Systems, 2018 — 325 с.