**SWD Page Recorder: Записывает PageObject-классы для Selenium WebDriver**

Уверен, что бывалые автоматизаторы из мира веб-тестирования на Selenium WebDriver, уже успели попробовать на вкус все достоинства использования паттерна PageObject в своем ремесле. Да, это и сокращение времени на обслуживание кода автоматизации, и более читабельный простой и надежный код. К сожалению, всё еще тратится слишком много рутинных усилий на создание новых PageObject-классов и поддержку существующих.

К счастью, есть инструменты, которые помогут сократить этот рутинный «ручной труд» автоматизатора.

SWD Page Recorder – это попытка создать один из таких инструментов.

КАТ

**Какую проблему решает SWD Page Recorder?**

Самая большая проблема – это отсутствие средств быстрой декларации элементов реальной веб-страницы. SWD Page Recorder не напишет за вас интеллектуальный PageObject, – зато позволит значительно сократит время в процессе поиска элемента страницы и его добавление в код фреймворка автоматизации тестирования. SWD Page Recorder позволяет не только найти необходимый локатор, но и оптимизировать его (сократить размер) и сгенерировать весь необходимый код для последующей вставки декларации элемента в код тест-фреймворка.

**Как эта проблема уже решается**

Вместо того чтобы внести нечто кардинально инновационное, миссия SWD Page Recorder состоит в том, чтобы оптимизировать и ускорить те подходы, которые применяются уже сегодня. Например, самым известным подходом в поиске и отладке необходимого локатора веб-элемента есть использование Инструментов Разработчика, которые есть в каждом браузере. Проблема в том, что для разных браузеров – инструментарий тоже разный. Например, для Firefox вы можете использовать FireBug + FirePath, для Google Chrome – команды консоли или прямые вызовы через JavaScript, для 10-й версии Internet Explorer … в общем, тут все как-то вообще плачевно: CSS-селекторы можно отлаживать через строку поиска, а XPath – вообще никак стандартными средствами.

SWD Page Recorder предоставляет единый инструментарий отладки локаторов для всех браузеров, с которыми умеет работать Selenium WebDriver. И тут стоит подчеркнуть, что Page Recorder тестирует всё через сам WebDriver – так что у вас не будет такой проблемы, когда локатор, найденный другим путем – может не работать в WebDriver.

**Основные возможности приложения**

**Исследователь веб-элементов (In-Browser Web Element Explorer)**

После запуска по кнопке «Start» из SWD Page Recorder – внедряет специальные скрипты в запущенный веб-браузер. Используя **Ctrl + указатель мыши**, можно подсветить искомый элемент на странице веб-браузера. При **нажатии Ctrl + Правая кнопка** – появится форма, которая предложит добавить найденный элемент в приложение. После заполнения поля имени и нажатии на кнопку “Add Element” – найденный элемент сохранится в дереве PageObject в SWD Page Recorder

**Генерация кода PageObject-класса на C#, Java, Perl, Python, Ruby**

По завершению работы над добавлением элементов в PageObject, на вкладке “Source Code” можно сгенерировать PageObject-класс на интересующим вас языке программирования.

Стоит отметить, что шаблоны PageObject-классов на разных языках программирования реализованы посредством шаблонизатора Razor – самого популярного среди программистов ASP.NET MVC.

Шаблоны позволяют использовать вставки C#-кода внутри. А для того, чтобы добавить новый шаблон – достаточно создать новый файл \*.cshtml в папке «CodeTemplates». Первым шаблоном в списке (по умолчанию) будет показываться файл со строкой «(Default)» в имени.

Шаблонный движок позволяет добавлять новые шаблоны без перекомпиляции приложения, но, для обновления списка, приложение необходимо перезапустить.

Оговорюсь сразу, что составление и отладка шаблонов – это не такое уже простое занятие в данной версии, но вы всегда можете попросить помощи у меня.

**Сохранение структуры PageObject**

Пока работа над конструированием PageObject-класса не завершена, приложение позволяет сохранить наработки в “\*.pox” файл, который будет находиться в папке с SwdPageRecorder.exe. Это стандартный XML-файл, полученный в результате XML-сериализации. Сохранив изменения, в любой момент вы можете закрыть приложение и восстановить работу позже.

Открыть папку с \*.pox файлами можно по кнопке “Files”. Удалить файл из списка существующих, можно обычным удалением или переносом файла.

**Отладка и поиск локаторов**

По двойному клику по элементу дерева PageObject, можно открыть локатор веб-элемента на редактирование во вкладке “Locators”.

Верхняя часть формы позволяет отредактировать локатор и задать необходимый метод поиска (Html ID, CSS, XPath и др. ).

После редактирования локатора, не забудьте нажать кнопку «Update» – и локатор будет обновлен во внутренней структуре PageObject-класса.

Нижняя часть служит для поиска локаторов. При нажатии на кнопку «Test» – все найденные веб-элементы будут отображены на вкладке «WebElements» справа. Двойной клик по найденному элементу – отобразит его в DOM дереве страницы справа. Двойной клик по элементу DOM-дерева – подсветит элемент в браузере.

**Вариант Использования: Создание PageObject класса для страницы регистрации Хабрахабр**

Необходимые условия: у вас установлен браузер Firefox. Для использования другого браузера – вам будет необходимо *скачать драйвер для нужного браузера(*[*https://code.google.com/p/selenium/downloads/list*](https://code.google.com/p/selenium/downloads/list)*)* и сохранить его рядом с “ SwdPageRecorder.exe”

**Запуск браузера**

На вкладке “Browser Settings” оставьте “FireFox” по умолчанию либо выберете другой браузер и нажмите кнопку «Start»

Дождитесь запуска браузера (кнопка “Start” поменяет заголовок на «Stop»)

Зайдите на страницу «https://auth.habrahabr.ru/register/» в открытом браузере

**Запуск исследователя веб-элементов в браузере**

На вкладе «Locators» в приложении, нажмите на кнопку «Start» в группе «Web Element Explorer»

Переключитесь на браузер

Зажав **Ctrl** и проводя мышкой, убедитесь в том, что элементы подсвечиваются (цвет фона станет серым)

Нажав **Ctrl+Правая кнопка** на элементе – вы вызовите форму добавления элемента. Придумайте и введите уникальное имя элемента в поле «Code identifier» и нажмите кнопку «Add Element»

**Редактирование найденных элементов в приложении**

Переключитесь на приложение и обратите внимание, что новый элемент добавился в дерево PageObject справа.

В поле, ниже надписи “Page Object File” – придумайте и введите имя файла для страницы.

Нажмите кнопку Save для сохранения

**Генерация кода**

Перейдите на вкладку «Source Code»

Нажмите на кнопку “Generate” для генерации кода PageObject-класса.

Сгенерированный код или его часть можно вставить в код тестового фреймворка, либо продолжить редактирование/добавление элементов в PageObject

**Качество продукта**

Данная версия – это попытка создать минимально жизнеспособный продукт (http://en.wikipedia.org/wiki/Minimum\_viable\_product), который с одной стороны, поможет вам ускорить и улучшить работу, а мне – собрать ценную обратную связь, которая поможет улучшить продукт в дальнейшем.

Следовательно, наличие недоработок в багов в нестандартных сценариях может быть не редким.

Отзывы об SWD Page Recorder, вы можете оставить как и в этом посте на Хабрахабр, на Github, в любом анонсе на моем блоге так и присылать по email (который также можно найти на моем блоге).

**Дополнительные ссылки:**

Все анонсы проекта в блоге:

<http://blog.zhariy.com/search/label/swd>

Моя презентация о вариантах использования PageObject (с видео и примерами кода; C#/.NET)

<http://blog.zhariy.com/2013/02/atdays-pageobject.html>

Страница проекта SWD Page Recorder на Github (на данный момент в процессе создания)

<http://dzhariy.github.io/swd-recorder/>