SWD Page Recorder BETA1 – записывает PageObject’ы на C#, Java, Ruby, Perl, Python!

Коллеги Автоматизаторы,

Если вы уже ощутили всю красоту и важность использования паттерна PageObject в своем проекте автоматизации на Selenium WebDriver, то, наверняка вы уже успели оценить какое громадное количество усилий тратится на создание новый и поддержку существующих классов PageObject.

Постоянно меняющиеся локаторы, дивы вылезающие из спанов, поля ввода, которые раз в неделю меняют свое имя… с этим, конечно-же можно жить. Это можно терпеть… К этому можно привыкнуть. Но, зачем? Почему-бы не свести долю рутинной работы по поиску локаторов, ну ладно, пусть не к нулю – но к минимуму.

Именно это позволит вам сделать инструмент SWD Page Recorder. Хорошо… пусть сейчас и не на 100%. А, впрочем, все зависит от вашего фидбека, потому что свои задачи на работе я решаю эффективно. «SWD Page Recorder» помог мне сократить «ручную автоматизацию» в разы. И я всем очень доволен. И вы тоже можете получить это удовольствие от работы приложения. Более детально – в самом низу.

Видео: основные варианты использования SWD Page Recorder для записи PageObjects

**Что нового?**

**Улучшен поиск элементов на странице (WebElement Explorer)**

Эта фича была еще в альфа версии проекта, но сейчас, наконец-то она работает стабильно!

WebElement Explorer, после запуска, позволяет найти элемент в браузере при помощи указателя мышки и зажатой клавиши Ctrl. После нажатия Ctrl+RightClick – в раузере появляется форма, через которую найденный элемент можно добавить в приложение.

Для найденного элемента будет сгенерирован уникальный XPath-локатор, который может быть модифицирован и протестирован уже в приложении.

**Шаблоны PageObjects на C#, Java, Ruby, Perl, Python… да хоть на Хаскеле!**

Для генерации кода SWD Page Recorder использует шаблонный движок Razor, который позволяет… хм… писать на C#, как это делают на PHP! Да, в папке «CodeTemplates» – вы найдете исходный код шаблонов. Да, вы можете его модифицировать без компиляции и других замуток.

Да, вы можете создать свой шаблон – просто добавьте новый файл. Да, если вам будет необходима помощь в составлении шаблона – обратитесь ко мне.

**Свойства WebElement**

Внизу дерева PageObject добавились свойства элемента. Это удобно, когда в процессе работы над одним элементом – можно скопи-пастить локатор другого, уже сохраненного элемента.

**Browser Capabilities**

На вкладке запуска браузера появилось окно установок для браузера (Capabilities). Пока-что можно использовать как справочную информацию, но вскоре настройки можно будет и менять.

**Поддержка Linux (в процессе)**

Мне удалось запустить приложение под Linux Mint + Mono. К сожалению, некоторые фичи работают очень нестабильно. Но, это только начало. И, по крайней мере, приложение удалось скомпилировать, успешно запустить и присоединится к WebDriver-серверу.

**Как вы можете помочь?**

А я не настаиваю. Но, проект можно использовать уже сейчас. И если этот проект принесет вам пользу, то почему бы не помочь в его разработке?

Как?

1. Тестирование. Я не уверен, что все отлично работает везде. Если вы столкнулись с серьезной проблемой – напишите о ней любым удобным для вас способом: на Гитхабе, тут в комментариях, мне на почту.
2. Предложения об улучшении. Если вам нужна новая фича – напишите и мы посмотрим, насколько просто будет ее добавить.

Планы на развитие проекта.

А вот тут: