

[EN](#) [RU](#)[С чего начать?](#) [Док](#) [ЧАВО](#) [Блог](#) [Javadoc](#) [Пользователи](#) [Отзывы](#)

Документация

Page Objects

Page Objects - это просто!

В мире тестирования очень популярен шаблон **Page Objects**. Суть его в том, что для каждой страницы тестируемого приложения создаётся отдельный объект, методы которого инкапсулируют логику работы с отдельными элементами. Считается, что **Page Object** позволяет избежать дублирования локаторов в тестах.

А **Selenide** позволяет писать более короткие и читаемые **Page Objects**.

Вот как выглядит **Page Object** с **Selenide**.

```
public class GoogleSearchPage {  
    public GoogleResultsPage search(String query) {  
        $(By.name("q")).setValue(query).pressEnter();  
        return page(GoogleResultsPage.class);  
    }  
}
```

```
}

public class GoogleResultsPage {
    public ElementsCollection results() {
        return $$("#ires li.g");
    }
}
```

Как видите, никаких `PageFactory`, никаких `initElements` и прочего мусора. Ваш `Page Object` содержит только вашу логику.

А как же выглядит тест с использованием этих `Page Objects`? А вот так:

```
GoogleSearchPage searchPage = open("/login", GoogleSearchPage.class);
GoogleResultsPage resultsPage = searchPage.search("selenide");
resultsPage.results().shouldHave(size(10));
resultsPage.results().get(0).shouldHave(text("Selenide: Concise UI Tests in Java"));
```

Кто скажет, что это не просто?

Классический вариант Page Object

Многим полюбился шаблон написания тестов, согласно которому на каждый веб-элемент создаётся поле в `page object`. У этого подхода есть недостатки, но `Selenide` позволяет делать и так:

```
public class GoogleSearchPage {
    @FindBy(how = How.NAME, using = "q")
    private SelenideElement searchBox;

    public GoogleResultsPage search(String query) {
        searchBox.setValue(query).pressEnter();
        return page(GoogleResultsPage.class);
    }
}

public class GoogleResultsPage {
    @FindBy(how = How.CSS, using = "#ires li.g")
    public ElementsCollection results;
}
```

И напоследок про ООП

Хочу напомнить, что изначальный смысл **Page Objects** состоял в том, чтобы инкапсулировать (то есть прятать!) логику работы с элементами. Тесты не должны ничего знать о веб-элементах, не должны оперировать напрямую с **XPath** или другими селекторами. Тесты должны использовать публичные методы пэдж объекта.

То есть если уж вы объявляете поля для элементов, то пусть они будут приватными, а все операции с ними пусть осуществляются через публичные методы.

Иначе зачем весь этот ООП?

76 Comments Selenide  Disqus' Privacy Policy

 Login ▾

 Recommend 9

 Tweet

 Share

Sort by Best ▾



Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS 

Name



Valerii Synenko • 3 years ago

Пытаюсь заюзать селенидовские "плюшки" в своем фреймворке который написан на чистом селениуме и получаю вот такое:

```
Error:(59, 24) java: cannot access org.openqa.selenium.WrapsDriver
class file for org.openqa.selenium.WrapsDriver not found
```

Помогите/поясните пожалуйста)

3 ^ | ▾ 2 • Reply • Share ›



Yuri Banny • a month ago

А как для Kotlin будет выглядеть эта строчка?

```
GoogleSearchPage searchPage = open("/login", GoogleSearchPage.class);
```

^ | ▾ • Reply • Share ›



Andrei Solntsev Mod → Yuri Banny • a month ago

А в чём вопрос-то? Если вы взяли писать тесты на котлине, но наверное, должны знать его синтаксис?

Как-то так:

```
val searchPage = open("/login", GoogleSearchPage::class.java);
```

^ | v • Reply • Share ›



Mike Golovan • 10 months ago

чем отличается `return page(GoogleResultsPage.class)` от `return this` ?

^ | v • Reply • Share ›



Andrei Solntsev Mod → Mike Golovan • 10 months ago

Да почти ничем. `return this` возвращает экземпляр текущей страницы, с `page(foo.class)` возвращает новый экземпляр данной страницы.

^ | v • Reply • Share ›



Андрей • 3 years ago

@Andrei Solntsev Добрый день, можете подробнее объяснить пример с Page Object, что происходит когда выполняется `GoogleSearchPage searchPage = open("/login", GoogleSearchPage.class);`? какая страница открывается?

^ | v • Reply • Share ›



Andrei Solntsev Mod → Андрей • 3 years ago

Метод `open("/login")` по умолчанию открывает страницу `http://localhost:8080/login`. Это потому, что по умолчанию `Configuration.baseUrl = "http://localhost:8080"`.

Можете при желании эту настройку переопределить. Либо вызывать метод `open` с абсолютным урлом: `open("https://google.com/login");`

^ | v • Reply • Share ›



Дима Костичев • 3 years ago • edited

Привет, есть вопрос, пытаюсь реализовать поддержку блоков внутри страницы(вложенные страницы). Объявляю поле страницы `List<custompage>`, но сталкиваюсь с такой проблемой, как выкидывание ошибки при инициализации page objects модели в `SelenidePageFactory.proxyFields`. Селенид пытается найти элемент до загрузки страницы, это работает только для списка кастомного класса, с `List<selenideelement>` table и CustomPage block все в порядке.

^ | v • Reply • Share ›



Andrei Solntsev Mod → Дима Костичев • 23 days ago

@disqus_liougsUcmq

Если не ошибаюсь, эта ошибка была исправлена в декабре 2020 в Selenium 5.17.0

См. <https://ru.selenium.org/202...>

^ | v • Reply • Share ›



Dmitry Zimin • 3 years ago

Привет всем! Такой вопрос: допустим есть page на которой есть возможность создания юзера. Но форма для создания юзера имеет много полей, порядка 20. Если придерживаться подхода Page Objects я должен создать метод который создает этого юзера, но тогда получается что в этом методе будет оч много кода 20 строк, что не совсем правильно с точки зрения написания кода, метод становится большим и трудно читаемым. Посоветуйте как быть в такой ситуации.

^ | v • Reply • Share ›



Andrei Solntsev Mod → Dmitry Zimin • 3 years ago

Ничего страшного, можно разбить этот большой метод на несколько маленьких частных методов.

^ | v • Reply • Share ›



Dmitry Zimin → Andrei Solntsev • 3 years ago

Ну я так и делал), спасибо

^ | v • Reply • Share ›



CaBocuk • 4 years ago • edited

@Andrei Solntsev, здравствуйте.

Подскажите, пожалуйста решение. Предположим, я автоматизирую ютуб. На главной странице есть список карточек видео, где каждая карточка содержит картинку, ссылку на канал и прочие элементы, которые есть и у остальных карточек.

Логично было бы сделать класс VideoCard extends ElementsContainer, который бы уже содержал SeleniumElement'ы вроде картинки, названия, названия канала итд.

Но мне немного непонятно, как с ним работать дальше на странице. Я бы хотел получить список этих карточек на странице и уже работать с ними как с коллекцией каких-то объектов.

ElementsCollection заставить работать с кастомными элементами нельзя, насколько я понимаю.

Например в HtmlElements я бы определил для VideoCard определенный локатор, по которому бы фреймворк искал, и потом на пейдже объявил бы List<videocard> cards.

Фреймворк бы уже при инициализации пейджи сам все сделал бы за меня, а я б смог работать с каждой карточкой в отдельности.

Есть ли аналог такой функциональности в Selenide?

Заранне спасибо за ответ и извините за, возможно, сумбурное объяснение :)

^ | v · Reply · Share ›



CaBocuk → CaBocuk · 4 years ago

Насколько я смог нагуглить, решение, где VideoCard extends ElementsContainer и у VideoCard есть аннотация @FindBy(...) замечательно заставит работать List<videocard> на странице, и все замечательно заработает. Пока, к сожалению нет возможности проверить, но если не прав - поправьте)

Но возник новый вопрос - с точки зрения идеологии фреймворка, как относитесь к объединению ElementsContainer и SelenideElement?) В качестве реализаций методов SelenideElement просто будет getSelf().doSmth()

Просто интересно, почему вы решили не делать так из коробки? Вроде удобнее, не надо везде делать прослойку в виде getSelf(). Или я не вижу каких-то очевидных подводных камней?)

^ | v · Reply · Share ›



Andrei Solntsev Mod → CaBocuk · 4 years ago

Привет!

С ходу немножко сложно во всё это вникнуть, но да, это должно работать.

Да, наверное, можно было бы объединить ElementsContainer и SelenideElement.

Хотя погоди... вообще-то обычно ElementsContainer - это ведь, допустим, DIV. Так? Он просто содержит другие элементы, и ему не нужны свои методы. Зачем понадобилось, чтобы он наследовал ещё и SelenideElement?

1 ^ | v · Reply · Share ›



CaBocuk → Andrei Solntsev · 4 years ago · edited

Привет. Спасибо за ответ) Возможно я немного ошибаюсь идеологически, но...

Если рассматривать конкретно мой случай - я описывал рорир с парой элементов и полей внутри. И попап, хоть и является DIV, но пару вариантов работы с ним как с обычным элементом все же есть.

Например, он может быть visible, а может быть не visible. У него есть bg-color. У него есть ширина и высота. И самое важное, относительно него можно искать элементы) Ну и по моей личной логике, попап является полноценным элементом, с которым, в принципе, можно делать много из того, что и с какой-нибудь кнопкой.

И мы получаем варианты - с объединением VS без объединения

1) `popup.shouldBe(visible)` VS `popup.getSelf().shouldBe(visible)`

2) `popup.shouldHave(attribute("width", 500))` VS
`popup.getSelf().shouldHave(attribute("width", 500))`

и им подобные.

3) `popup.$('#submit-btn')` VS `popup.getSelf().$('#submit-btn')`

И получается, что при позиционировании продукта, как "тесты с супер

[see more](#)

^ | v • Reply • Share ›



Andrei Solntsev Mod → CaBocuk • 4 years ago

Да нет, вроде всё логично звучит.

Конечно, `getSelf()` - не айс. Просто понять не могу, почему никто на это до сих пор не жаловался.

Кстати, а как реализован метод `getSelf()`?

^ | v • Reply • Share ›



CaBocuk → Andrei Solntsev • 4 years ago • edited

Он просто возвращает `self` :)

Где `self` - простой `SelenideElement`, который, как я вангую, инициализируется при вызове `page(...)` по данным из аннотации `@FindBy`, которую повесили на класс или объявление переменной `ElementsContainer`'а

То есть в любом случае такое понятие как "элемент контейнера" есть, но доступ к нему мы получаем через обязательный дополнительный вызов `getSelf()`

^ | v • Reply • Share ›



Andrei Solntsev Mod → CaBocuk • 4 years ago

Я не понял, как именно инициализируется `self`. Можешь кусок кода кинуть?

^ | v • Reply • Share ›



CaBocuk → Andrei Solntsev • 4 years ago • edited

Так это ж делает ваш же `SelenideFieldDecorator` :)

```
private ElementsContainer createElementsContainer(By selector, Field field) {
    try {
```

```

SelenideElement self = ElementFinder.wrap(searchContext, selector, 0);
return initElementsContainer(field.getType(), self);
} catch (Exception e) {
throw new RuntimeException("Failed to create elements container for field " +
field.getName(), e);
}
}
}

```

^ | v • Reply • Share ›



Andrei Solntsev Mod → CaBocuk • 4 years ago

А, точно. Я и забыл. :)
Давно это было. С тех пор я сам ElementsContainer почему-то не использовал.

^ | v • Reply • Share ›



CaBocuk → Andrei Solntsev • 4 years ago

Так все-таки реализацию фичи рассматриваете?)

^ | v • Reply • Share ›



Andrei Solntsev Mod → CaBocuk • 4 years ago

Да, конечно. Почему бы нет.

1 ^ | v • Reply • Share ›



CaBocuk → Andrei Solntsev • 4 years ago

е-е-е, я не бесполезный)

^ | v • Reply • Share ›



iherb iherb • 4 years ago

А что что означает instance?

<https://preply.com/question...>

jj

^ | v • Reply • Share ›



AlexM • 4 years ago

И это опять я. При использовании SelenideElement тест не работает с ошибкой:
java.lang.IllegalArgumentException: Can not set com.codeborne.selenium.SelenideElement field
SomeClass.Some2ndClass*.searchBox to com.sun.proxy.\$Proxy5

Помогите, пожалуйста!

^ | ▾ · Reply · Share ›



AlexM → AlexM · 4 years ago

И я снова разобрался сам :D Спасибо еще раз!

^ | ▾ · Reply · Share ›



AlexM · 4 years ago

Здравствуйте, Андрей!

Подскажите, пожалуйста, можно ли, используя ПО, обращаться к методам Селенида? Немного запутанно написал, так что вот пример:

```
@FindBy(how = How.NAME, using = "name")
@CacheLookup
private WebElement nameBox;
```

а потом

```
public void nameText(){
    nameBox.shouldHave(text("name"));
}
```

?

То есть, я ищю локатор с помощью @FindBy, а не \$("locator"). Idea не определяет это, как кол к какому-то методу Селенид.

А мне очень не хочется использовать Селениум + дописывать ожидания. Уже привык к комфорту Селенида)

Спасибо за ответы!

^ | ▾ · Reply · Share ›



AlexM → AlexM · 4 years ago

Или с Селенидом лучше не использовать @FindBy вообще? А просто создать класс, в котором я буду создавать методы со всеми поисками\действиями, а в главном классе просто вызывать эти методы?

^ | ▾ · Reply · Share ›



AlexM → AlexM · 4 years ago

Вопрос отпал, нужно было всего лишь использовать SelenideElement вместо WebElement :)

В любом случае, спасибо за ваш шикарный продукт!

^ | v · Reply · Share ›



Andrei Solntsev Mod → AlexM · 4 years ago

Спасибо!

Да, тут есть несколько вариантов:

1. Ваш - Использовать SeleniumElement вместо WebElement
2. Не использовать аннотацию @FindBy. Определять поля ПО с типом By и обращаться к ним в методах ПО через \$().

^ | v · Reply · Share ›



Ярослав Орёл · 5 years ago

Андрей, а как подтягивается аннотация "FindBy" с локатором, когда мы ищем элемент с помощью \$(WebElement) ?

^ | v · Reply · Share ›



Andrei Solntsev Mod → Ярослав Орёл · 5 years ago

Привет!

Я не понял вопроса. Думаю, что вам не нужна аннотация @FindBy. Просто используйте \$() везде, где нужен веб-элемент.

^ | v · Reply · Share ›



Ярослав Орёл → Andrei Solntsev · 5 years ago

Я разобрался. Мне казалось вебэлементы инициализируются без участия PageFactory где-то после вызова \$(WebElement). Потом я нашел метод page(PageObject.class) и понял, что его надо вызвать до вызова \$(WebElement), чтобы все прошло успешно. Спасибо.

^ | v · Reply · Share ›



Andrei Solntsev Mod → Ярослав Орёл · 5 years ago

Слушайте, похоже, ваши тесты можно сильно упростить. Можете скинуть пример своего теста?

^ | v · Reply · Share ›



Ярослав Орёл → Andrei Solntsev · 5 years ago

На самом деле меня не тесты волнуют, а то как все работает под капотом в Вашем фреймворке. Для общего образования :)

^ | v · Reply · Share ›



Andrei Solntsev Mod → Ярослав Орёл • 5 years ago

Это похвально, конечно.

Но тесты должны волновать больше!

P.S. Если вы используете пэдж объект и инициализируете его методом `page(class)`, то метод `$` вам по идее больше не нужен. Вот что меня смутило. Получается, вы делаете двойную работу, если используете и `page()`, и `$()`.

^ | v • Reply • Share ›



Ярослав Орёл → Andrei Solntsev • 5 years ago

Разве что, если я захочу завернуть его в селенид элемент и пользоваться всеми его преимуществами по сравнению с голым веб элементом.

^ | v • Reply • Share ›



Andrei Solntsev Mod → Ярослав Орёл • 5 years ago

Это можно сделать проще. Объявите его сразу в пэдж объекте как `SelenideElement`:

```
class MYPage {  
  @FindBy(...)  
  private SelenideElement myElement;  
}
```

^ | v • Reply • Share ›



Ярослав Орёл → Andrei Solntsev • 5 years ago

Логично. Спасибо.

^ | v • Reply • Share ›



Василий • 5 years ago • edited

Здравствуйте, Андрей. Я пытаюсь найти элемент на странице с помощью Селенид и, к сожалению, элемент не находится. Адрес <http://manage.onesporttechn...> и элемент `id="email"`. Когда ищу с помощью селениума, то элемент виден.

Код для нахождения брал из quick start раздела `$(By.id("email")).shouldBe(visible);`

Заранее спасибо за помощь!

^ | v • Reply • Share ›



Andrei Solntsev Mod → Василий • 5 years ago

Таки неправда!

Вот этот тест у меня вполне зелёный.

^ | v • Reply • Share ›



^ | v • Reply • Share ›



Например, такой минус. Когда объявляешь элементы как поля класса, невозможно определить "динамические" элементы - имя которых заранее не знаешь. Или элементы, которых слишком много. Пример из виселицы: на экране горят все буквы алфавита, тест кликает некоторые буквы. Это что ж теперь, сделать пэдж объект с 33 полями? А когда добавится поддержка английского языка, добавить в него ещё 26 полей?

```
public void guessLetter(char letter) {
    $(by("letter", letter)).click();
}
```

CM. <https://github.com/selenide...>

^ | v • Reply • Share ›



12/14

Т.Д.

^ | v · Reply · Share ›

**Andrei Solntsev** Mod → ваня иванов · 5 years ago · edited

Очень просто - проверить. А ещё лучше сделать с ними какие-то полезные действия. Вы же не хотите просто проверить, что они есть. Вы ведь хотите проерить, что у них правильный текст, стиль или ещё что-нибудь. Или если это интерактивные элементы, хотите проверить, что они работают правильно.

То есть что-то типа такого:

```
page.header.shouldHave(text("Hello"));
page.loginWith("john", "qwerty");
page.error
.shouldBe(visible)
.shouldHave(cssClass("error"), text("Invalid password"));
```

^ | v · Reply · Share ›

**ваня иванов** → Andrei Solntsev · 5 years ago

эти проверки насколько я понял нужно делать в самом тесте? Не в классе страницы?

^ | v · Reply · Share ›

**Andrei Solntsev** Mod → ваня иванов · 5 years ago

Это уж как вы сами решите.

Вообще решение о том, что нужно что-то проверить, должно быть в тесте. Это и есть его задача - знать, что тестировать.

А знание о том, КАК это проверить, должно быть в пэджд обьекте. Это его задача - инкапсулировать знание об веб-элементах и работе с ними.

Поэтому я вижу так: метод для проверки должен быть в пэджд-объекте, а вызывать этот метод должен тест.

^ | v · Reply · Share ›

**ваня иванов** → Andrei Solntsev · 5 years ago

Теперь гораздо понятнее. Смотрел ваше выступление по поводу unit тестов. Еще скажите, Андрей, каким образом у вас хранятся пользователи для тестирования онлайн-банка? В каких файлах - xml, xls? Сейчас выбираю в каких файлах хранить пользователей для своих тестов.

^ | v · Reply · Share ›

**Andrei Sointsev** Mod → ваня иванов • 5 years ago

Привет!

В разных проектах по-разному. В некоторых пользователи прописаны прямо в тестах. Это самый простой и надёжный способ, рекомендую его.

В других проектах с более сложной инфраструктурой тестовые данные генерируем в базе данных с помощью, например, LiquiBase. В других - храним их в *.uml файлах (и написали свою утилитку, которая из uml генерирует INSERT в базу).

^ | v • Reply • Share ›

[Load more comments](#)

[✉ Subscribe](#) [D Add Disqus to your site](#)[Add Disqus](#)[Add](#) [⚠ Do Not Sell My Data](#)*The project is maintained by*

well-crafted software