PageObject Tour

Using the Page Object Pattern in ScalaTest

Page Object Pattern

This tour will show you how to use the PageObject library to write tests using the Page Object Pattern in ScalaTest.

How to define Page Objects?

`PageObject` / `PageModule`

PageObject / PageModule

A `PageObject` represents a “Page”

It has...

- `PageModule`s like “`content`”, “`header`”, “`footer`” or “`navigation`” to access the content of the page.
- an `atChecker()` (trait `AtChecker`) to see if the browser is currently on this page.
- no DOM access!

A `PageModule` represents an area of the Page that represents one logical unit.

It...

- can access the DOM using `BrowserPageDsl`.
- Should shield the DOM from other parts of the test.
- can contain other `PageModule`s if needed.

Example `PageObject`: `GoogleSearchHomePage`

Now we are going and try to test Google's search homepage.

First we will define a `PageObject`, later we write a test.

As we can expect, we need a function to search for something:

```
        def search(searchTerm: String): Unit
```

We define it inside of an `PageModule` because `search` need to access the DOM.

Example PageObject: GoogleSearchHomePage

```
case class GoogleSearchHomePage()
```

We use a case class to be able to write GoogleSearchHomePage() without the new Keyword.

agido PageObject

>ScalaTest™

Example PageObject: GoogleSearchHomePage

```scala
case class GoogleSearchHomePage() extends PageObject
```

Because we want to implement a Page Object, we need to implement the trait `PageObject`.

agido PageObject

SOFTWARE

>ScalaTest™

Example PageObject: GoogleSearchHomePage

```
case class GoogleSearchHomePage() extends PageObject {








  override def atChecker(): Boolean = pageTitle == "Google"
}
```

This is now a valid PageObject, only the atChecker() is required.

Example PageObject: GoogleSearchHomePage

```
case class GoogleSearchHomePage() extends PageObject {







  override def atChecker(): Boolean = pageTitle == "Google"
}
```

We just compare the pageTitle to see if the Browser is at the Page we expect.

Example PageObject: GoogleSearchHomePage

```
case class GoogleSearchHomePage() extends PageObject {







  override def atChecker(): Boolean = pageTitle == "Google"
}
```

But this PageObject is useless
because we can’t do anything with it...

agido PageObject SOFTWARE

>ScalaTest™

Example `PageObject`: `GoogleSearchHomePage`

```
case class GoogleSearchHomePage() extends PageObject {


  object content {






  }

  override def atChecker(): Boolean = pageTitle == "Google"
}
```

Because of this we add some content.

Example PageObject: GoogleSearchHomePage

```
case class GoogleSearchHomePage() extends PageObject {

  object content extends PageModule {




  }

  override def atChecker(): Boolean = pageTitle == "Google"
}
```

A PageModule is required to access the DOM.

agido PageObject SOFTWARE

>ScalaTest™

Example PageObject: GoogleSearchHomePage

```
case class GoogleSearchHomePage() extends PageObject {


  object content extends PageModule {
    private val q = textField(name("q"))




  }

  override def atChecker(): Boolean = pageTitle == "Google"
}
```

The search box on Google’s Homepage
has the attribute name=”q”

Example PageObject: GoogleSearchHomePage

```
case class GoogleSearchHomePage() extends PageObject {


  object content extends PageModule {
    private val q = textField(name("q"))




  }

  override def atChecker(): Boolean = pageTitle == "Google"
}
```

And we expect an Element of type TextField.

Example PageObject: GoogleSearchHomePage

```scala
case class GoogleSearchHomePage() extends PageObject {


  object content extends PageModule {
    private val q = textField(name("q"))




  }

  override def atChecker(): Boolean = pageTitle == "Google"
}
```

We don’t want to expose this DOM stuff…

Example PageObject: GoogleSearchHomePage

```
case class GoogleSearchHomePage() extends PageObject {


  object content extends PageModule {
    private val q = textField(name("q"))






  }

  override def atChecker(): Boolean = pageTitle == "Google"
}
```

...because the PageModule should shield all PageObject clients from DOM details.

agido PageObject SOFTWARE

>ScalaTest™

Example PageObject: GoogleSearchHomePage

```
case class GoogleSearchHomePage() extends PageObject {


  object content extends PageModule {
    private val q = textField(name("q"))

    def search(searchTerm: String): Unit = {


    }
  }

  override def atChecker(): Boolean = pageTitle == "Google"
}
```

As a last step we provide a “public API“ to allow the clients of this `PageObject` to search for something.

Example PageObject: GoogleSearchHomePage

```
case class GoogleSearchHomePage() extends PageObject {


  object content extends PageModule {
    private val q = textField(name("q"))

    def search(searchTerm: String): Unit = {
      q.value = searchTerm
      submit()
    }
  }

  override def atChecker(): Boolean = pageTitle == "Google"
}
```

We fill the search term into the q Element.

Example PageObject: GoogleSearchHomePage

```
case class GoogleSearchHomePage() extends PageObject {


  object content extends PageModule {
    private val q = textField(name("q"))

    def search(searchTerm: String): Unit = {
      q.value = searchTerm
      submit()
    }
  }

  override def atChecker(): Boolean = pageTitle == "Google"
}
```

And finally we `submit()` the search form.

agido PageObject

>ScalaTest™

Example PageObject: GoogleSearchHomePage

```
case class GoogleSearchHomePage() extends PageObject with UrlPage {
  val url = "https://www.google.com/"

  object content extends PageModule {
    private val q = textField(name("q"))

    def search(searchTerm: String): Unit = {
      q.value = searchTerm
      submit()
    }
  }

  override def atChecker(): Boolean = pageTitle == "Google"
}
```

Because we want to navigate to this page, we also have to provide the URL.

agido PageObject

>ScalaTest™

Example PageObject: GoogleSearchHomePage

```scala
case class GoogleSearchHomePage() extends PageObject with UrlPage {
  val url = "https://www.google.com/"

  object content extends PageModule {
    private val q = textField(name("q"))

    def search(searchTerm: String): Unit = {
      q.value = searchTerm
      submit()
    }
  }

  override def atChecker(): Boolean = pageTitle == "Google"
}
```

agido PageObject

>ScalaTest™

How to write a Test Spec?

`GoogleSearchSpec` example using ScalaTest

agido PageObject SOFTWARE

>ScalaTest™

`GoogleSearchSpec` example using ScalaTest

For this example we want to test the Google Search Homepage.

After the search was submitted we expect the corresponding result page:

- Given
  - The Google Search Homepage
- When
  - Searching for “Cheese!”
- Then
  - We are at the Google Search Result Page

GoogleSearchSpec example using ScalaTest

```
class GoogleSearchSpec extends FunSpec {

  describe("Google Search") {
    it("should change its title based on the term searched") {









    }
  }
}
```

Initial we write a plain ScalaTest TestSpec.

GoogleSearchSpec example using ScalaTest

```
class GoogleSearchSpec extends FunSpec {

  describe("Google Search") {
    it("should change its title based on the term searched") {





    }
  }
}
```

You can also use any other testing style supported by ScalaTest.

GoogleSearchSpec example using ScalaTest

```
class GoogleSearchSpec extends FunSpec with GivenWhenThen {

  describe("Google Search") {
    it("should change its title based on the term searched") {
      Given("The Google Search Homepage")


      When("Searching for “Cheese!”")


      Then("We are at the Google Search Result Page")

    }
  }
}
```

Optionally you can use GivenWhenThen.

GoogleSearchSpec example using ScalaTest

```
class GoogleSearchSpec extends FunSpec with GivenWhenThen with PageObjectSuite
{
  describe("Google Search") {
    it("should change its title based on the term searched") {
      Given("The Google Search Homepage")


      When("Search


      Then("We are at the Google Search Result Page")


    }
  }
}
```

Because we want to test a web page
the TestSpec needs to launch a Web Browser…
The PageObject library will do this for you!

agido PageObject SOFTWARE

>ScalaTest™

GoogleSearchSpec example using ScalaTest

```
class GoogleSearchSpec extends FunSpec with GivenWhenThen with PageObjectSuite
{
  describe("Google Search") {
    it("should change its title based on the term searched") {
      Given("The Google Search Homepage")
      val page = to(GoogleSearchHomePage())

      When("Searching for “Cheese!”")


      Then("We are at the Google Search Result Page")


    }
  }
}
```

Navigate the Browser to the given Page represented by a PageObject.

agido PageObject SOFTWARE

>ScalaTest™

GoogleSearchSpec example using ScalaTest

```
class GoogleSearchSpec extends FunSpec with GivenWhenThen with PageObjectSuite
{
  describe("Google Search") {
    it("should change its title based on the term searched") {
      Given("The Google Search Homepage")
      val page = to(GoogleSearchHomePage())

      When("Searching for “Cheese!”")
      page.content.search("Cheese!")

      Then("We are at the Google Search Result Page")

    }
  }
}
```

Search for “Cheese!”

agido PageObject SOFTWARE

>ScalaTest™

GoogleSearchSpec example using ScalaTest

```
class GoogleSearchSpec extends FunSpec with GivenWhenThen with PageObjectSuite
{
  describe("Google Search") {
    it("should change its title based on the term searched") {
      Given("The Google Search Homepage")
      val page = to(GoogleSearchHomePage())

      When("Searching for “Cheese!”")
      page.content.search("Cheese!")

      Then("We are at the Google Search Result Page")
      at(GoogleSearchResultPage("Cheese!"))
    }
  }
}
```

And finally check that we are on the desired page...

agido PageObject SOFTWARE

>ScalaTest™

GoogleSearchSpec example using ScalaTest

```
class GoogleSearchSpec extends FunSpec with GivenWhenThen with PageObjectSuite
{
  describe("Google Search") {
    it("should change its title based on the term searched") {
      Given("The Google Search Homepage")
      val page = to(GoogleSearchHomePage())

      When("Searching for “Cheese!”")
      page.content.search("Cheese!")

      Then("We are at the Google Search Result Page")
      at(GoogleSearchResultPage("Cheese!"))
    }
  }
}
```

agido PageObject SOFTWARE

>ScalaTest™

Example Page Object: GoogleSearchResultPage

```
case class GoogleSearchResultPage(searchTerm: String) extends PageObject {
  override def atChecker(): Boolean = {
    pageTitle.startsWith(s"$searchTerm - Google")
  }
}
```

We do not want to withhold the `GoogleSearchResultPage`...

agido PageObject

>ScalaTest™

agido PageObject SOFTWARE

>ScalaTest™

Colors Used

Links

Comments

Types

Keywords

"Strings"

Functions

Variables (val, var and object)