EAST
COAST
PRODUCT

Page Objects in WebdriverIO

Ana Prpic, QA Analyst

@ECPBoston

EastCoastProduct

About me

Contact info:

- Twitter: @a_prpic
- Email: ana@ecp.io
- www.eastcoastproduct.com

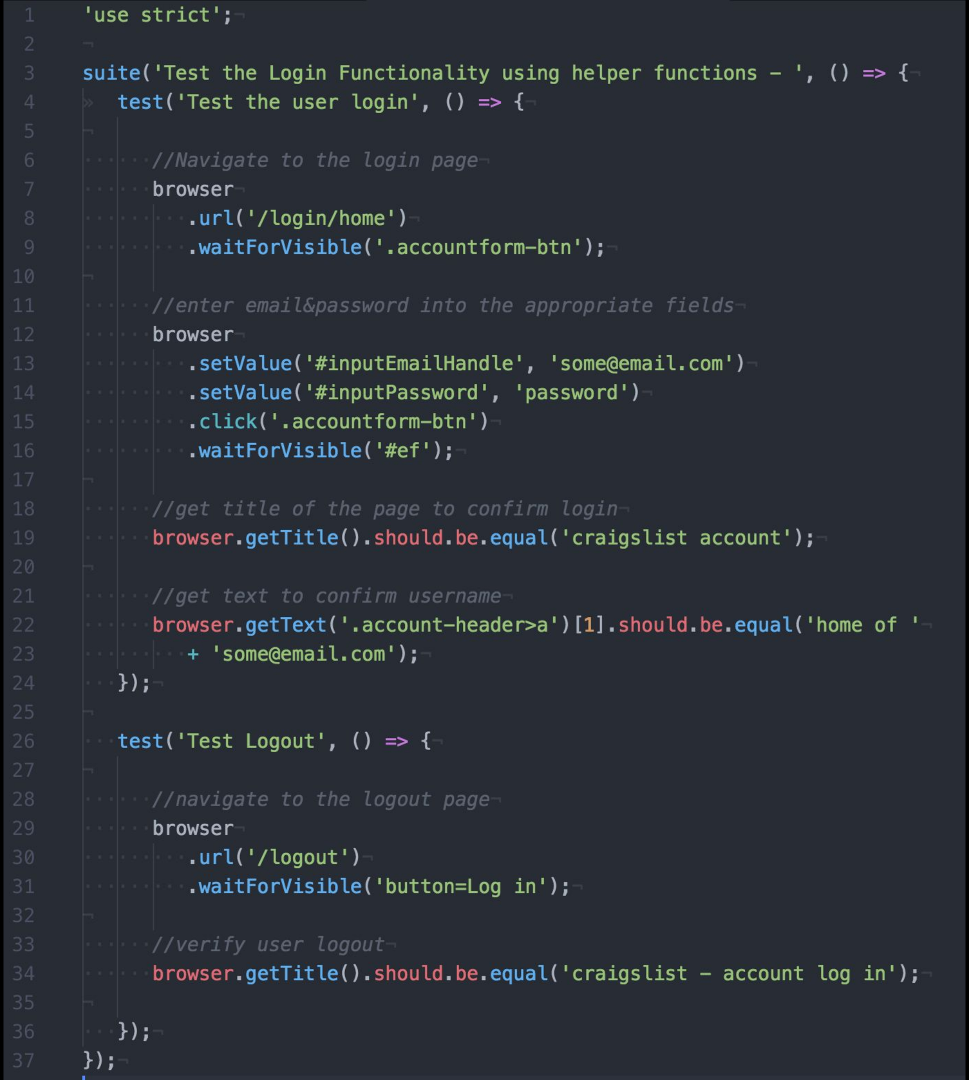
Agenda

- What is Page Object Pattern?
- How does a good Page Object look like?
- Page Objects in WebdriverIO
- Examples, tips & tricks
- Discussion

Problems

- When writing tests - referring to elements within the web page - imagine you have 10 different scripts that are using the same element on a page
- What happens when the element changes?
- Poor readability of the test scripts
- No code reuse

‘Login’ test script



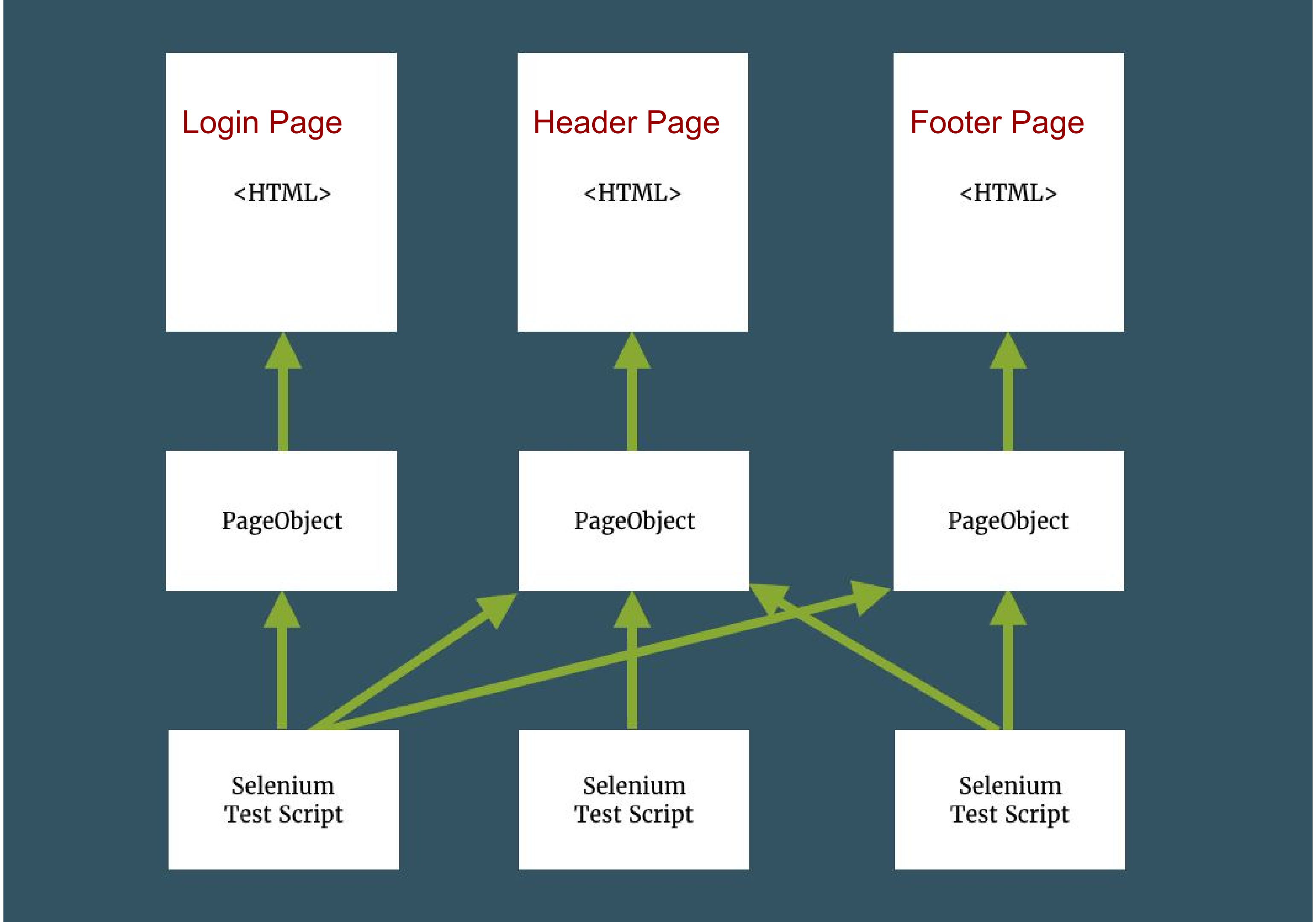
```
'use strict';

suite('Test the Login Functionality using helper functions - ', () => {
  test('Test the user login', () => {

    //Navigate to the login page
    browser
      .url('/login/home')
      .waitForVisible('.accountform-btn');

    //enter email&password into the appropriate fields
    browser
      .setValue('#inputEmailHandle', 'some@email.com')
      .setValue('#inputPassword', 'password')
      .click('.accountform-btn')
      .waitForVisible('#ef');

    //get title of the page to confirm login
    browser.getTitle().should.be.equal('craigslist account');

    //get text to confirm username
    browser.getText('.account-header>a')[1].should.be.equal('home of '
      + 'some@email.com');
  });

  test('Test Logout', () => {

    //navigate to the logout page
    browser
      .url('/logout')
      .waitForVisible('button=Log in');

    //verify user logout
    browser.getTitle().should.be.equal('craigslist - account log in');

  });
});
```

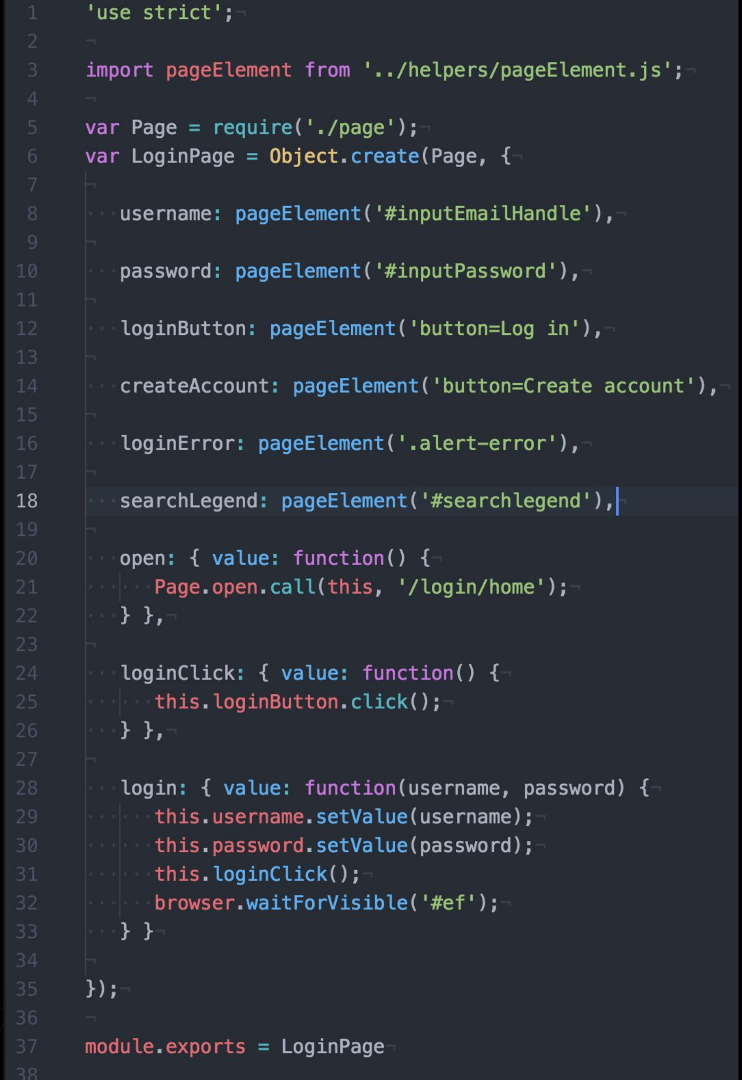
- This one script looks easy enough to maintain, but what happens when the suite grows?

- ….or when a UI element changes?

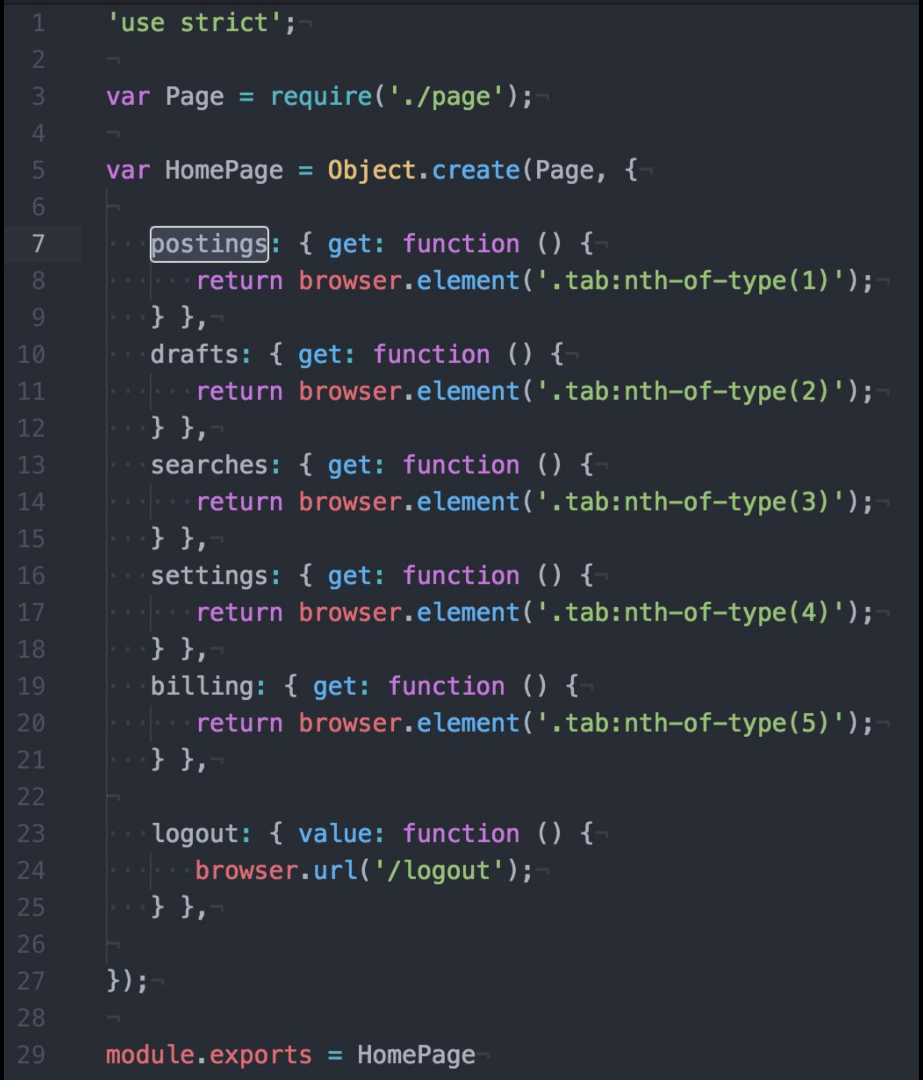
Page Object Pattern

e.g.

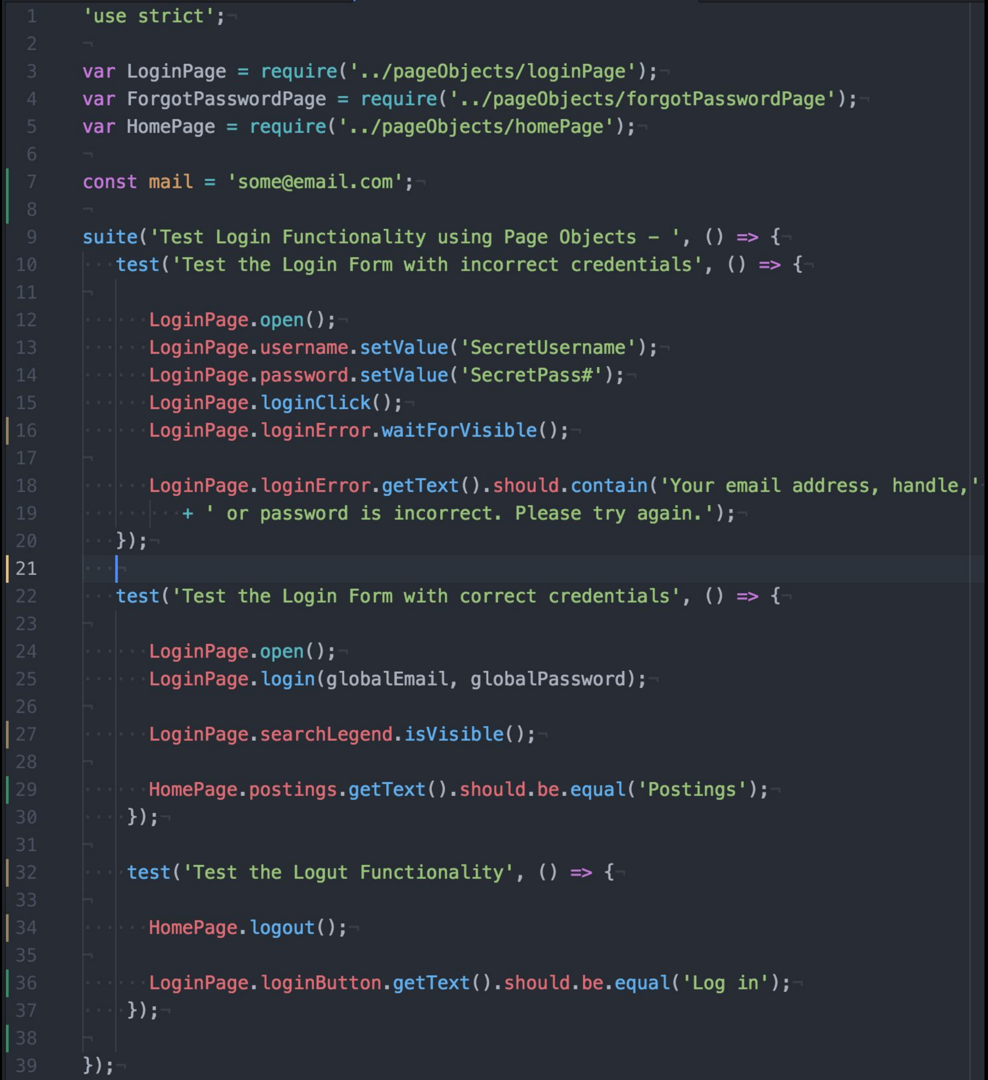

Login Page
<HTML>
Header Page
<HTML>
Footer Page
<HTML>
PageObject
PageObject
PageObject
Selenium Test Script
Selenium Test Script
Selenium Test Script



```
'use strict';

import pageElement from '../helpers/pageElement.js';

var Page = require('./page');
var LoginPage = Object.create(Page, {

  username: pageElement('#inputEmailHandle'),

  password: pageElement('#inputPassword'),

  loginButton: pageElement('button=Log in'),

  createAccount: pageElement('button=Create account'),

  loginError: pageElement('.alert-error'),

  searchLegend: pageElement('#searchlegend'),

  open: { value: function() {
    Page.open.call(this, '/login/home');
  } },

  loginClick: { value: function() {
    this.loginButton.click();
  } },

  login: { value: function(username, password) {
    this.username.setValue(username);
    this.password.setValue(password);
    this.loginClick();
    browser.waitForVisible('#ef');
  } }

});

module.exports = LoginPage
```



```
'use strict';

var Page = require('./page');

var HomePage = Object.create(Page, {

  postings: { get: function () {
      return browser.element('.tab:nth-of-type(1)');
  } },
  drafts: { get: function () {
      return browser.element('.tab:nth-of-type(2)');
  } },
  searches: { get: function () {
      return browser.element('.tab:nth-of-type(3)');
  } },
  settings: { get: function () {
      return browser.element('.tab:nth-of-type(4)');
  } },
  billing: { get: function () {
      return browser.element('.tab:nth-of-type(5)');
  } },

  logout: { value: function () {
      browser.url('/logout');
  } },

});

module.exports = HomePage
```



```
'use strict';

var LoginPage = require('../pageObjects/loginPage');
var ForgotPasswordPage = require('../pageObjects/forgotPasswordPage');
var HomePage = require('../pageObjects/homePage');

const mail = 'some@email.com';

suite('Test Login Functionality using Page Objects - ', () => {
   test('Test the Login Form with incorrect credentials', () => {

      LoginPage.open();
      LoginPage.username.setValue('SecretUsername');
      LoginPage.password.setValue('SecretPass#');
      LoginPage.loginClick();
      LoginPage.loginError.waitForVisible();

      LoginPage.loginError.getText().should.contain('Your email address, handle,'
         + ' or password is incorrect. Please try again.');
   });
   
   test('Test the Login Form with correct credentials', () => {

      LoginPage.open();
      LoginPage.login(globalEmail, globalPassword);

      LoginPage.searchLegend.isVisible();

      HomePage.postings.getText().should.be.equal('Postings');
    });

    test('Test the Logut Functionality', () => {

      HomePage.logout();

      LoginPage.loginButton.getText().should.be.equal('Log in');
    });

});
```

A good Page Object

1. Contains state (locators!)
2. Contains behavior (methods that enable interaction with the page)
3. Returns info about the page (e.g. a new page object, **never** a web element!)
4. Verifies a page ready state as part of initialization with a found element (visible on the page, clickable...)
5. Has less than 200 lines of code
6. Has no hard-coded sleeps!
7. Uses explicit waits
8. Contains no assertions!

Useful links:

- Page Objects by Martin Fowler
- Selenium HQ about Page Objects
- Page Objects by WebdriverIO

Want More?

- GitHub repo with a few examples: https://github.com/EastCoastProduct/webdriverio-blog-example
- ECP is hiring senior developers!
    - Contact monika@eastcoastproduct.com
- Come to our **LevelUp Development** meetups in Split:
    - We're discussing the latest tech over pizza and beer
    - Taking suggestions for topics
    - Looking for presenters

Thank you!

@ECPBoston

EastCoastProduct