

Functional (automatic) testing

Manual testing is:

- ▶ Boring
- ▶ Labor intensive
- ▶ Not repeatable
- ▶ Costs time



if you can

Selenium

web testing automation framework



Selenium tester's friend and developer's enemy (?)

- ▶ Records actions within browser
 - ▶ Exports actions to code (java, C#, ruby)
 - ▶ Replicates web actions
 - ▶ Supports automatic testing
-

How Selenium works



1. Browser (actions) ->
 2. Selenium IDE (record actions & export) ->
 3. Selenium Java code (+ other coding stuff) ->
 4. Java code – unit testing or not ->
 - 5.1 (run tests) with unit testing framework in browser
 - 5.2 (run code) in browser without unit testing
-

Selenium WebDriver



- ▶ Better than Selenium RC Server
 - ▶ Native calls to browser
 - ▶ RC Server injected JavaScript functions
 - ▶ Simple to integrate – package repos (Maven, NuGet)
-

Selenium Locators

Search Html elements by:

- ▶ Id
 - ▶ TagName
 - ▶ ClassName
 - ▶ LinkText
 - ▶ CSS
 - ▶ XPATH
-

Selenium Locators example



```
<form id="login">
  <input id="123" name="username" type="text"/>
  <input name="password" type="password"/>
  <input name="submit" type="submit" value="Continue!"/>
</form>
```

```
var driver = new FirefoxDriver();
driver.Navigate().GoToUrl("http://www.mywebpage.ro/");
IWebElement query =
    driver.FindElement(By.Name("username"));
query = driver.FindElement(By.Id("123"));
query = driver.FindElement(By.XPath("//*[@id="123"]));
```

Extensions of WebDriver



Can automate testing on several browsers:

- ▶ ChromeDriver
 - ▶ HtmlUnitDriver
 - ▶ FirefoxDriver
 - ▶ InternetExplorerDriver
-

Explicit & Implicit waits



Web major drawback : latency of page/elements

loading

```
var driver = new FirefoxDriver();
```

```
driver.Navigate().GoToUrl("http://www.google.com/");
```

```
IWebElement query = driver.FindElement(By.Name("q"));
```

```
query.SendKeys("Cheese");
```

```
query.Submit();
```

```
// Will NOT see: "Cheese - Google Search"
```

```
Console.WriteLine("Page title is: " + driver.Title);
```

Page with “Cheese” search request do not loads
before the *title* is printed

Explicit & Implicit waits



```
var driver = new FirefoxDriver();  
driver.Navigate().GoToUrl("http://www.google.com/");  
IWebElement query = driver.FindElement(By.Name("q"));  
query.SendKeys("Cheese");  
query.Submit();
```

```
WebDriverWait wait = new WebDriverWait(driver,  
    TimeSpan.FromSeconds(10));  
    wait.Until((d) => { return  
        d.Title.ToLower().StartsWith("cheese"); });
```

```
// Should see: "Cheese - Google Search"  
Console.WriteLine("Page title is: " + driver.Title);
```

Explicit & **Implicit** waits



```
var driver = new FirefoxDriver();
driver.Manage().Timeouts().ImplicitlyWait(TimeSpan.FromSeconds(1
0));
driver.Navigate().GoToUrl("http://www.google.com/");
IWebElement query = driver.FindElement(By.Name("q"));
query.SendKeys("Cheese");
query.Submit();

// Should see: "Cheese - Google Search"
Console.WriteLine("Page title is: " + driver.Title);
```

Arrange -> Act -> Assert



// arrange

... create and set SUT parameters

// act

... call SUT method/methods

// assert

... verify the obtained result vs. expected result

Verify vs. Assert

- ▶ If **Assert** fails -> Test **fails immediately**
 - ▶ If **Verify** fails -> Test continues and **fails at the end**
-

Examples

... TO BE CONTINUED
