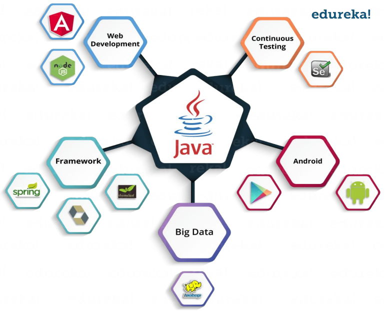
Testing Web Apps 101

Presenter: Rashan Smith

About me

Red Hat

INFORMATION TECHNOLOGY DISASTER RESOURCE CENTER



GRACE HOPPER CELEBRATION

ANITA B.ORG

MLH

MAJOR LEAGUE HACKING

Web Testing

Is this website/web app **INFORMATIVE**?

Is this website/web app **ACCESSIBLE**?

Is this website/web app **USER FRIENDLY**?

The process to evaluate the functionality of a web application with an intent to find whether the developed web app met the specified requirement or not and to identify the defects to ensure that the product is defect free in order to produce the quality product.

Types of Web Testing

Functionality Testing: Validating forms, text boxes, links, etc

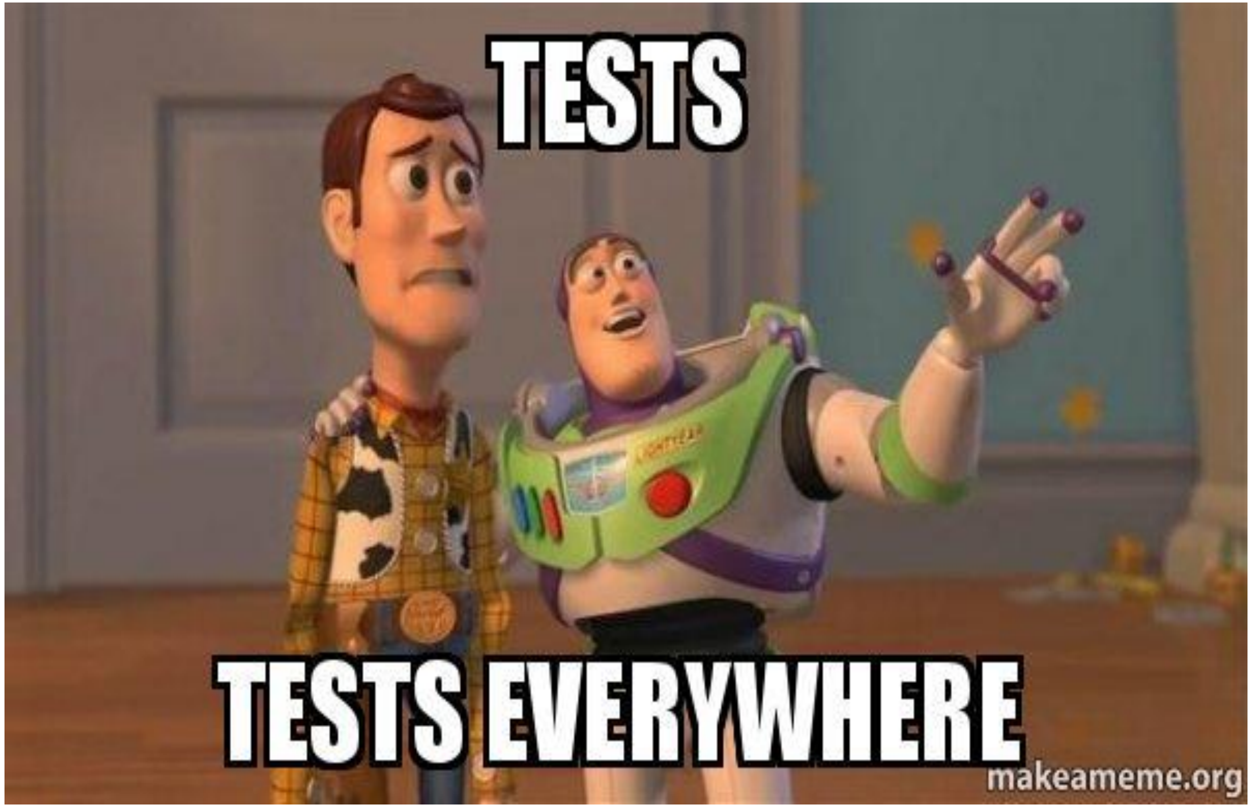
Usability Testing: How user friendly is the app, alignment issues, etc

Interface Testing: Interactions between servers

Compatibility Testing: Ensure web app is displayed across different browsers

Security Testing: Protected against harmful or unauthorized actions

Performance Testing: How system performs under different workloads (load and stress)


TESTS
TESTS EVERYWHERE
makeameme.org

Manual Testing

Pros:

- Cheaper initial investment
- Easier to onboard
- Better for short term projects
- Better for projects where requirements or GUI changes frequently

Cons:

- Time consuming
- Less reliable
- Expensive in the long run

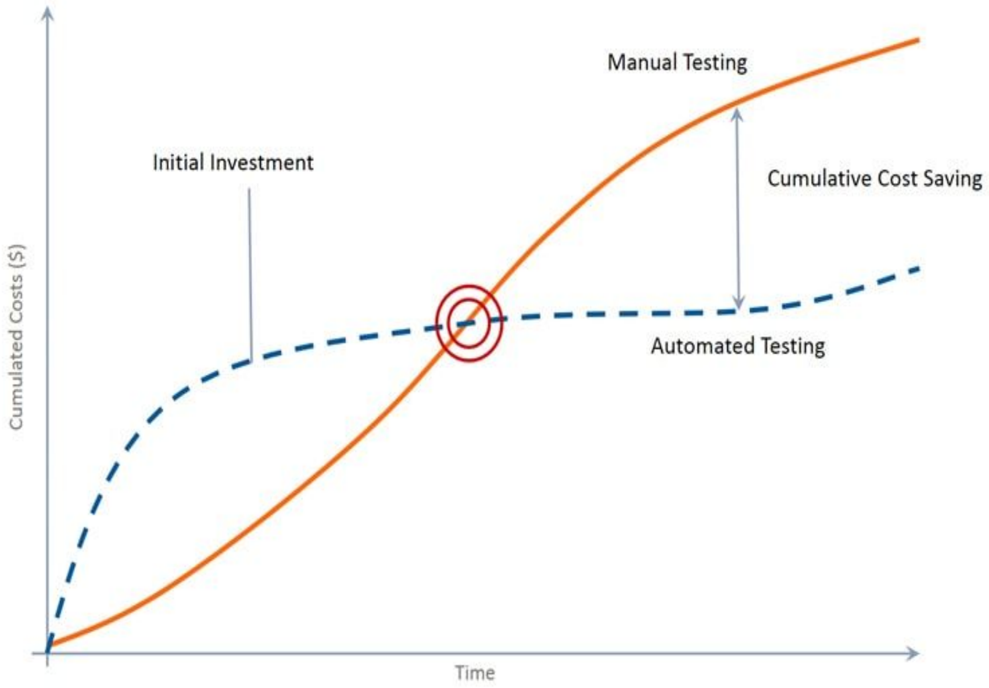
Automated Testing

Pros:

- Cost Effective in the long run
- Faster/Efficient
- Visibility of results

Cons:

- Can be an expensive investment
- Takes time to learn
- They can’t test for everything (color, font size) - Applitools*



Selenium Webdriver

Free, open source test automation tool

Demonstration in Java

Basic Programming Knowledge

Understanding of HTML/CSS elements

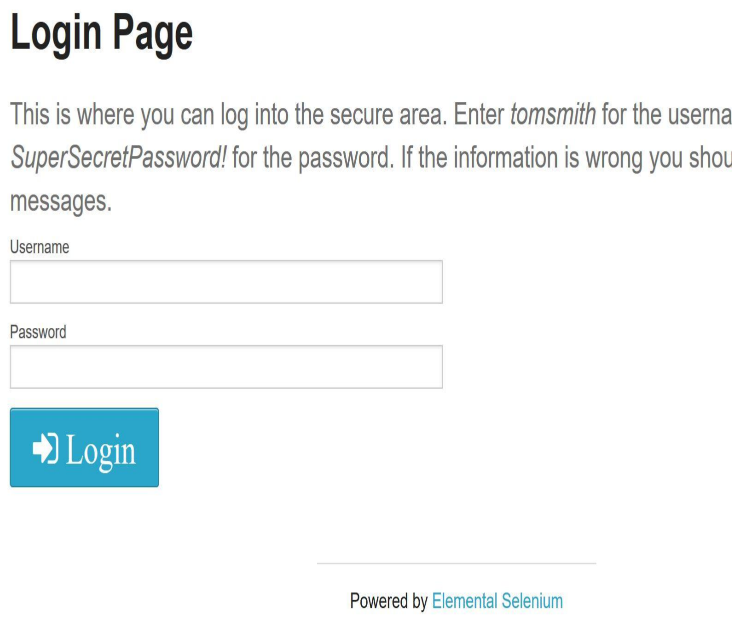
Key Selenium Concepts

How to automate interacting with a web app:

1. Create Selenium WebDriver Instance
2. Choose your web browser
3. Navigate to the website
4. Find the html element
5. Perform an action on the html element
6. Validate that the result is as expected
7. Close Driver

Demo

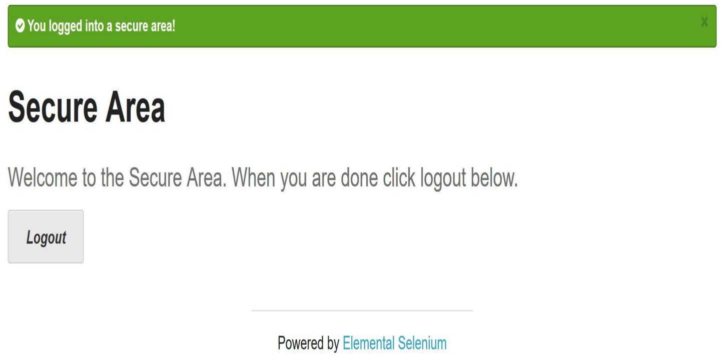
http://the-internet.herokuapp.com/login



Login Page

This is where you can log into the secure area. Enter *tomsmith* for the userna
SuperSecretPassword! for the password. If the information is wrong you shou
messages.

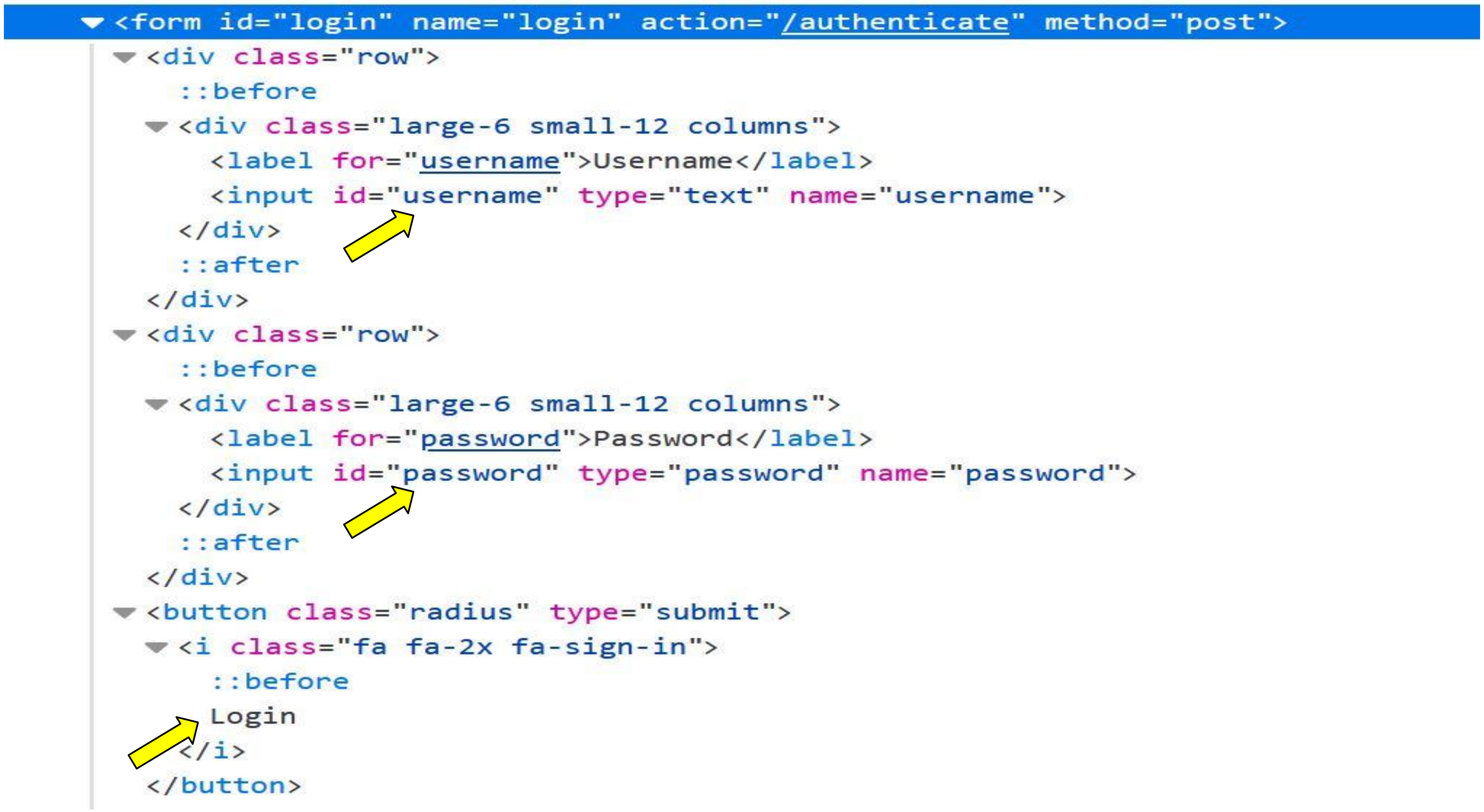
http://the-internet.herokuapp.com/secure



You logged into a secure area!

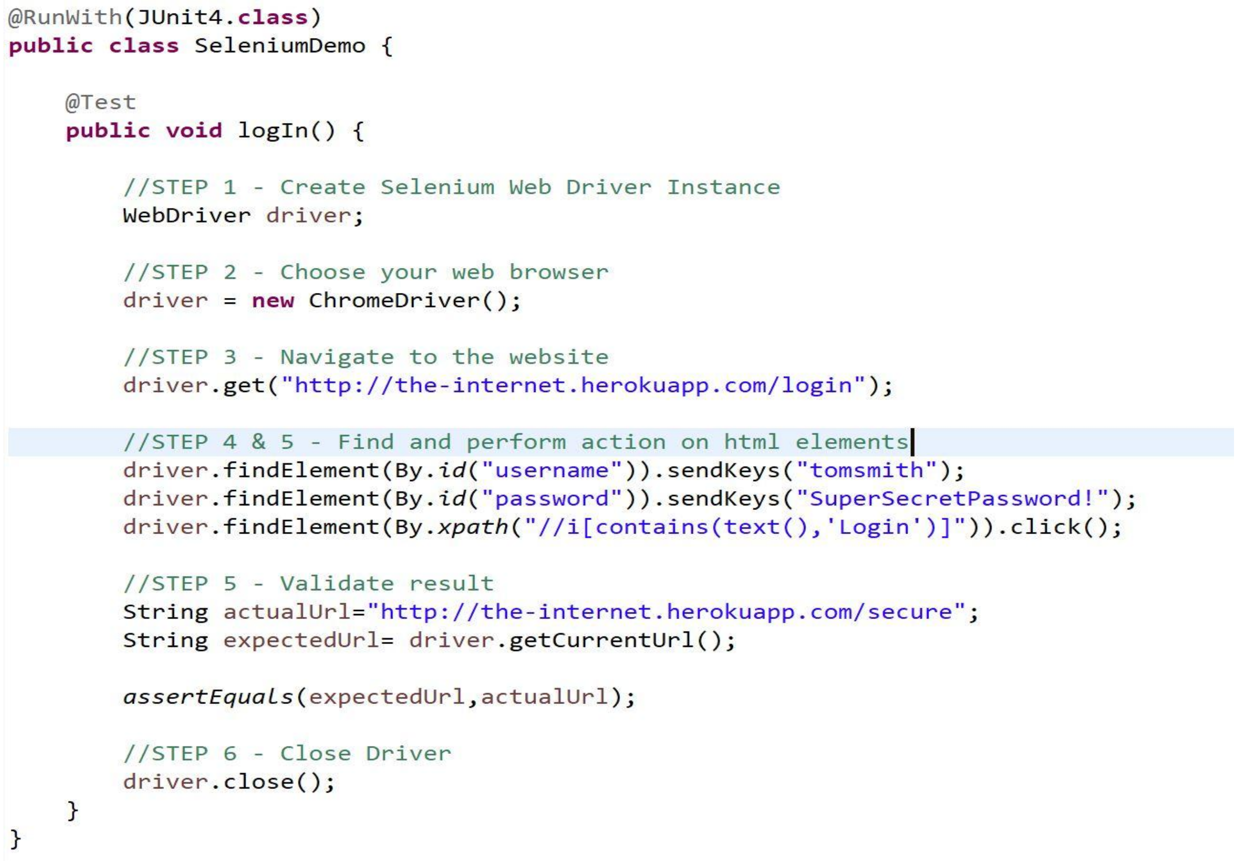
Secure Area

Welcome to the Secure Area. When you are done click logout below.



```
▼ <form id="login" name="login" action="/authenticate" method="post">
  ▼ <div class="row">
      ::before
    ▼ <div class="large-6 small-12 columns">
        <label for="username">Username</label>
        <input id="username" type="text" name="username">
      </div>
      ::after
    </div>
  ▼ <div class="row">
      ::before
    ▼ <div class="large-6 small-12 columns">
        <label for="password">Password</label>
        <input id="password" type="password" name="password">
      </div>
      ::after
    </div>
  ▼ <button class="radius" type="submit">
    ▼ <i class="fa fa-2x fa-sign-in">
        ::before
        Login
      </i>
    </button>
```

Java Code



```java
@RunWith(JUnit4.class)
public class SeleniumDemo {

    @Test
    public void logIn() {

        //STEP 1 - Create Selenium Web Driver Instance
        WebDriver driver;

        //STEP 2 - Choose your web browser
        driver = new ChromeDriver();

        //STEP 3 - Navigate to the website
        driver.get("http://the-internet.herokuapp.com/login");

        //STEP 4 & 5 - Find and perform action on html elements
        driver.findElement(By.id("username")).sendKeys("tomsmith");
        driver.findElement(By.id("password")).sendKeys("SuperSecretPassword!");
        driver.findElement(By.xpath("//i[contains(text(),'Login')]")).click();

        //STEP 5 - Validate result
        String actualUrl="http://the-internet.herokuapp.com/secure";
        String expectedUrl= driver.getCurrentUrl();

        assertEquals(expectedUrl,actualUrl);

        //STEP 6 - Close Driver
        driver.close();
    }
}
```

Automated Visual Testing

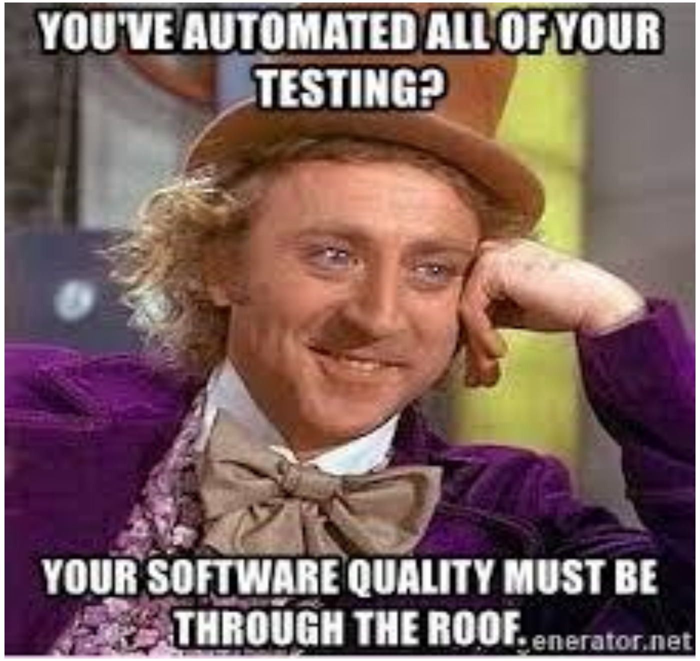
Regression Testing

Key Applitools Concepts

How to automate interacting with a web app:

1. Create Selenium WebDriver Instance
2. Choose your web browser
3. Initialize & open eyes
4. Navigate to website
5. Take screenshot
6. Close eyes
7. Close driver

Next Steps


YOU'VE AUTOMATED ALL OF YOUR TESTING?
YOUR SOFTWARE QUALITY MUST BE THROUGH THE ROOF.
enerator.net

Learning Resources

TAU TEST AUTOMATION UNIVERSITY

Thank You! :)

Rashan Smith

@shanster_242