Introduction To Selenium Testing

interworks.cloud™

Manolis Theodoroudis

- Software Automation Test Lead in interworks.cloud
- 10 years experience in Automation testing
- Worked in other fields like sales, industrial automation and IT for 5 more years.

https://www.linkedin.com/in/manolis-theodoroudis/

Expectations

interworks.cloud™

- Learn Basic Concepts of Quality and Testing
- Learn how to use Webdriver Selenium to locate and use web elements in DOM.
- Learn how to setup a basic Testing framework in Java
- Learn what page object model is
- Learn how to run and debug a test.
- Learn how to make a basic test plan

What is Quality?

interworks.cloud™

Software Quality Definition

Quality means “How well software complies to a given design, based on requirements or specifications.”

interworks.cloud™

Quality is everyone's responsibility

Processes that standardize the work and minimize the human error

<u>Technical</u>

- Testing
- Continuous Integration / Continuous Deployment
- Code Reviews
- Static Analysis tools

<u>Non Technical</u>

- Good Design e.g. Test-Driven Development
- Process / Document Reviews
- Training

What is Testing?

Testing is the process where we ensure our software

- Does not have bugs
- Meets the technical requirements
- Meets the user requirements

interworks.cloud™

How do we ensure the technical and user requirements?

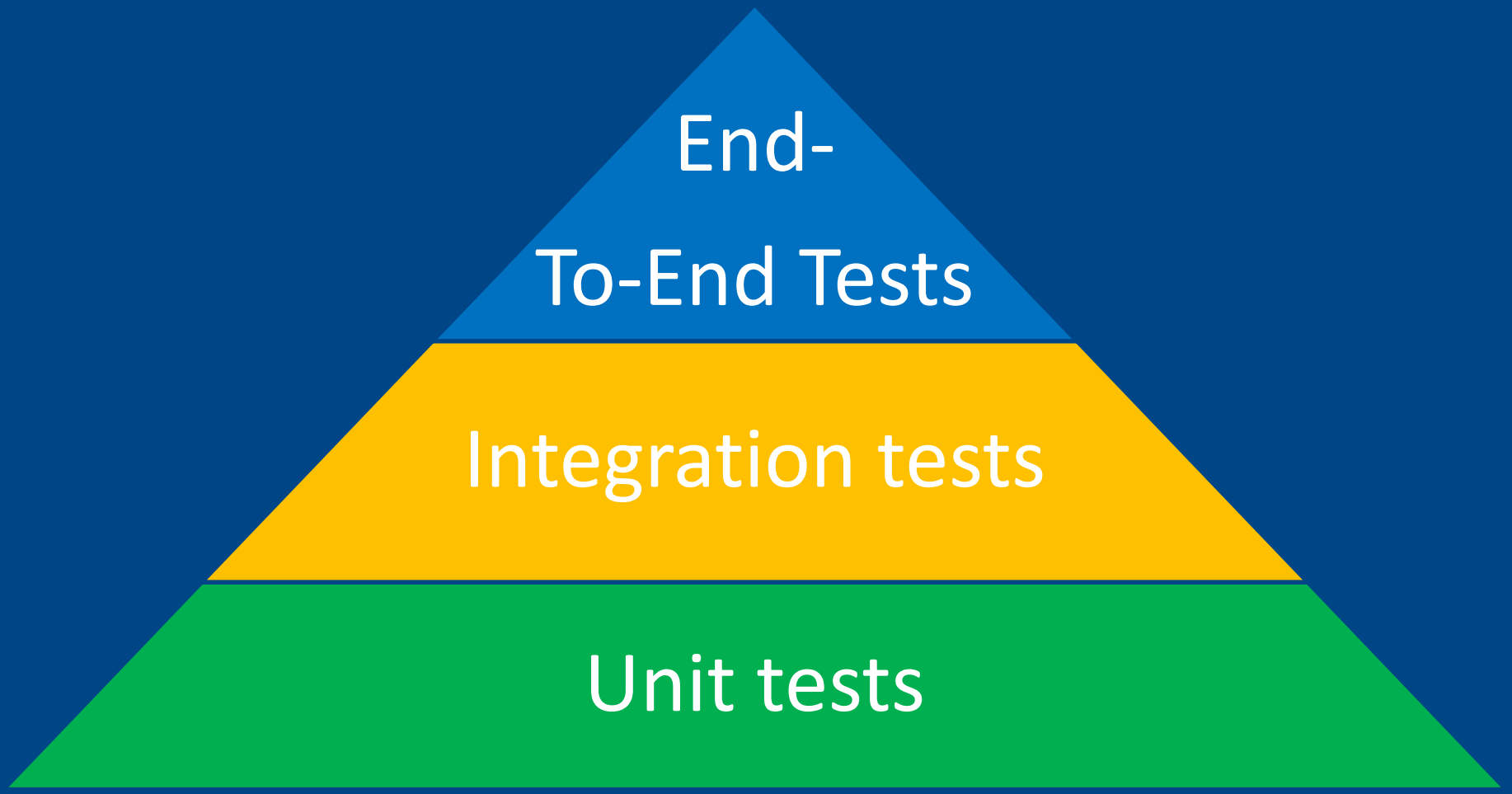
- Create a Test Plan
  - Testing Scope
  - Test Approach
  - Test Environment
  - Test Scenarios
  - Test Entry and Exit Criteria

interworks.cloud™

How do we test Software?

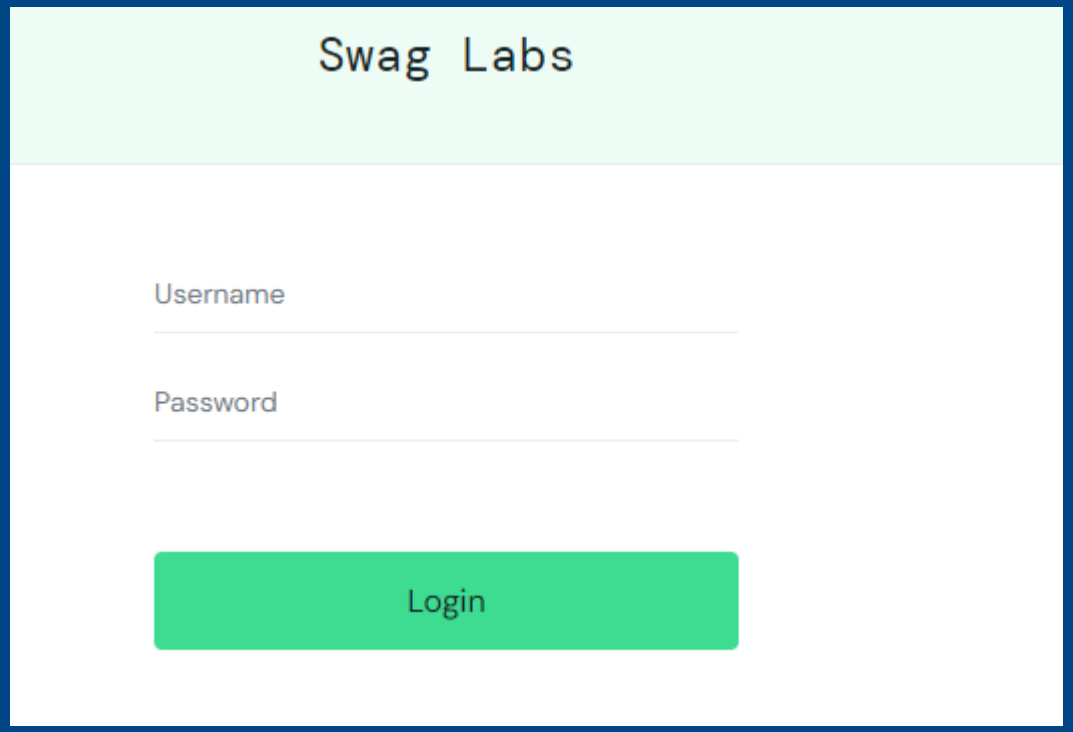
- Manual Testing
  - Exploratory Testing
  - User Acceptance Testing
- Automation Testing
  - Functional Testing
    - UI tests
    - Api Tests
  - Non-Functional Testing
    - Security
    - Performance

interworks.cloud™



Testing Pyramid

- **How would you test a login page?**



interworks.cloud™

Test Scenarios

- Sunny day scenarios
  - Valid User
  - Valid Trial User
- Rainy day scenarios
  - Invalid password
  - Invalid username
  - Password of another user
  - Expired Password
  - Other language characters
  - SQL injection

interworks.cloud™

Test plan of automated test cases

- UI Tests
  - Valid User
  - Invalid password
- Api Tests
  - Valid Trial User
  - Invalid username
  - Password of another user
  - Expired Password
  - Other language characters
  - SQL injection

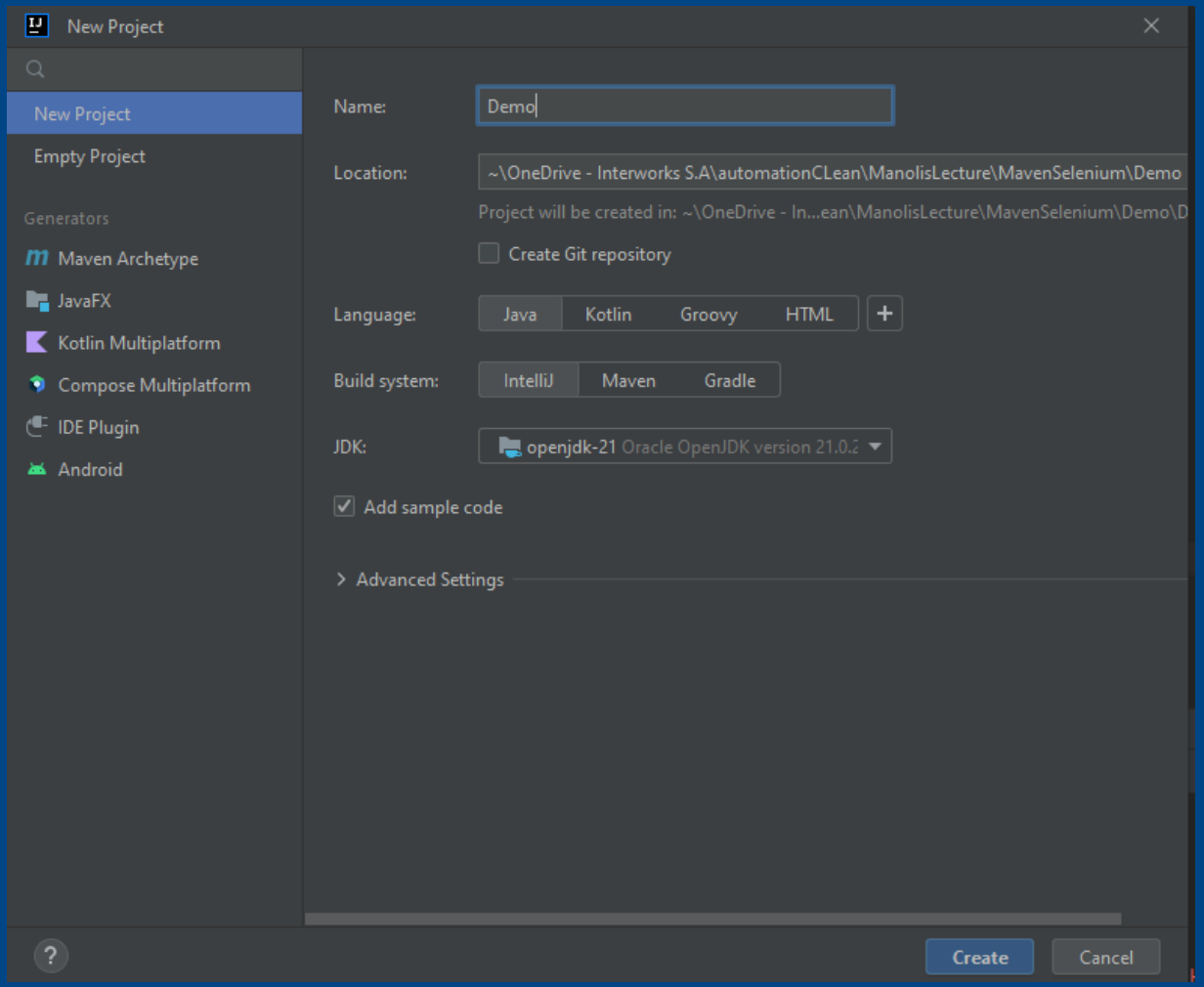
How to begin writing code?

- **System requirements**
  - Download and install Java 21 JDK
    - https://www.oracle.com/java/technologies/downloads/
  - Download IntelliJ community edition
    - https://www.jetbrains.com/idea/
  - Download Selenium Server (Grid) jar files
    - https://www.selenium.dev/downloads/
  - Download Gecko Driver for Mozilla Firefox
    - https://github.com/mozilla/geckodriver/releases

interworks.cloud™

Open IntelliJ and click

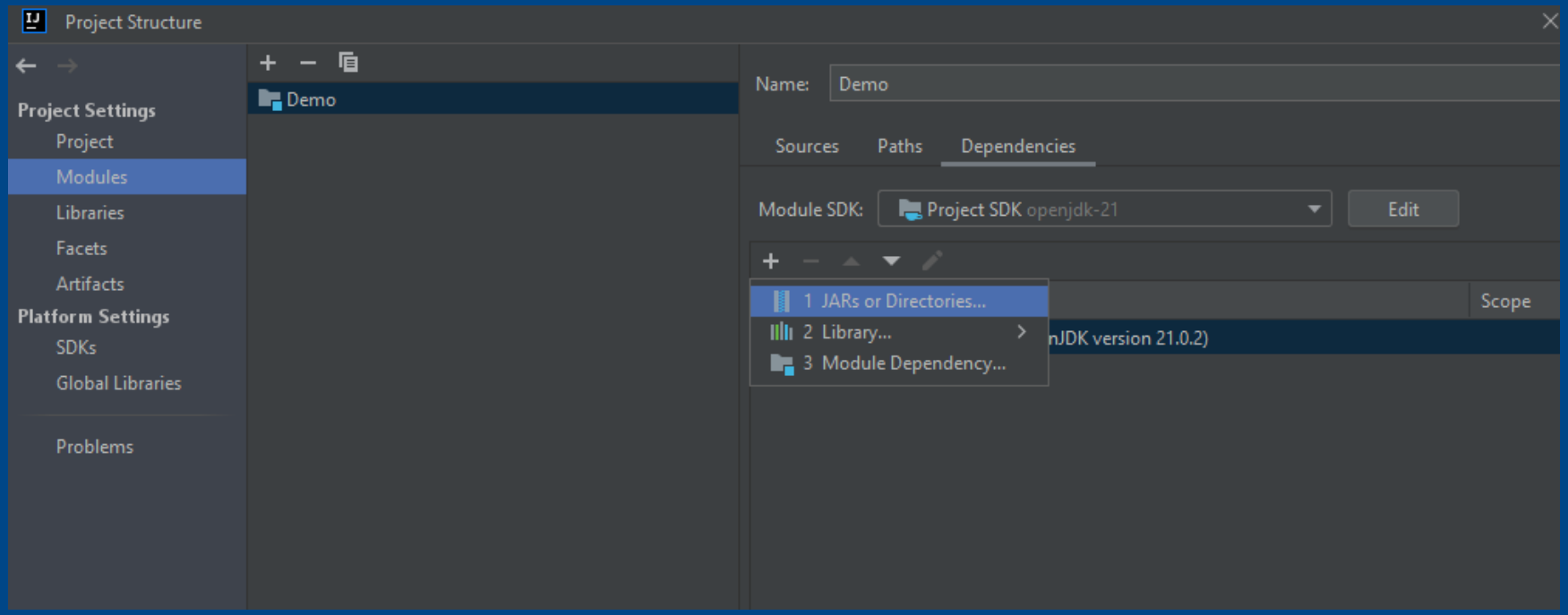
- File > New > Project
- Give a name and select a JDK



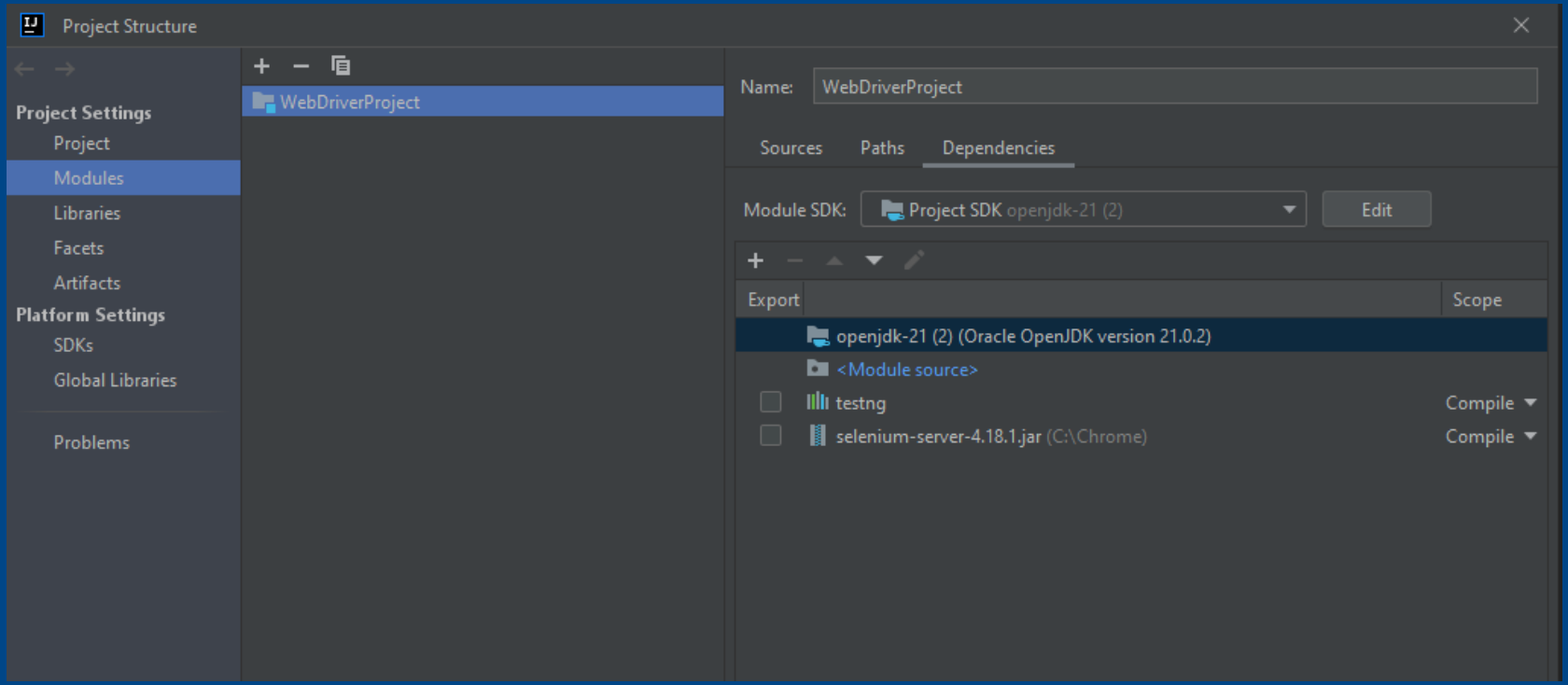
interworks.cloud™

Open IntelliJ and click

- File > Project Structure
- Go to Modules
- Add the Selenium Server (Grid) Jar file



How your installation should look like.



Introduction to TestNG

interworks.cloud™

Why do we need a test framework?

- Structured Testing Approach
- Test Case Management
- Parameterized Testing
- Parallel Test Execution
- Reporting and Logging

TestNG is a testing framework inspired from JUnit and NUnit but introducing some new functionalities that make it more powerful and easier to use.
https://testng.org/

- Annotations

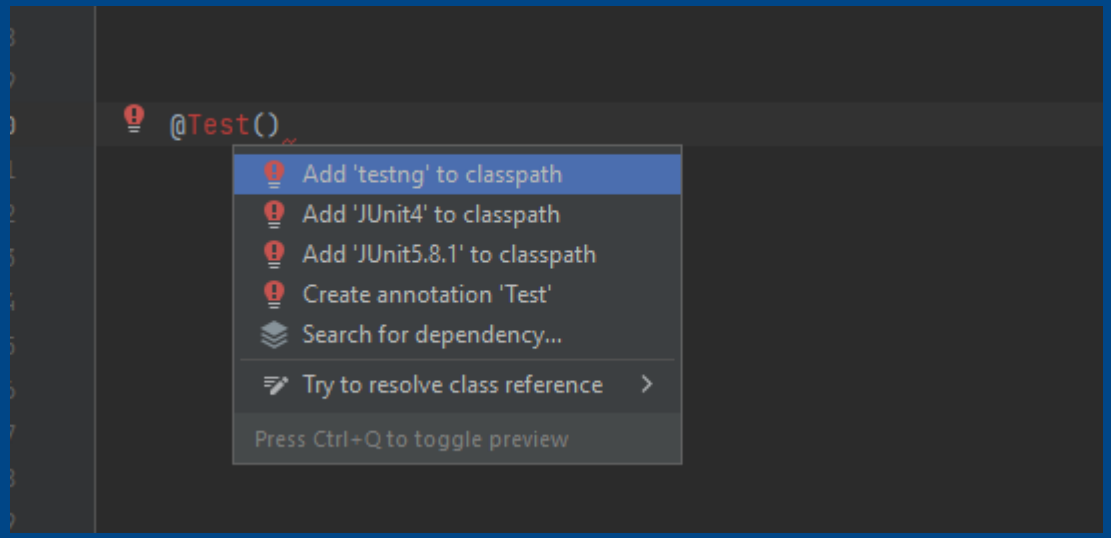
```
@BeforeTest
public void Test_Setup() throws Exception {
    System.out.println("--- Open Firefox ---");
}

@Test
public void Test() {
    System.out.println("--- Navigate to a page ---");
}

@AfterTest
public void Test_Teardown() throws Exception {
    System.out.println("--- Close firefox ---");
}
```

How to add TestNg

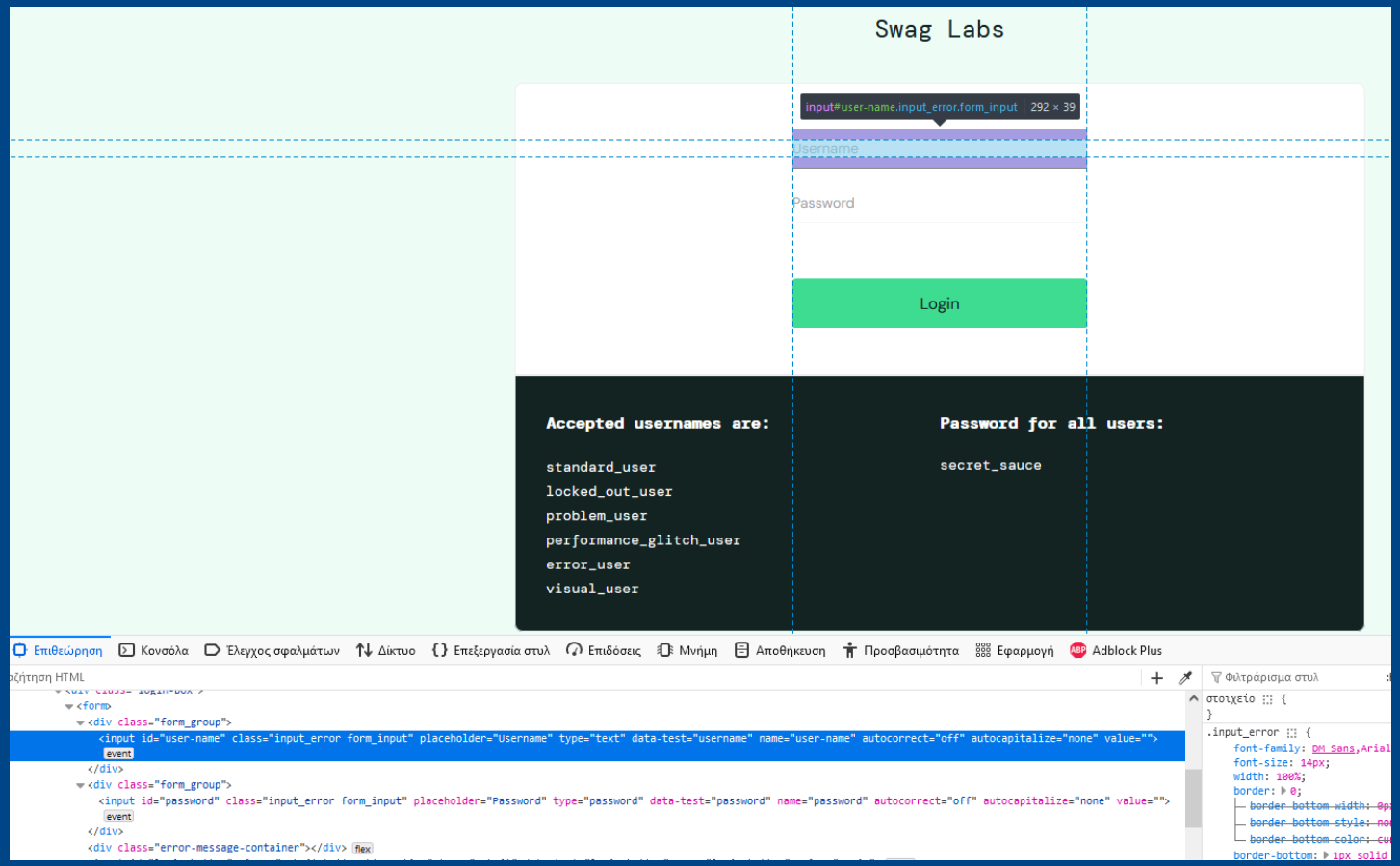
- Type @Test
- Click on Test and select show Context Actions
- Select the testng and it will be installed.



Introduction to UI Testing

The Document Object Model (DOM)

Dom is a programming interface for web documents,
that represents the structure of a webpage as a tree-like structure of nodes



interworks.cloud™

Basic Locators

- ID
- Name
- Class
- CSS Selector
- Xpath

- **Finding an element by Id**
- **Java code**

driver.findElement(By.id("user-name"));

```
<div class="login_wrapper">
  <div class="login_wrapper-inner">  grid
    <div id="login_button_container" class="form_column">
      <div class="login-box">
        <form>
          <div class="form_group">
            <input class="input_error form_input" placeholder="Username" type="tex
            t" data-test="username" id="user-name" name="user-name" autocorrect="o
            ff" autocapitalize="none" value> == $0
          </div>
          <div class="form_group">...</div>
          <div class="error-message-container"></div>  flex
          <input type="submit" class="submit-button btn_action" data-test="login-b
          utton" id="login-button" name="login-button" value="Login">
        </form>
      </div>
    </div>
  </div>
<div class="login_credentials_wrap">...</div>
```

- **Finding an element by Name**
- **Java code**

driver.findElement(By.name("login-button"));

- **Finding an element by Class**
- **Java code**

driver.findElement(By.className("btn_action"));

```
<div class="login-box">
  <form>
    <div class="form_group">…</div>
    <div class="form_group">
      <input class="input_error form_input" placeholder="Password" type="pas
      sword" data-test="password" id="password" name="password" autocorrect=
      "off" autocapitalize="none" value data-lmo-id="exvWYMcAWI">
    </div>
    <div class="error-message-container"></div> flex
    <input type="submit" class="submit-button btn_action" data-test="login-b
    utton" id="login-button" name="login-button" value="Login"> == $0
  </form>
</div>
</div>
</div>
<div class="login_credentials_wrap">…</div>
</div>
</div>
</div>
<script>…</script>
<script src="/static/js/2.9b02e67e.chunk.js"></script>
<script src="/static/js/main.9735b7ab.chunk.js"></script>
</body>
</html>
```

- **Finding an element by Css Selector**
- **CSS Selector: tagname[attribute='value']**
- **Java code**

driver.findElement(By.cssSelector("button[data-test*='backpack']"));

```
▶ <div id="header_container" class="header_container">…</div> flex
▼ <div id="inventory_container">
  ▼ <div>
    ▼ <div id="inventory_container" class="inventory_container">
      ▼ <div class="inventory_list"> flex
        ▼ <div class="inventory_item"> flex
          ▶ <div class="inventory_item_img">…</div>
          ▼ <div class="inventory_item_description"> flex
            ▶ <div class="inventory_item_label">…</div>
            ▼ <div class="pricebar"> flex
              ▶ <div class="inventory_item_price">…</div>
…             <button class="btn_primary btn_small btn_inventory " data-
              test="add-to-cart-sauce-labs-backpack" id="add-to-cart-sauce-lab
              s-backpack" name="add-to-cart-sauce-labs-backpack">Add to cart
              </button> == $0
            </div>
          </div>
        </div>
        ▶ <div class="inventory_item">…</div> flex
        ▶ <div class="inventory_item">…</div> flex
        ▶ <div class="inventory_item">…</div> flex
        ▶ <div class="inventory_item">…</div> flex
```

- **Finding an element by Xpath**
- **XPATH= //tagname[@attribute='value']**
- **Java code**

```
driver.findElement(By.xpath("//button[@data-test*='backpack']"));
```

Introduction to Selenium Webdriver

interworks.cloud™

What is Selenium Webdriver?

Selenium WebDriver is a powerful automation tool used for controlling web browsers, allowing testers and developers to interact with web applications programmatically.

Selenium WebDriver

How to initialize Webdriver

```
String geckoLocation = "drivers/geckodriver.exe";
System.setProperty("webdriver.gecko.driver", geckoLocation);
WebDriver driver = new FirefoxDriver();
```

interworks.cloud™

How to find a web element and interact with it

```
WebElement username = driver.findElement(By.id("user-name"));
username.sendKeys( ...keysToSend: "Manolis");

WebElement submitButton = driver.findElement(By.id("submit-btn"));
submitButton.click();

WebElement errorMessage = driver.findElement(By.id("error-msg"));
String messageText = errorMessage.getText();
```

interworks.cloud™

Custom Wait methods with find elements

```
List<WebElement> elements = new ArrayList<>();
while (elements.isEmpty()) {
    elements = driver.findElements(By.xpath( xpathExpression: "//input[@id='username']"));
    Thread.sleep( millis: 1000); // Sleep for 1 second before retrying
}
```

Custom Wait methods with WebDriverWait

```
// Define custom wait condition for an element to be visible
WebDriverWait wait = new WebDriverWait(driver, timeout: 10); // Wait up to 10 seconds
new *
WebElement element = wait.until(new ExpectedCondition<WebElement>() {
    new *
    public WebElement apply(WebDriver driver) {
        return driver.findElement(By.xpath( xpathExpression: "//input[@id='username']"));
    }
});
```

Introduction to Page Object Model

Page Object Model

Is a design pattern in Selenium that creates an object repository for storing all web elements.

What is the problem page object model tries to solve?

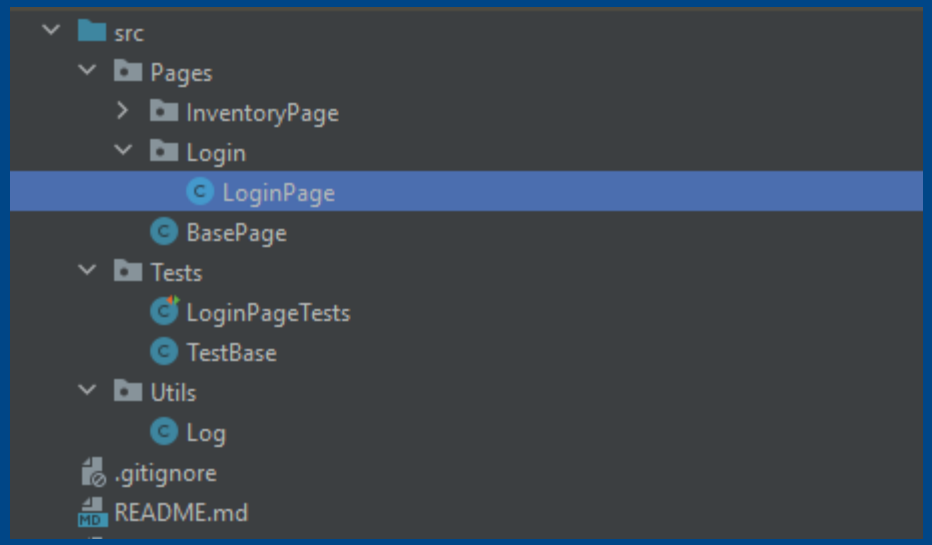
```
WebElement username = driver.findElement(By.id("user-name"));
username.sendKeys( ...keysToSend: "Manolis");

WebElement submitButton = driver.findElement(By.id("submit-btn"));
submitButton.click();

WebElement errorMessage = driver.findElement(By.id("error-msg"));
String messageText = errorMessage.getText();
```

interworks.cloud™

Better File Structure
Divide and conquer



interworks.cloud™

Lets see this POM
Under the hood

```
5 usages   Manolis Theodoroudis
public class LoginPage extends BasePage {

    2 usages   Manolis Theodoroudis
    public LoginPage(WebDriver driver) { super(driver); }

    2 usages
    public String loginPageUrl = "https://www.saucedemo.com/";
    1 usage
    @FindBy(id = "user-name")
    private WebElement userNameInput;
    1 usage
    @FindBy(id = "password")
    private WebElement passwordInput;
    1 usage
    @FindBy(id = "login-button")
    private WebElement loginButton;
    2 usages
    @FindBy(css = "h3[data-test='error']")
    private WebElement errorMessage;

    2 usages   Manolis Theodoroudis
    public void Login(String userName, String password) {
        driver.get(loginPageUrl);
        driver.navigate().refresh();
        userNameInput.sendKeys(userName);
        passwordInput.sendKeys(password);
        loginButton.click();
    }
```

interworks.cloud™

POM advantages

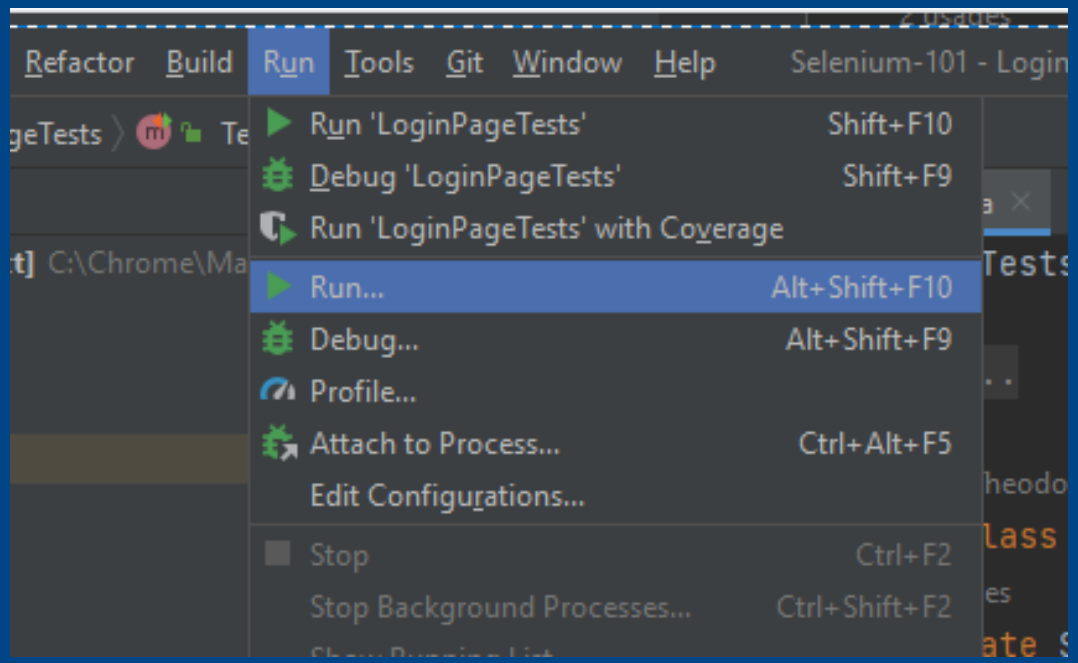
- Easy Maintenance
- Code Reusability
- Readability

How to run a Test

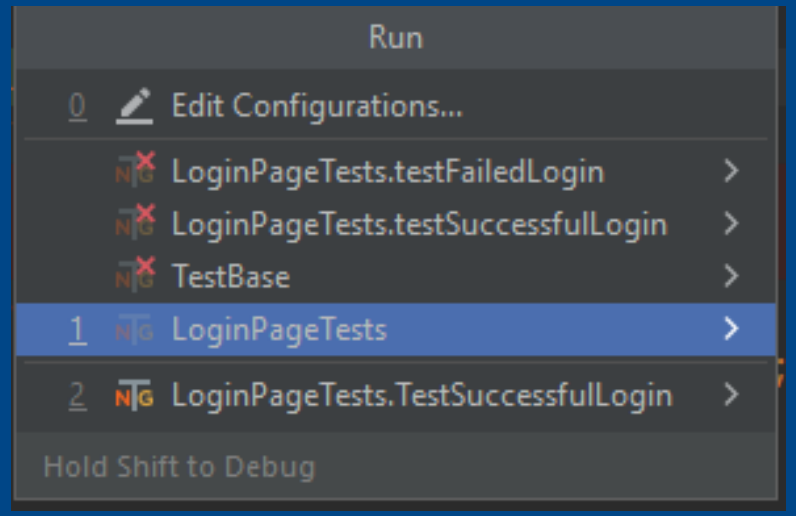
interworks.cloud™

Go to Run Tab.

Click Run..



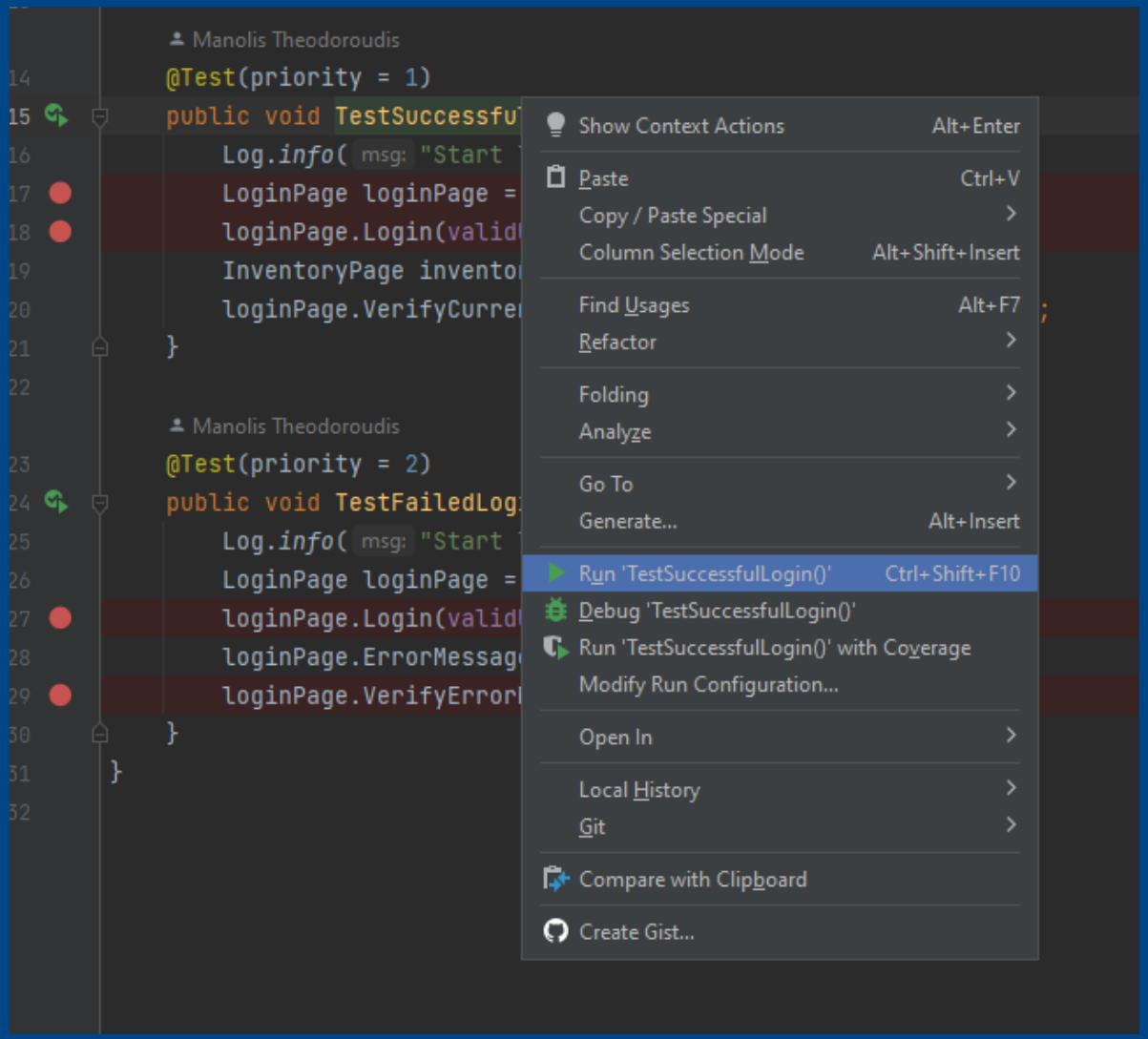
Select Test to run



interworks.cloud™

Go to test case

Click run

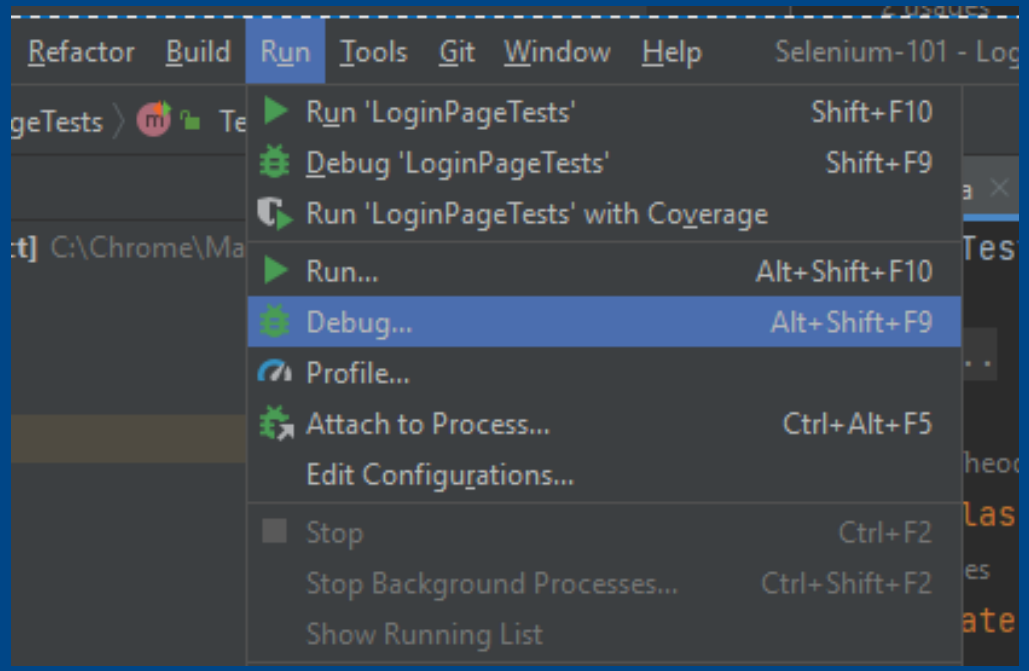


How to debug a Test

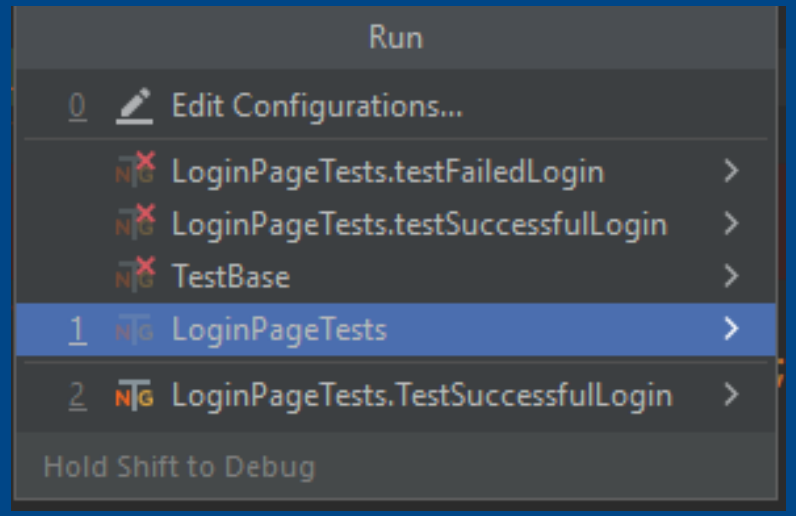
interworks.cloud™

Go to Run Tab.

Click Debug..



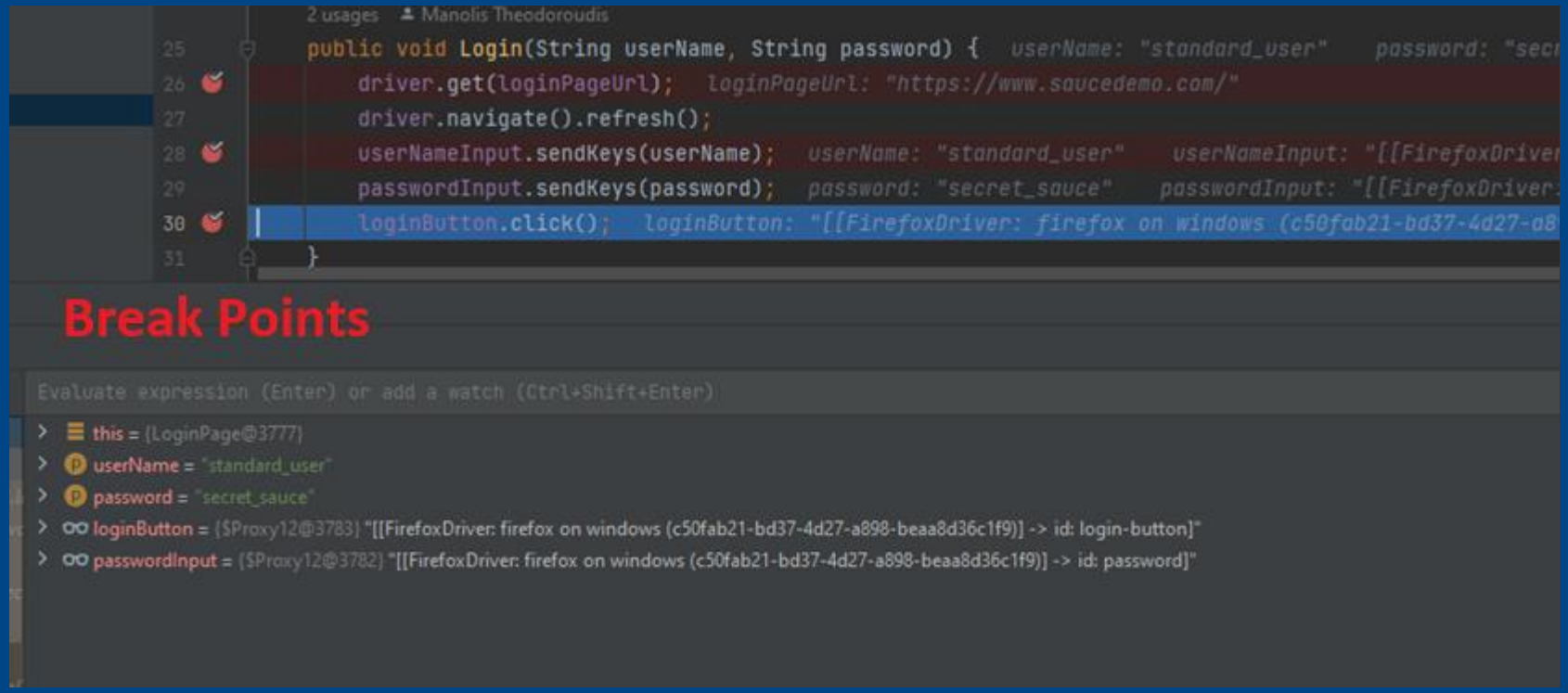
Select Test to Debug



interworks.cloud™

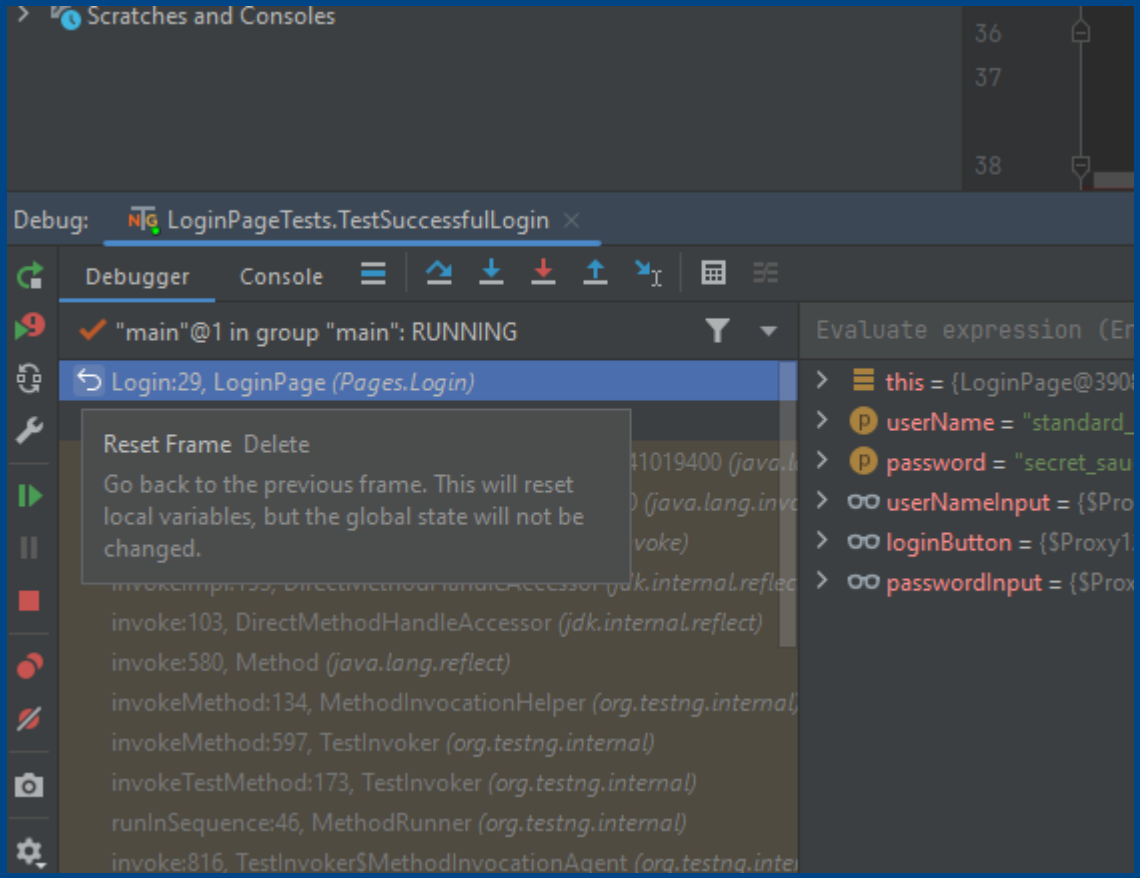
Debug Mode with break points and shortcuts

- **F9 Resume**
- **F8 Step over**
- **F7 Step into**



Reset Frame

Go back to the start of the method execution

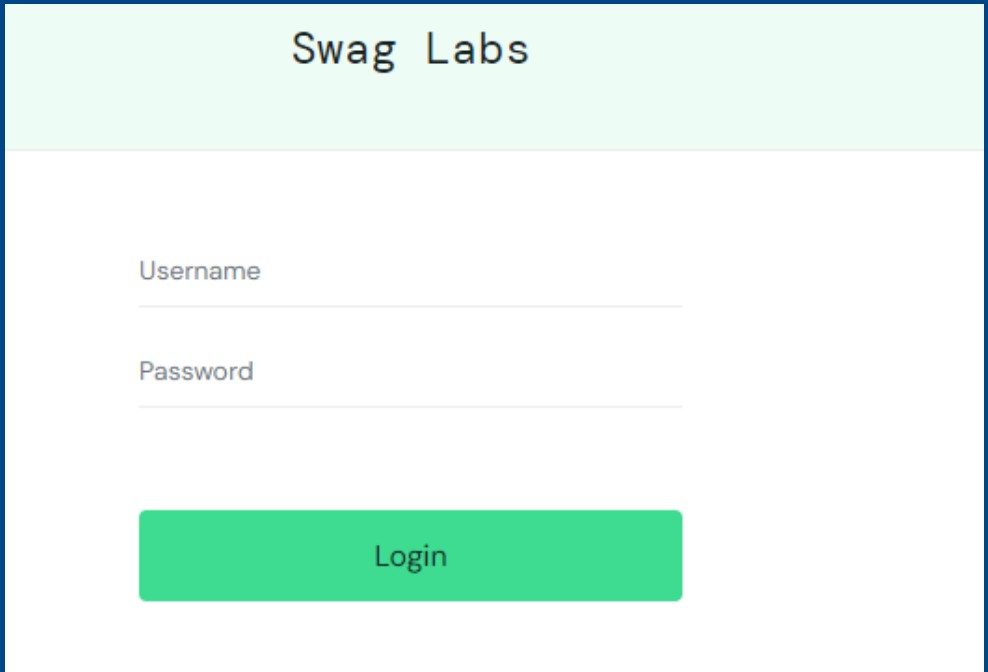


Demo

interworks.cloud™

Demo project having 2 test cases

- **TestSuccessfulLogin**
- **TestFailedLogin**



interworks.cloud™

Git Repo for the Demo.

https://github.com/ManolisTheodoroudis/Selenium-101

Download a Git Bash emulator

- **https://gitforwindows.org/**

Open emulator in a folder in your system and give the below command

Git Commands

- **git clone https://github.com/ManolisTheodoroudis/Selenium-101**

Readme.md has basic instructions to work

What we learned

- Learned Basic Concepts of Quality and Testing
- Learned how to make a basic test plan
- Learned how to use Webdriver Selenium to locate and use web elements in DOM.
- Learned how to setup a basic Testing framework in Java
- Learned what page object model is
- Learned how to run and debug a test.

Reading Material

- **Reading Material**
  - Site to test freely
    - https://www.saucedemo.com/
  - Selenium locators tutorial
    - https://www.guru99.com/locators-in-selenium.html
  - Selenium tutorial
    - https://www.tutorialspoint.com/selenium/index.htm
  - TestNG Documentation
    - https://www.tutorialspoint.com/software_testing/index.htm
  - TestNG Documentation
    - https://testng.org/
  - Git Tutorial
    - https://www.w3schools.com/git/

interworks.cloud™

Thank you!

Questions?

Manolis Theodoroudis