

Testing

Projekt: DigitalSchoolNotes

Projekt Team: Adler, Brinnich, Hohenwarter, Karic, Stedronsky

Version 1.1

15.12.2015

Status: [RELEASE]

| | Datum | Name | Unterschrift |
|---|------------|--------------------|--------------|
| Erstellt | 24.10.2015 | Niklas Hohenwarter | |
| Geprüft | 15.12.15 | Niklas Hohenwarter | |
| Freigegeben | | | |
| Git-Pfad: /doc/technologien Dokument: testing_technologie.doc | | | |

Inhaltsverzeichnis

| | | |
|---|-----------------------------|---|
| 1 | CHANGELOG..... | 3 |
| 2 | PROJEKT SETUP..... | 4 |
| 3 | ERSTELLEN VON TESTS | 5 |
| 4 | ABBILDUNGSVERZEICHNIS | 5 |
| 5 | CODEVERZEICHNIS | 5 |
| 6 | QUELLEN..... | 5 |

1 Changelog

| Version | Datum | Status | Bearbeiter | Kommentar |
|---------|------------|------------|--------------------|-----------------------------|
| 0.1 | 2015-10-24 | Erstellt | Niklas Hohenwarter | Draft |
| 1.0 | 2015-11-04 | Geprüft | Thomas Stedronsky | Rechtschreibfehler |
| 1.1 | 2015-12-15 | Bearbeitet | Selina Brinnich | Codeverzeichnis hinzugefügt |
| 2.0 | 2015-12-15 | Geprüft | Niklas Hohenwarter | QA |

2 Projekt Setup

Nachdem sich komplett automatisiertes Testing mittels Jenkins nur schwer realisieren lässt, testen wir nun auf unseren Entwicklungsmaschinen. Hierfür verwenden wir Selenium.

Es wird vorausgesetzt, dass IntelliJ bereits auf dem PC installiert ist.

- Erstellen eines neuen Projektes
- Herunterladen von Selenium für Java und Selenium Server von der offiziellen Webseite[1]
- Entpacken der ZIPs
- Hinzufügen aller JARs als externe Library zum IntelliJ Projekt
 - STRG+SHIFT+ALT+S
 - Modules>Libraries>Add>Java
 - Nun die JAR Files auswählen

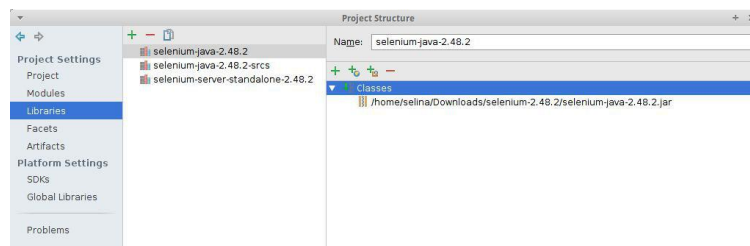


Figure 1: Screenshot IntelliJ Dependencies

- Run>Edit Configurations...

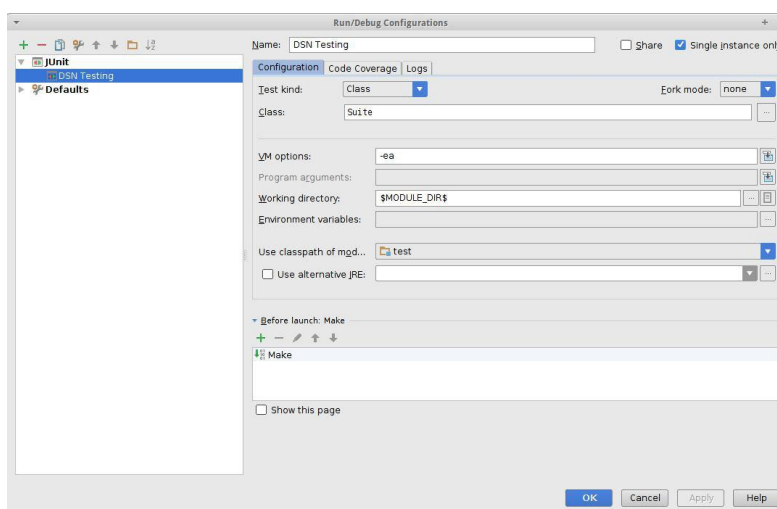


Figure 2: Run Configuration

3 Erstellen von Tests

Um Tests zu erstellen muss ein neues Java File erstellt werden. Als Beispiel erstellen wir nun einen Test, welcher die Hauptseite des Projektes aufruft.

```
import junit.framework.TestCase;
import org.junit.*;
import org.openqa.selenium.*;
public class TestSample extends TestCase{
    private WebDriver driver;
    private String baseUrl;
    private boolean acceptNextAlert = true;
    private StringBuffer verificationErrors = new StringBuffer();
    @Before
    public void setUp() throws Exception { // Startet Browser
        Parameters.setUpBrowser();
        this.driver = Parameters.driver;
        this.baseUrl = Parameters.baseUrl;
    }
    @Test
    public void testLoginUser() throws Exception { // Öffnet die Hauptseite
        driver.get(baseUrl + "/");
    }
    @After
    public void tearDown() throws Exception { // Beendet den Browser
        driver.quit();
        String verificationErrorString = verificationErrors.toString();
        if (!"".equals(verificationErrorString)) {
            fail(verificationErrorString);
        }
    }
    [...]
}
```

Code 1: Beispiel eines Selenium Test-Cases

Sobald der Test beendet wurde, werden die Ergebnisse angezeigt.

4 Abbildungsverzeichnis

| | |
|---|---|
| Figure 1 Screenshot IntelliJ Dependencies | 4 |
| Figure 2 Run Configuration | 4 |

5 Codeverzeichnis

| | |
|--|---|
| Code 1: Beispiel eines Selenium Test-Cases | 5 |
|--|---|

6 Quellen

[1], SeleniumHQ, "Downloads",
<http://docs.seleniumhq.org/download/>, zuletzt besucht: 24.10.2015