www.qatestlab.com
contact@qa-testlab.com

Selenium Script for
PROJECT A

General information	
Customer	<Project name>
Created by (Author)	
Preparation date	
Version	
Status	

Revision History					
Version	Description	Author	Date	Approved by	
				Author	Date

Summary

Selenium Test Automation

1. General information

The sample is a Maven project. The project has the following dependencies implemented: Selenium WebDriver (framework for test automation), TestNG (framework for testing), Allure (framework for report generation), Html Elements (framework for the description of pages block structure)

2. Project Configuration

The required dependencies should be added to the pom.xml:

```
<properties>
        <selenium.version>3.14.0</selenium.version>
        <testng.version>6.14.3</testng.version>
        <htmlelement.version>1.8</htmlelement.version>
        <allure.version>2.8</allure.version>
        <aspectj.version>1.9.5</aspectj.version>
</properties>

<dependencies>
  <dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>${selenium.version}</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.testng</groupId>
    <artifactId>testng</artifactId>
    <version>${testng.version}</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>ru.yandex.qatools.htmlelements</groupId>
    <artifactId>htmlelements-java</artifactId>
    <version>${htmlelement.version}</version>
  </dependency>
  <dependency>
    <groupId>ru.yandex.qatools.allure</groupId>
```

```
    <artifactId>allure-testng-adaptor</artifactId>
    <version>${allure.version}</version>
  </dependency>
  <dependency>
    <groupId>org.aspectj</groupId>
    <artifactId>aspectjweaver</artifactId>
    <version>${aspectj.version}</version>
  </dependency>
</dependencies>
```

Also, the Maven Surefire plugin should be configured. Specify the test classes required for execution. For report generating, plug Allure:

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-surefire-plugin</artifactId>
  <version>2.22.2</version>
  <configuration>
    <properties>
      <property>
        <name>delegateCommandSystemProperties</name>
        <value>true</value>
      </property>
      <property>
        <name>haltOnfailure</name>
        <value>false</value>
      </property>
      <property>
        <name>usedefaultlisteners</name>
        <value>false</value>
      </property>
    </properties>
    <systemProperties>
      <property>
        <name>allure.results.directory</name>
        <value>${project.build.directory}/allure-results</value>
      </property>
    </systemProperties>
    <suiteXmlFiles>
      <suiteXmlFile>${project.build.testOutputDirectory}/testng.xml</suiteXmlFile>
    </suiteXmlFiles>
    <argLine>-
```

```
        javaagent:${settings.localRepository}/org/aspectj/aspectjweaver/${aspectj.version}/aspectjweaver-
        ${aspectj.version}.jar
      </argLine>
    </configuration>
  </plugin>
```

A test suite is described in testng.xml:

```
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd" >
<suite name="Simple test suite" verbose="1" >
  <test name="Doodles Archive Page Functionality" >
    <classes>
      <class name="com.qatestlab.automation.demodoodle.tests.DoodlesArchivePageTests"/>
    </classes>
  </test>
</suite>
```

3. Test Scripts Design. Project Structure

The pattern PageFactory is used in the project. It enables us to specify the work with the website under test by using pages classes the elements of which will be automatically initialized.

Also, do not forget to annotate pages methods using @Step. The annotation specifies to the Allure framework what methods calls should be interpreted as scenario steps and displayed in reports.

Description of Google Main Page:

```
package com.qatestlab.automation.support.pages;
import org.openqa.selenium.support.FindAll;
import org.openqa.selenium.support.FindBy;
import ru.yandex.qatools.allure.annotations.Step;
import ru.yandex.qatools.htmlelements.element.Button;

public class MainPage extends BasePage {
  @FindAll({
        @FindBy(css = "input[jsaction='sf.lck']"),
```

```
        @FindBy(css = "input[onclick*=doodles]")
    })
    Button iFeelLuckyButton;

    @Step("Open main page")
    public MainPage open() {
        open("https://google.com");
        return this;
    }
    @Step
    public void clickIFeelLuckyButton() {
        iFeelLuckyButton.click();
    }
}
```

Description of Doodles Archive Page:

```
package com.qatestlab.automation.support.pages;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import ru.yandex.qatools.allure.annotations.Step;

public class DoodlesArchivePage extends BasePage {
    @FindBy(id = "loading")
    WebElement loader;
    List<DoodleWebElement> doodleCards;

    @Step("Open doodles page")
    public DoodlesArchivePage open() {
        open("https://google.com/doodles");
        return this;
    }

    @Step
    public DoodlesArchivePage loadNewArchiveDoodles() {
        scrollPageDown();
        driverWait.waitForElementInvisibility(loader);
        return this;
    }

    @Step
```

```
  public int getArchiveDoodlesAmount() {
    return doodleCards.size();
  }
}
```

Based on the sample provided above, to describe the Archive Page, the block DoodleWebElement is used. It is responsible for the specification of one block on the Doodle Page.

Blocks description and use of the basic elements, e.g., buttons or fields, are conducted using the framework Html Elements. The framework enables us to specify the repeated blocks as separate classes and use them in page descriptions.

Description of our element DoodleWebElement:

```
import org.openqa.selenium.support.FindBy;
import ru.yandex.qatools.allure.annotations.Step;
import ru.yandex.qatools.htmlelements.annotations.Name;
import ru.yandex.qatools.htmlelements.element.HtmlElement;
import ru.yandex.qatools.htmlelements.element.TextBlock;

@Name("Doodle card block")
@FindBy(css = "#archive-list .doodle-thumb")
public class DoodleWebElement extends HtmlElement {
  @FindBy(className = "title")
  private TextBlock titleLabel;

  @Step
  public String getTitle() {
    return titleLabel.getText();
  }
}
```

During pages description, the basic class BasePage is used. It includes the core functionality for all the pages and is responsible for proper initialization of pages samples (initialization of web elements):

```
import com.qatestlab.automation.demodoodle.tests.DoodlesArchivePageTests;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.PageFactory;
```

```
import org.openqa.selenium.support.ui.WebDriverWait;
import ru.yandex.qatools.allure.annotations.Step;
import ru.yandex.qatools.htmlelements.loader.decorator.HtmlElementDecorator;
import ru.yandex.qatools.htmlelements.loader.decorator.HtmlElementLocatorFactory;

public abstract class BasePage {
  private WebDriver driver;
  protected WebDriverWait driverWait;
  private JavascriptExecutor jsExecutor;

  public BasePage() {
     driver = DoodlesArchivePageTests.driver;
     driverWait = new WebDriverWait(driver, 10);
     jsExecutor = new JavascriptExecutorLogged(driver);
     PageFactory.initElements(new HtmlElementDecorator(new HtmlElementLocatorFactory(driver)),
           this);
  }

  @Step
  protected void open(String url) {
     driver.navigate().to(url);
  }

  @Step
  protected void scrollPageDown() {
     jsExecutor.executeScript("window.scrollBy(0,document.documentElement.scrollHeight)");
  }
}
```

After describing all the required pages, it is time to develop test scripts for interaction with our pages and to check the behavior of the website under test:

```
import java.io.File;
import java.util.concurrent.TimeUnit;

import com.qatestlab.automation.support.pages.DoodlesArchivePage;
import com.qatestlab.automation.support.pages.MainPage;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.Assert;
import org.testng.annotations.AfterClass;
import org.testng.annotations.BeforeClass;
```

```
import org.testng.annotations.Listeners;
import org.testng.annotations.Test;
import ru.yandex.qatools.allure.testng.AllureTestListener;

@Listeners({AllureTestListener.class})
public class DoodlesArchivePageTests {
  public static WebDriver driver;

  @BeforeClass
  public void setUp() {
    System.setProperty(
        "webdriver.chrome.driver",
        new File(DriverFactory.class.getResource("/chromedriver.exe").getFile()).getPath());
    driver = new ChromeDriver();
    driver.manage().timeouts().implicitlyWait(15, TimeUnit.SECONDS);
    driver.manage().timeouts().pageLoadTimeout(30, TimeUnit.SECONDS);
    driver.manage().window().maximize();
  }

  @AfterClass
  public void tearDown() {
    if (driver != null) {
      driver.quit();
    }
  }

  @Test
  void navigateToDoodlePageTest() {
    new MainPage()
        .open()
        .clickIFeelLuckyButton();
    Assert.assertTrue(
        driver.getCurrentUrl().endsWith("/doodles"),
        "Doodles page is not opened.");
  }

  @Test
  void loadMoreArchiveDoodlesTest() {
    DoodlesArchivePage page = new DoodlesArchivePage().open();
    int doodlesBefore = page.getArchiveDoodlesAmount();
    page.loadNewArchiveDoodles();
```

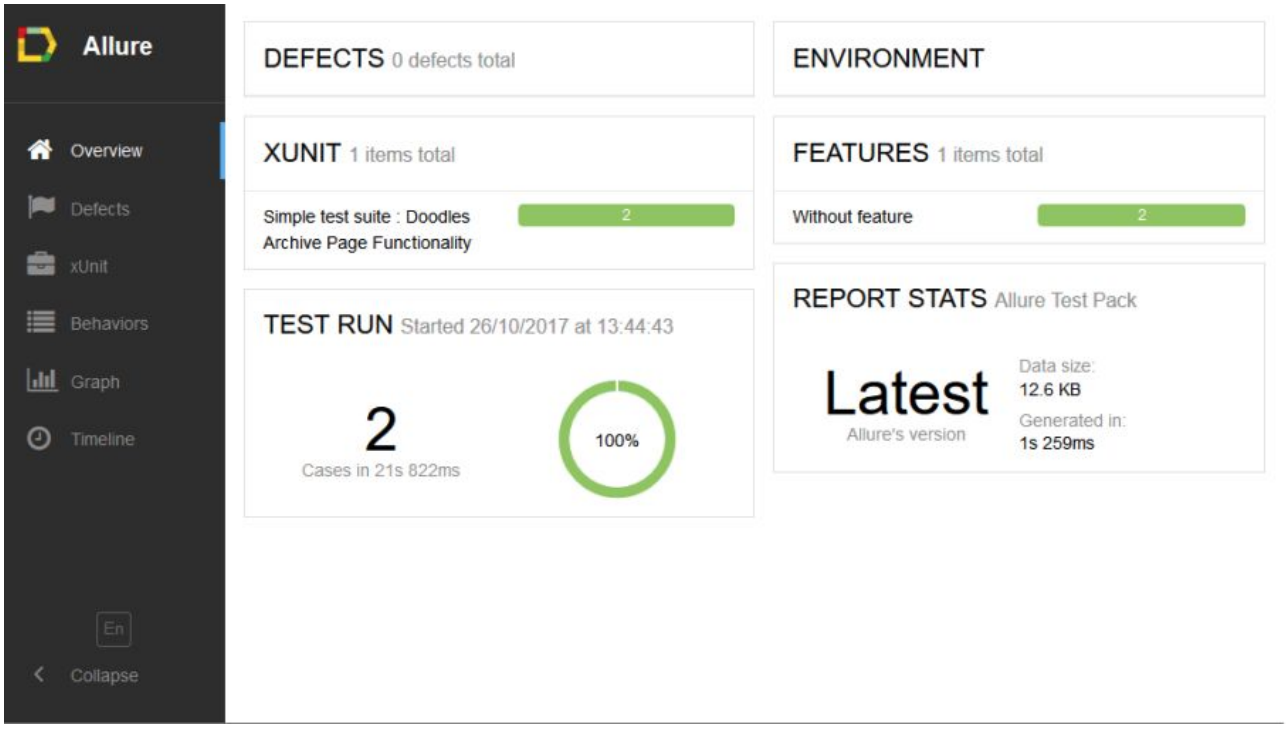
```
        Assert.assertTrue(
                doodlesBefore < page.getArchiveDoodlesAmount(),
                "Doodles are not loaded after scrolling.");
    }
}
```

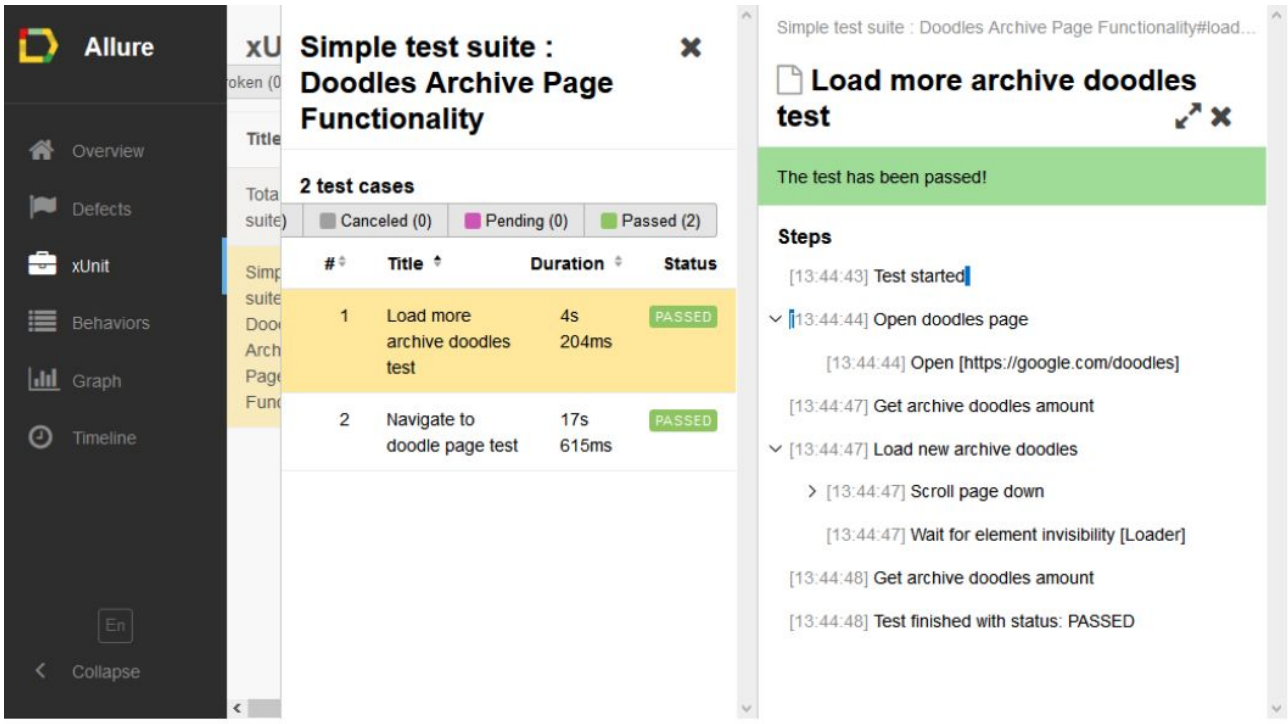
To describe test classes, TestNG annotations are used. Also, the listener of Allure framework is connected to have the required report generated after test execution

4. Test Execution and Result Generation

Tests can be executed by calling a test phase in Maven. And the tests will be called in such a way as we have configured at the first stage.

After test execution, a user has Allure reports available. They can contain a lot of additional information added to a list of performed steps. For example, in case of error, the option of a screenshot and saved web page attachment is available.





The example of a test report with detected issue (the error of checking page functionality, scrolling does not work):