**Document complete instructions to execute automated tests**

Selenium Architecture

Selenium has a client-server architecture, and includes both client and server components.

Selenium Client includes:

* The WebDriver API, which you use to develop test scripts to interact with page and application elements
* The **RemoteWebDriver** class, which communicates with a remote Selenium server

Selenium Server includes:

* A server component, to receive requests from Selenium Client 's RemoteWebDriver class
* The WebDriver API, to run tests against web browsers on a server machine
* Selenium Grid, implemented by Selenium Server in command-line options for grid features, including a central hub and nodes for various environments and desired browser capabilities

There are **seven basic elements** of a Selenium test script, which apply to any test case and any application under test (AUT):

1. Create a WebDriver instance.
2. Navigate to a Web page.
3. Locate an HTML element on the Web page.
4. Perform an action on an HTML element.
5. Anticipate the browser response to the action.
6. Run tests and record test results using a test framework.
7. Conclude the test.

The following example includes code for all steps. The example also defines a Java test class Example, and its **main** method, so that the code can be run.

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.By;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.WebDriverWait;

import org.junit.Assert;

public class Example  {

  public static void main(String[] args) {

    // Create an instance of the driver

    WebDriver driver = new FirefoxDriver();

    // Navigate to a web page

    driver.get("[http://www.foo.com](http://www.foo.com/)");

    // Perform actions on HTML elements, entering text and submitting the form

    WebElement usernameElement     = driver.findElement(By.name("username"));

    WebElement passwordElement     = driver.findElement(By.name("password"));

    WebElement formElement        = driver.findElement(By.id("loginForm"));

    usernameElement.sendKeys("Alan Smithee");

    passwordElement.sendKeys("twilightZone");

    //passwordElement.submit(); // submit by text input element

    formElement.submit();        // submit by form element

    // Anticipate web browser response, with an explicit wait

    WebDriverWait wait = new WebDriverWait(driver, 10);

    WebElement messageElement = wait.until(

           ExpectedConditions.presenceOfElementLocated(By.id("loginResponse"))

            );

    // Run a test

    String message                 = messageElement.getText();

    String successMsg             = "Welcome to foo. You logged in successfully.";

    Assert.assertEquals (message, successMsg);

    // Conclude a test

    driver.quit();

  }

}