v

Neha Sureshkumar

Feb 2025

This document outlines about Selenium testing tool

SELENIUM - overview

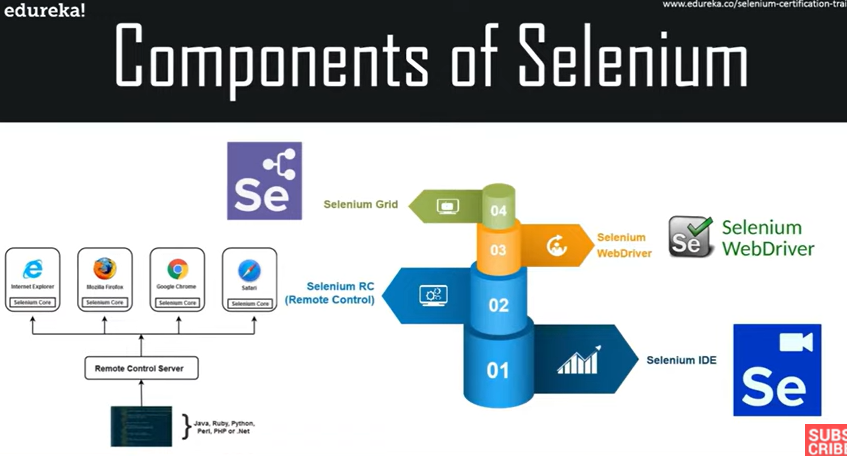
Contents

|  |
| --- |
|  |
| S. No | Topics | Page Number |
| 1 | Architecture Overview | 2 |
| 2 | Normalized Referential Database | 2 |
| 3 | Database Linking – Equal Join | 3 |
| 4 | Database Linking – Outer Join | 5 |
| 5 | Data Presentation | 7 |
| 6 | Database Views | 7 |
| 7 | Online help | 8 |

Introduction

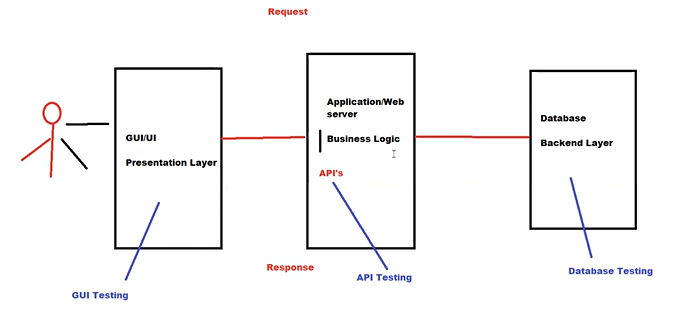
* [Selenium](https://www.geeksforgeeks.org/selenium-with-java-tutorial/)is a widely used tool for [testing web-based applications](https://www.geeksforgeeks.org/software-testing-web-based-testing/)that checks if they are doing as expected.
* It is a prominent preference amongst testers for [cross-browser testing](https://www.geeksforgeeks.org/why-cross-browser-testing-important/)and is viewed as one of the most reliable systems for [web application automation](https://www.geeksforgeeks.org/browser-automation-using-selenium/)evaluation.
* Selenium is also platform-independent, so it can provide distributed testing using the Selenium Network.
* Selenium is a powerful tool for controlling web browsers through programs and performing browser automation.
* It is functional for all browsers, works on all major OS and its scripts are written in various languages.

Components of Selenium



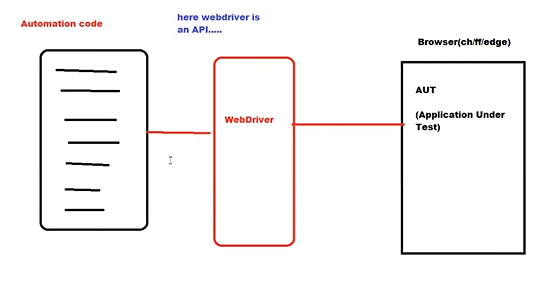
|  |  |
| --- | --- |
| S.No | Tool and Description |
| 1 | **Selenium IDE**  Selenium **I**ntegrated **D**evelopment **E**nvironment (IDE) is a Firefox plugin that lets testers to record their actions as they follow the workflow that they need to test. |
| 2 | **Selenium RC**  Selenium **R**emote **C**ontrol (RC) was the flagship testing framework that allowed more than simple browser actions and linear execution. It makes use of the full power of programming languages such as Java, C#, PHP, Python, Ruby and PERL to create more complex tests. |
| 3 | **Selenium WebDriver**  Selenium WebDriver is the successor to Selenium RC which sends commands directly to the browser and retrieves results. |
| 4 | **Selenium Grid**  Selenium Grid is a tool used to run parallel tests across different machines and different browsers simultaneously which results in minimized execution time. |

Structure of a web based Application



* In a higher level Architecture Every Web based application has three types of layers namely the Presentation layer (GUI), Application layer (Business logic), Database layer (Database)
* Also different types of testing has to be carried out like GUI testing, API testing and Database testing for each layer respectively.
* Selenium concentrates only on GUI testing, ie it can access the front end elements and then locate them during automation to perform testing

Selenium WebDriver



* A web driver is a module that contains many predefined classes and methods which can be used while writing a test script
* Here the Web driver acts as an API that connects Automation code with the browser (Application Under test)
* Selenium WebDriver is a tool for automating web browsers, primarily used for functional and UI testing of web applications.
* It allows you to write test scripts in multiple programming languages, including Java, Python, C#, JavaScript, and Ruby.
* WebDriver interacts with the browser directly, unlike Selenium RC, which relies on a server to communicate with the browser.
* Supports multiple browsers like Chrome, Firefox, Safari, and Edge, providing cross-browser compatibility.
* WebDriver provides APIs for interacting with web elements like buttons, links, forms, and inputs.
* It supports actions like clicking, typing, navigating, and scrolling, as well as complex interactions like drag-and-drop and mouse movements.
* WebDriver can handle dynamic web pages with JavaScript execution and AJAX calls.
* Selenium WebDriver supports both desktop and mobile browser automation through mobile emulation or integration with tools like Appium.
* It supports headless browser testing, allowing tests to be executed without opening a GUI browser window.
* WebDriver has built-in methods for waiting for elements to appear, either through implicit or explicit waits, ensuring synchronization between the browser and test actions.
* It allows for screenshot capture during tests, useful for visual validation or reporting.
* Selenium WebDriver is compatible with Continuous Integration (CI) tools like Jenkins, Bamboo, and TeamCity, enabling automated test execution as part of a CI pipeline.
* It integrates with various testing frameworks like TestNG, JUnit, and PyTest for organizing and running tests, along with generating detailed test reports.

Features of Selenium

Selenese Commands

* Actions − The action commands are used to manipulate the state of the application under test. For example, click, type, open, close, and so on.
* Accessors − The accessor commands are used to assess the state of the application under test and store it in variables. For example, store title, store text, and store value.
* Assertions − The assertion commands are used to check the state of the application under test. It has three sub-types namely, assert, verify, and waitfor.

Third party Plugins

* Many Third Party Plugins can be installed within the Selenium Integrated Development Environment to perform additional actions.

CI/CD Support

* The Selenium Integrated Development Environment tests can run from the side-runner (i.e from the command line) using the .side file of Selenium Integrated Development Environment test. This makes it easily configurable with the CI/CD tools like Jenkins, CircleCI, and so on.

Record and Playback

* Selenium Integrated Development Environment has the feature which allows recording of the user actions performed on the application under test. The recording of these user actions can be replayed and used as a test. If not required, the record and playback feature may not be used, and test scripts can be developed from scratch. Also, the test created by recording, can be modified as per the requirement.

Navigating links using get method

* The first thing one’ll want to do with WebDriver is navigate to a link. The normal way to do this is by calling get method:
* Syntax – driver.get(url)
* Example - driver.get("http://www.google.com")

Interacting with Webpage

* First of all, we need to find one. WebDriver offers several ways to find elements. For example, given an element defined as:

<input type="text" name="passwd" id="passwd-id" />

* To interact with elements in Selenium, various locating strategies can be used. For example, you can locate elements by ID, name, or XPath:

element = driver.find\_element(By.ID, "passwd-id")

element = driver.find\_element(By.NAME, "passwd")

element = driver.find\_element(By.XPATH, "//input[@id='passwd-id']")

* If you need to find multiple elements, use:

elements = driver.find\_elements(By.NAME, "passwd")

* To find a link by its text, ensure the text is an exact match:

element = driver.find\_element(By.LINK\_TEXT, "Link Text")

After Locating an Element – What’s Next?

If you want to input text into a field, you can use:

element.send\_keys("some text")

You can also simulate pressing arrow keys or other keys using the Keys class:

element.send\_keys(" and some", Keys.ARROW\_DOWN)

It’s worth noting that you can call send\_keys on any element, which makes it possible to test keyboard shortcuts, like those used in Gmail. To clear the contents of a text field or textarea, use the clear method:

element.clear()

Locating single elements

* Selenium Python follows different locating strategies for elements. For locating elements, you have to import By

from selenium.webdriver.common.by import By

|  |  |
| --- | --- |
| Locators | Description |
| By.ID | The first element with the id attribute value matching the location will be returned. |
| By.NAME | The first element with the name attribute value matching the location will be returned. |
| By.XPATH | The first element with the xpath syntax matching the location will be returned. |
| By.LINK\_TEXT | The first element with the link text value matching the location will be returned. |
| By.PARTIAL\_LINK\_TEST | The first element with the partial link text value matching the location will be returned. |
| By.TAG\_NAME | The first element with the given tag name will be returned. |