What is WebDriverIO?

Web Driver Input/Output

WebdriverIO is a **JavaScript testing framework** that acts as a **package** that you install via npm (Node Package Manager).

WebDriverIO is a popular open-source test automation framework that allows you to control web and mobile applications through the WebDriver protocol. It provides an easy-to-use JavaScript API for interacting with web elements, running tests, and integrating with various testing tools. WebDriverIO supports multiple browsers and platforms, including mobile testing through Appium and other services, making it a versatile tool for automating functional tests.

Note: Not to be confused with Selenium Webdriver, WebdriverIO is an independent implementation of Selenium Webdriver’s communication protocol. WebdriverIO is supported by all browsers and does not require you to download a modified browser that you would not actually use.

Protocol used by WebDriverIO is two types:

1. JSON Wire protocol (Out Dated version).
2. W3C WebDriver Protocol (Currently use because it is a new version of protocol).

Command to install WebDriverIO

Install Latest: npm install webdriverio@latest –save or npm init wdio@latest .

Install specific:

Install a specific path or your project: npm init wdio@latest ./path/to/new/project

Features of WebDriverIO

1. **Automation testing framework**: Enables browser and mobile testing.
2. **Free and Open-Source**: It's free to use and constantly evolving with contributions from the open-source community.
3. **Owned by OpenJS**: Managed under the OpenJS Foundation, which ensures its development and maintenance.

Note: **WebdriverIO** essentially refers to a tool that uses the **WebDriver** protocol to perform **Input/Output** operations with a web browser for automated testing.

**Workflow Example:**

1. **You write test scripts** in WebdriverIO using JavaScript.
2. The scripts define actions like tapping on buttons, entering text, or navigating between screens in a mobile app.
3. WebdriverIO sends these commands to the Appium server, specifying the device and the actions to perform.
4. Appium communicates with the mobile device (or simulator/emulator) to execute these actions.
5. The results are sent back to WebdriverIO, and your test can assert behaviors (e.g., checking if an element is visible, validating UI states).