WebDriverJS –Selenium Runner On Node with Promise Mechanism

Agenda Of the document: Introduction of WebDriverJS (**not WebDriver**)–A Selenium Runner on Node

What is WebDriverJS: A selenium Runner on NODE,Highly extendible,compatible with all common JavaScript Testing Framework,hugely used for Angular JS application.

WebDriverJS & WebDriver are Same:NO.
The above two are **related** but not the same.

How both are related but not the same:
A. WebDriverJS is the JavaScript Language bindings for WebDriver,It runs on NODE,It uses an own chain API to execute and handle **Asynchronous Calls** in specific and in right order.

B.In **Addition to NODE**, WebDriverJS may also be **used directly in the browser by JSonWire Protocol Communication** as the Other Language binding of WebDriver(i.e Java,Python,Ruby etc. etc.).One of The **Advantages** for running **WebDriverJS in the browser is that it can control the browser running the Script**,as long as The URL for the server and Session ID for the browser are known.

C.**Unlike other** language **bindings**,which all provide **blocking APIs**,**WebDriverJS** is purely **Asynchronous**,In order to track handle exceptions WebDriverJS makes extensive use of "**Promises**."
While there are various promise implementations are there for JS,the WebDriverJS promise is based on the proposal from **CommonJS**,which defines a promise as any object with a "**then**" function property.

Asynchronous Computation:Its a concept of **passing a callback function** that is called when the **result** becomes **available** at some **later time** ,rather than a function returning a result value immediately.

Example:
Browser Offers a primarily single thread runtime environment,so if you request your browser to fetch some information which takes 3 seconds,then the browser will not be able to do anything else than to wait for HTTP call.

So, **Asynchronous** call came to the picture to overcome this kind of scenarios.

Concept of Promise:The promise object is used for deferred and Asynchronous computation.

A promise is in one of three different states:

1.**Pending**: Promise's initial state.

2.**Fulfilled**: Representing promise's state of successful operation.

3.**Rejected**: Represents promises's failed operation.

How WebDriverJS works with Asynchronous calls:

WebDriverJS **uses a wrapper for Promise called ControlFlow Class**:

A.It maintains a list of scheduled actions

B.The exposed functions in WebDriverJS do not actually do their stuffs,they just push the required actions into ControlFlow Class.

Example:

With Simple WebDriver with Java we write :

```
driver.get("www.xyzabc.com");
driver.findElement(By.id("k1")).sendKeys("WebDriver");
```

Using **Promise we need to write** the same like:

```
driver.get("www.xyzabc.com").  then(function() {
   return driver.findElement(By.id("k1"));
 }).
 then(function(k1) {
   return k1.sendKeys("WebDriver");  }).
```

By this way WebDriverJS is related with WebDriver,uses Promise,JSON Wire Protocol and handles Asynchronous Calls very efficiently.