**Repository**

A repository is a centralized location to store UI elements (Web, Android, iOS, MobileWeb), program elements, and step groups in an organized manner, which are used in script development.

**Elements** - UI elements or components are the building blocks that constitute a website or a mobile application's interface.

Example : - buttons, text boxes, menus, forms, checkboxes, and radio buttons, among others.

**Why do we need Elements?**

FireFlink utilizes UI elements as building blocks for automation script development. This enables test engineers to create automation test scripts and execute

### The Process of Creating a Project Element

### Step – I : Creating Pages & Sub-Pages in the Repository

### Step – II : Steps to Create a Program Element

**Users can add UI elements to a page/sub-page in 3 ways:**

* FireFlink Finder
* Manually Adding the element
* Through Shared Elements

### Shared Elements

Used to share project elements across pages, sub-pages, screens, and subscreens. This feature ensures proper maintenance of UI elements within the project.

##### **Steps**

##### Project Selection > Repository > Elements > Shared Elements

**Test Development**

## **Step Groups**

Step groups are used to convert common automation steps into reusable components. By doing so, users can eliminate the redundancy of rewriting the same steps in multiple test scripts.

Step Groups can be created in repository and test development.

**Types of Step Group**

* With parameters and with return value.
* With parameters and without return value.
* Without parameters and with return value.
* Without parameters and without return value.

//step group variable

**How to create Step Group with Parameter?**

Go to Repository 🡪 Go to Step group tab 🡪 Create Step Group🡪 Click on Parameters and create the parameters here by selecting the datatypes.

//replace

**Locators**

Locators are used to locate the element accurately on the web page. Before performing any action on the element, we need to find or inspect the elements.

**Web Locators supported by FireFlink: –**

* id
* name
* className
* tagName
* linkText
* partialLinktext
* cssSelector
* xpath

**Pre And Post Condition**

A **"pre-condition"** is a set of **"Step groups (set of automation steps)"**, that are to be executed prior to the execution of the **"actual automation steps"** within the test script.

While **"post-conditions"** are, a designated collection of Step groups that are executed after the execution of the actual automation steps within an automation test script.

**Can be given in 3 different levels**

* Script level.
* Module level.
* Root module level.

**Note :** only step groups can be given as pre and post conditions.

Cascading is applicable only for module level and root module level

Cascading should be enabled if the pre and post condition should be applicable for all the scripts.

**Steps**

**Test Development🡪Automation Script🡪settings🡪pre-condition**

If pre condition is given at module level , first it will execute the pre condition then executes all the scripts. Pre condition is applicable for individual scripts in the module after enabling cascading radio button.

**Depends on Script**

"Depends on scripts" refers to a scenario where one script is dependent on another script.

**Steps**

**Test development🡪Automation Script🡪Settings🡪Add depends On Script.**

* For first script we cannot add depends on script.
* If dependent script has to be executed first base script should be executed and be passed.
* Depends on script is only possible with in the module itself.
* Sub module can be dependent on parent module but not on the parallel modules.

**When do we use Depends on Script?**

* //If any script has to be executed before the execution of a — "**Depends on script**" is used.

**Synchronization**

Matching the application loading time with the execution time.

**There are three types**

* Explicit wait
* Implicit wait
* Custom wait

**Explicit wait** : if the element is found before the specified wait time ,it will still wait till the completion of wait time and then continues the execution of script.

**Implicit wait :** If the element is found before the specified wait time ,it will not wait further and continues its execution.

**Data Provider**

**What is a Data provider?**

* Data Providers is a methodology set to pass data explicitly to the scripts.
* The data is stored in a data source file like an Excel sheet, property file, etc. and then it is used as input for a script during execution.

**Why do we need a Data provider?**

* Data providers are used to avoid hard coding.
* Instead of having hard-coded input, you have new data each time the script loads the data from the external data source.
* //
* If the user uses a Data provider, there is no need to Re-Compile the Script for data modification.
* Using a Data provider, data can be modified easily without intervening in the code.
* Maintenance of test data will be easy if the Data provider is used.

**Navigation to the data provider settings screen:**

**Test development**🡪 Automation Script 🡪**Settings** tab 🡪**Data provider**

**Variables**

**What is a Variable?**

Variables are the containers that are used to store the data, it can store different values

and be utilized within a script or a project which avoids data hardcoding.

**What is the use of a Variable?**

* Variables created are used to pass the value in the step, for the execution.
* It avoids the re-work when the value changes.
* To reuse the values at multiple places.

**Types of Variables:**

1. **Local Variables:** The local variables can be accessed only to those steps created under a

particular script.

1. **Global Variables:** The global variables can be accessed throughout the Project and it

can be used to store step returning value.

1. **Project Environment Variables:** The project environment variables can be accessed for

all the modules, sub-modules, and scripts present under that project.

**There are three ways to add variables:**

* In the steps screen, on clicking the "**More** " icon, the context menu displays. Select

the "**Add Variable**" option in the list, "**Create Variable**" pop-up appears.

* In the variables screen, on clicking the "**+Variable**" button, the "**Create Variable**" pop-up

displays. The "**Create Variable**" popup .

While adding step aswell we can create variable.

**Test Execution**

**Execution**

Execution is the workspace where a user **can create and execute Suites**(collection of modules and scripts).

Navigation for Execution

FireFlink > Left Navigation Panel > Execution

**Suite**

The suite is a **collection of modules, sub-modules, and scripts**. Suites are created to test a behavior or set of behaviors of an application.

**Navigation for Suite**

Fireflink > Left Navigation Panel > Execution > Suite

Two types of suites can be created

* Automation Suite.
* Manual suite.