**Extent Manager**

**ExtentHtmlReporter -** The ExtentHtmlReporter creates a rich standalone HTML file. It allows several configuration options via the code configuration method.

It is a library that can be used to build a customized detailed report.

It generates HTML report based on a path specified by the tester.

**Extent Reports –** Extent report is an open-source reporting library used to create visually attractive reports for Selenium tests using Junit and TestNG. Extent reports produce HTML-based documents that offer several advantages like pie charts, graphs, screenshots, and test summary.

* **Set System Info-**  It is used to set system information. There are basically Two parameters Name of system variable and Value of system variable.
* **Attach Reporter -** Attach a Extent Reporter reporter, allowing it to access all started tests, nodes and logs.

**Get Report Name –** It helps generating report name for the test using File Name and simple Date and Time format with the help of Selenium Helper.

**Create Instance -** It create report with given attributes like Document title, report name, report theme, system info etc.

* **Set Encoding(UTF-8) –** It is an encoding system for Unicode. It can translate any Unicode character to a matching unique binary string and can also translate the binary sting back to a Unicode.
* **Set Document Title-** It is used to set the title of the document as per the requirement.
* **Set Report Name –** It is a configuration which is used to set name of the report as per the requirement.
* **Set Theme -** This configuration is used to set the theme for Extent report

**Test Listeners**

**TestListeners** - It has configuration for xml report creation using Extent Manager.

It is a handy feature for testing to easily automate testing projects and easily identify how each is performing. For a failed test or a skipped test, you can take a screenshot why the test failed or skipped.

* **ITestResult -** Invoked each time before a test will be invoked. The ITestResult is only partially filled with the references to class, method, start millis and status.
* **getSimpleName-** Returns the simple name of the underlying class as given in thesource code.
* **Substring –** A part of string is called as Substring. It is a subset of another string.
* **getTestClass -** The test class used this object is a result for.
* **getMethodName-** It returns method name used in the test.
* **MarkupHelper** – It helps creating test label and shows the color according to the result of the test
* **getThrowable** – It helps showing error message in reports when test fails
* **getStackTrace –** It returns an array of Stack Trace Elements that represents the current stack trace of a program
* **getLocalizedMessage –** The getLocalizedMessage method of throwable class is used to get a local specific description of the Throwable object when an exception occurred

**Test Start** - This function is used start the test. It shows class name & test name in console

**Test Success** – This creates a green color success message in the report when test passes

**Test Failure**

* It shows result in Red color if test is failed. It takes a screenshot when test is failed and stores it in Framework directory – “build/screenshot”.
* It also uses Get Throwable to get the error message to display in the report

**Shows Test Skipped** - This function shows Blue color when test is skipped

**Result on Finished Test** - This function gives the test result once test is completed

All xmls uses Listners so they are activated before the test. It listens to every event that occurs in a selenium code and creates a report in reports folder.