# Simple Framework for selenium automation

This is a simple test automation framework built in JAVA for automating web application testing using Selenium-WebDriver with descriptive programming and can be adapted as key-word driven framework with slight modification.

Components:

## Core framework

Contains main framework functionality. The FWDriver is starting point of execution of the framework from where the framework initializes logger, WebDriver, etc and then reads the master excel sheet (path is hardcoded as of now) and takes up test cases for execution as listed in the sheet.

Framework uses reflection in order to execute the test cases.

## Logger

Custom logger logs status and information in xml format and has multiple functions and parameters for efficient logging and has selenium independent logic for capturing screenshots (Path hard coded as for now)

## Workflows

Here the actual scripts are placed.

## ObjectRepo

This is object repository and has property files for each UI page and each page file consists of name-value pair for every object on that UI page.

Name will be an identifier as specified by user and value will again be a pair of strings concatenated with “#” in between. The part of string before # is identification method for object in WebDriver (i.e. for now one of classname, name, id, xpath, linktext – this list will be extended later to accommodate all mechanism provided by Selenium)

The part of string after # will be the identifier for object. Or value for the identification mechanism.

## Adding new script

In order to add new script, following should be done:

1. Add class file under workflows and import required packages
2. Create a new property file if UI page in test case workflow is not already present else update existing page with any new object on page which is not already added in property file.
3. Add a method with Page name in ObjectRepo class which returns PageObject for that page
4. In script, use WFDriver.GetWebDriverInstance() to get instance of web driver and Initialize PageObjects for each UI page that might be required.
5. In order to get Webelement on that UI page, call GetElement method on object of respective pageObject and pass the identifier for element as specified in property file.