Class Notes-24

https://www.guru99.com/data-driven-testing.html

Data Driven Testing

Data Driven Testing is a software testing method in which test data is stored in table or spreadsheet format. Data driven testing allows testers to input a single test script that can execute tests for all test data from a table and expect the test output in the same table. It is also called table-driven testing or parameterized testing.

Data Driven Framework

Data Driven Framework is an automation testing framework in which input values are read from data files and stored into variables in test scripts. It enables testers to build both positive and negative test cases into a single test. Input data in data driven framework can be stored in single or multiple data sources like .xls, .xml, .csv and databases.

Why Data Driven Testing?

Data Driven Testing is important because testers frequently have multiple data sets for a single test and creating individual tests for each data set can be time-consuming. Data driven testing helps keeping data separate from test scripts and the same test scripts can be executed for different combinations of input test data and test results can be generated efficiently.

Example:

For example, we want to test the login system with multiple input fields with 1000 different data sets.

To test this, you can take following different approaches:

Approach 1) Create 1000 scripts one for each dataset and runs each test separately one by one.

Approach 2) Manually change the value in the test script and run it several times.

Approach 3) Import the data from the excel sheet. Fetch test data from excel rows one by one and execute the script.

In the given three scenarios first two are laborious and time-consuming. Therefore, it is ideal to follow the third approach.

Thus, the third approach is nothing but a Data-Driven framework.

Test code/script ----------test data
test data-->excel file,json,xml,csv,database,api

In DDF(Data Driven Framework) we seperate test data and test code. we design our test code in a way that it can read the test data and execute tests base on it. Our test code will have the test steps. Thest input data will be in external sources. Our test data will decide how many times a test case runs. (day 16 dataprovider run de ve consola bak-anla)

we use apache-poi dependencies to work with office(excel) files.

ExcelUtil
1.Right click to utilities package
2.New->Java Class
3.Name: ExcelUtil
4.OK

Copy excel file
1.Go to #jamal-only slack channel
2.Download vytracktestdata.xlsx file
3.find your file in your computer(probably in the downloads folder)
4.Right click copy file
5.go to intellij
6.find you resources folder under the test
7.right click your resources folder
8.click paste
9.click ok
10. you will see the file under resources

Exel in uzantisi 2007 ye kadar xls di simdi xlsx butun windows proogramlarinin sonuna x eklendi , xls ve xlsx in ikisini de gozonunde bulundurmamiz gerekiyor

Mvn repository sitesinden copy paste to pom.xml dosyasina

Apache POI poi olan BIRINCISI 4.1.2

Apache POI poi_ooxml. Ikincisi 4.1.2

Boylece exelin eski ve yeni surumleri calisir

Day16. Apache poi dependincies added

Asagidak icode hata Verdi hersey ayni olmasina ragmen

```
package com.cybertek.tests.day16_ddf;

import com.cybertek.utilities.ExcelUtil;
import org.testng.annotations.Test;

import java.util.Arrays;
import java.util.List;
import java.util.Map;

public class ExcelUtilsDemo {


    @Test
    public void readExcelFile(){

        //Create an object from ExcelUtil
        //it accepts two argument
        //Argument 1: location of the excel file(path)  --right click on
excel file-->copy path root
        //Argument 2: sheet that wew want to open(sheetName)

        ExcelUtil qa3short = new
ExcelUtil("src/test/resources/Vytracktestdata.xlsx","QA3-short"); //exelin
icini ac--alttaki sheetlerden birinin ismi QA3-short

        //how many rows in the sheet
        System.out.println("qa3short.rowCount() = " + qa3short.rowCount());

        //how many columns in the sheet
        System.out.println("qa3short.columnCount() = " +
qa3short.columnCount());

        //get all column names
        System.out.println("qa3short.getColumnsNames() = " +
qa3short.getColumnsNames());


        //get all data in list of maps
        List<Map<String, String>> dataList = qa3short.getDataList();
//qa3short.getDataList() yaz ve option+sag enter click 2 times //map lerde
```

```
key and value var


        for (Map<String, String> onerow : dataList) { // iter
+enter//string1 is value (name of the column) string 2 is value , each raw
represents one map
            System.out.println(onerow);

        }

        //get Nona as a value
        System.out.println("dataList.get(2) = " +
dataList.get(2).get("firstname")); //index 0 dan basliyor-->yani bu 3.row--
> 3.map  //2.get methodu map te bilgi icin kullaniyoruz

        //get Harber
        System.out.println("dataList.get(8) = " +
dataList.get(8).get("lastname"));  //excel-ilk row u saymadigi icin index
8=9 uncu rowda harber

        //get all data in 2d array
        String [][] dataArray = qa3short.getDataArray();

        //print 2d array
        System.out.println( Arrays.deepToString(dataArray));

    }
}


//soltarafta src nin altindaki main ile isimiz yok, developerslar
kullaniyor onu
//excel file yuklemek istersek javanin okuna tikla ,hemen altinda ki
resources u kullan sol tarafta
//masaustundeki vytrack exel dosyasin da copy --> resources a git -->right
click-->paste dedik ve
// eklendi resources dosyasina intelij de --> dikkat main dosyasinin
altindaki resources a copyalama
//how do you handle,read nad write excel files in intellij --> i use apache
poi dependincies sonra i have my excelUtil class that
// i have ready methods to read excel files in different format  > we can
say we created with our team or i created
```