Last updated: Aug 12, 2020

Selenium Java Training - Session 27 - Log4j2 and Test Data from Database

Log4j2

1. Log4j is for logging
2. Logs are like running commentary
3. The purpose of logs is to know how the script got executed at later point of time and if it failed where exactly it gone wrong
    - We can know what got executed in the code and how it got executed.
    - We can know where exactly in the code, Exceptions / errors occurred.
4. Implement logging in Selenium Automation code using **System.out.println()** statements
    - Create a Java Project
    - Configure the Project with Selenium WebDriver
    - Create Selenium Automation code to visit Omayo blog, navigate to Compendium site, navigate back to Omayo blog, forward again to Compendium site and close the browser.
    - Write **System.out.println()** statements for logging - View here
5. Disadvantages of SOP logging
    - SOP's are for simple logging, and cannot be used for advanced logging.
        - We cannot turn off the logs when required
        - Logs are captured in the console, instead of a separate file
        - No Time-stamp will be displayed in the required format (Developers may need this from testers)
        - We cannot provide earlier logs
        - Cannot differentiate between logs (Level of logs)
        - And many more
6. To resolve the above disadvantages, we have to use advanced logging - **Log4j logging**
7. Similar to Java, Selenium, TestNG and POI API's , Log4j is released into market as API by Apache guys
8. Implementing Log4j in Selenium Automation
    - Step1: Go to the downloads page of Log4j and download the zip file
    - Step2: Extract the Zip file and configure the Project with log4j-core ad log4j-api jars only
    - Step3: Write the below code
        - Logger logger = LogManager.getLogger(Demo.class.getName());
        - logger.debug for all the general logs
        - logger.info for successfull test
        - logger.error for failure test
        - Execute and observe that no logs will be displayed in the output console, **except error logs**
        - We need a configuration file for all the logs to work
    - Step4: Create log4j configuration file by following the below steps:
        - Search for 'Log4j configuration' and go to the required URL page and find any xml

        - ration files
            - There are two types of tags
                - Appenders (Information on where to log)
                - Loggers (Information on what to log)
        - Create resources folder under the project
            - Create an xml file with the name log4j2.xml
            - Paste the xml configuration things into the file
            - Change the root level to All (Log Levels - All < Trace < Debug < Info < Warn < Error < Fatal < Off)
            - Build the project and add the resources folder
    - Step5: Printing the logs to a file instead of console
        - Add RollingFile tags between Appenders tags - View here
            - Change the AppenderRef to File from Console
                - <AppenderRef ref="File" />
        - Create logs folder under Project and create prints.log file
            - Add the properties tags before the Appenders tag - View here
        - Run the Demo and observe that the logs got printed into the File instead of output console

Test Data from Database

1. Install the MySQL Server in your machine as explained at this Video - https://www.youtube.com/watch?v=NQXxFPyqmDg
2. Using MySQL Workbench, connect to MySQL and perform the below:
    1. Connect to a DB
        1. create table Employees(id int, name varchar(20),location varchar(20),experience int);
        2. describe Employees;
    2. Insert records into Table
        1. insert into Employees values(1,'Arun','Hyderabad',12);
        2. insert into Employees values(2,'Varun','Bangalore',9);
        3. insert into Employees values(3,'Tharun','Delhi',7);
    3. Select a record from the Table
        1. Select * from Employees where id=3;
3. Create a Maven Project
    1. Configure this Project with 'MySQL Java Connector' Jar file
    2. Write the Java program to print a single record data - View here
4. Use the same concept to read the Username and Password from the Table at Database and pass it into the Selenium Code
    1. Use http://tutorialsninja.com/demo/ for this Demonstration to Login

By,
Arun Motoori