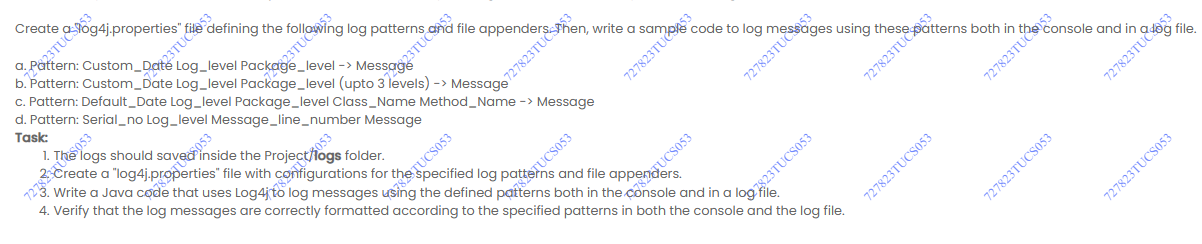
Log4j\_Session 2



Project/src/test/java/runner/TestAmazon.java

package runner;

import java.net.MalformedURLException;

import java.net.URL;

import java.time.Duration;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeOptions;

import org.openqa.selenium.remote.RemoteWebDriver;

import org.openqa.selenium.support.events.EventFiringDecorator;

import org.openqa.selenium.support.events.WebDriverListener;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.BeforeMethod;

import org.testng.annotations.Test;

import org.openqa.selenium.remote.DesiredCapabilities;

import utils.EventHandler;

import utils.LoggerHandler;

public class TestAmazon {

    public static WebDriver driver;

    public static int IMPLICIT\_WAIT\_TIME =10;

    public static int  PAGE\_LOAD\_TIME =15;

 @BeforeMethod

    public void openBrowser() throws Throwable {

        DesiredCapabilities dc = new DesiredCapabilities();

        dc.setBrowserName("chrome");

        driver = new RemoteWebDriver(new URL("http://localhost:4444/"), dc);

        driver.manage().window().maximize();

        driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(IMPLICIT\_WAIT\_TIME));

        driver.manage().timeouts().pageLoadTimeout(Duration.ofSeconds(PAGE\_LOAD\_TIME));

        //get url

        WebDriverListener listener = new EventHandler();

        driver = new EventFiringDecorator<>(listener).decorate(driver);

    }

    @Test

    public void testAmazonLogin(){

        LoggerHandler.info("Entered successfully the username");

    }

    @AfterMethod

    public void closeBrowser(){

        if(driver!=null){

            driver.quit();

        }

    }

    }

Project/src/test/java/utils/LoggerHandler.java

package utils;

import java.text.SimpleDateFormat;

import java.util.Date;

import org.apache.log4j.Logger;

import org.apache.log4j.PropertyConfigurator;

public class LoggerHandler {

    private static final Logger logger=Logger.getLogger(LoggerHandler.class);

    static{

        try{

            String timestamp=new SimpleDateFormat("yyyyMMdd-HHmmss").format(new Date());

            System.setProperty("current.date",timestamp);

            PropertyConfigurator.configure(

                LoggerHandler.class.getClassLoader().getResource("log4j.properities")

            );

        }

        catch(Exception e){

            e.printStackTrace();

        }

    }

    public static void info(String msg){

        logger.info(msg);

    }

    public static void warn(String msg){

        logger.warn(msg);

    }

    public static void debug(String msg){

        logger.debug(msg);

    }

    public static void error(String msg){

        logger.error(msg);

    }

}

Project/src/main/resources/log4j.properities

log4j.rootLogger=DEBUG, file

log4j.appender.file=org.apache.log4j.RollingFileAppender

log4j.appender.file.File=${user.dir}/logs/myapp-${current.date}.log

log4j.appender.file.MaxFileSize=5MB

log4j.appender.file.MaxBackupIndex=10

log4j.appender.file.layout=org.apache.log4j.PatternLayout

log4j.appender.file.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%L - %m%n