package tek.sdet.framework.base;

import java.io.FileNotFoundException;

import java.time.Duration;

import java.time.temporal.ChronoUnit;

import java.util.HashMap;

import org.apache.log4j.PropertyConfigurator;

import org.apache.log4j.Logger;

import org.openqa.selenium.WebDriver;

import tek.sdet.framework.config.Browser;

import tek.sdet.framework.config.ChromeBrowser;

import tek.sdet.framework.config.ChromeHeadless;

import tek.sdet.framework.config.EdgeBrowser;

import tek.sdet.framework.config.FirefoxBrowser;

import tek.sdet.framework.utilities.ReadYamlFiles;

public class BaseSetup {

private static WebDriver webDriver;

private final ReadYamlFiles environmentVariables;

public static Logger logger;

public BaseSetup() {

String filePath = System.getProperty("user.dir") + "/src/main/resources/env\_config.yml";

String log4JPath = System.getProperty("user.dir") + "/src/main/resources/log4j.properties";

try {

environmentVariables = ReadYamlFiles.getInstance(filePath);

} catch (FileNotFoundException e) {

System.out.println("Failed for load environment context. check possible file path errors");

throw new RuntimeException("Failed for environment context with message " + e.getMessage());

}

logger = logger.getLogger("logger\_File");

PropertyConfigurator.configure(log4JPath);

}

public WebDriver getDriver() {

return webDriver;

}

public void setupBrowser() {

HashMap uiProperties = environmentVariables.getYamlProperty("ui");

System.out.println(uiProperties);

String url = uiProperties.get("url").toString();

Browser browser;

switch (uiProperties.get("browser").toString().toLowerCase()) {

case "chrome":

if ( (boolean) uiProperties.get("headless")) {

browser = new ChromeHeadless();

} else {

browser = new ChromeBrowser();

}

webDriver = browser.OpenBrowser(url);

break;

case "firefox":

browser = new FirefoxBrowser();

webDriver = browser.OpenBrowser(url);

break;

case "edge":

browser = new EdgeBrowser();

webDriver = browser.OpenBrowser(url);

break;

default:

throw new RuntimeException("unknown Browser check environment properties");

}

webDriver.manage().window().maximize();

webDriver.manage().timeouts().implicitlyWait(Duration.of(20, ChronoUnit.SECONDS));

webDriver.manage().timeouts().pageLoadTimeout(Duration.ofSeconds(20));

}

public void quitBrowser() {

if (webDriver != null)

webDriver.quit();

}

}