1. **As a QA Engineer, how do you start your day?**

**Ans :**

**As a QA I start my by writing my TODO list for the day then start targeting to complete those with efficient way possible.**

1. **Do you care about your fellow QA engineer developing the same automation code base? If yes, please describe the action while developing.**

**Ans: Yes, I do. Need to make sure the coding guidelines are followed and use utility class or OOPS concept as far as possible.**

1. **Base on below context, develop the structure for automation script using any language revolving around *Selenium, Gherkins(BDD)* with your best ideology.** *(There will be no real application; just pseudo exercise)**“Our client covid-rox is developing an application to track how many people are infected by it worldwide. It will be available in three different platform (iOS,Android and Web); fast forward, below will be the pseudo application just enough for you to code”***User stories** *As a user, I would like to reach landing screen after login successfully with a valid credential.  
     
   As a user, I would like to be prompt with error message if I enter password and username length greater than 8 and invalid credential entere.  
     
   As a user, I would like to see submit button disable if either one of the fields(username/password) is empty.*  
     
   **iOS element IDs**  
   username field: ios\_username  
   password field: ios\_password  
   error message field: ios\_error\_message  
   login button: ios\_login\_button  
     
   **Android element IDs**username field: android\_username  
   password field: android\_password  
   error message field: android\_error\_message  
   login button: android\_login\_button  
     
   **Web app element IDs**username field: web\_username  
   password field: web\_password  
   error message field: web\_error\_message  
   login button: web\_login\_button

Solution :

Please find attached javaFiles :

1. Feature file for all scenario

Text

Description automatically generated

1. Step Definations

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

1. Page Objects:

Text

Description automatically generated

Text

Description automatically generated

Text, chat or text message

Description automatically generated

Feature: Track covid Infected people around the world  
  
 Scenario Outline: Successful login with valid credentials  
 """  
 As a user, I would like to reach landing screen after login successfully with a valid credential.  
 """  
 Given I am on the login page for **<Platform>** When I enter the valid credentials  
 Then I shall see successful login to landing screen  
 Examples:  
 | **Platform** |  
 | **web** |  
 | **ios** |  
 | **android** |  
  
  
 Scenario Outline: UnSuccessful login with password and username length greater than 8 and invalid credential  
 """  
 As a user, I would like to be prompt with error message if I enter password and username length greater than 8 and invalid credential entere.  
 """  
 Given I am on the login page for **<Platform>** When I enter the invalid credentials  
 Then I shall see error message  
 Examples:  
 | **Platform** |  
 | **web** |  
 | **ios** |  
 | **android** |  
  
 Scenario Outline: I would like to see submit button disable if either one of the fields(username/password) is empty.  
 """  
 As a user, I would like to see submit button disable if either one of the fields(username/password) is empty.  
 """  
 Given I am on the login page for **<Platform>** When I enter either username or password only  
 Then I shall not see submit button  
 Examples:  
 | **Platform** |  
 | **web** |  
 | **ios** |  
 | **android** |

package covidrox.stepDefinations;

import io.appium.java\_client.android.AndroidDriver;

import io.appium.java\_client.ios.IOSDriver;

import io.appium.java\_client.remote.MobileCapabilityType;

import io.cucumber.java.en.Given;

import io.cucumber.java.en.Then;

import io.cucumber.java.en.When;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.remote.DesiredCapabilities;

import org.testng.Assert;

import java.net.MalformedURLException;

import java.net.URL;

public class CovidRox {

private WebDriver driver;

public static String PLATFORM = null;

PageObjectSelector pageObject;

String validUserName = "userName1";

String validPassword = "password1";

String inValidUser = "inValidUser2";

String inValidPassword = "inValidPassword2";

@Given("^I am on the login page for (android|ios|web)$")

public void OpenTheApplication(String deviceType) throws MalformedURLException {

PLATFORM = deviceType;

driver = getDriver(deviceType);

pageObject = getPageObject();

if (PLATFORM.contains("web")) {

driver.get("https://yoururlforcovidrox.com");

Assert.assertEquals("yourExpectedTitle", driver.getTitle());

} else {

//Assertion for ios/android

Assert.assertEquals("", "");

}

}

@When("I enter the valid credentials")

public void EnterCredentials() {

pageObject.LoginWithCredentials(validUserName, validPassword);

}

@When("I enter the invalid credentials")

public void EnterInValidCredentials() {

pageObject.LoginWithCredentials(inValidUser, inValidPassword);

}

@Then("I shall see successful login to landing screen")

public void LandingScreen() {

boolean isLoginSuccess = pageObject.isSuccessLogin();

Assert.assertEquals(true, isLoginSuccess);

}

@Then("I shall see error message")

public void ErrorMessage() {

boolean isErrorMessage = !pageObject.isSuccessLogin();

Assert.assertEquals(true, isErrorMessage);

}

@When("I enter either username or password only")

public void EnterUserNameOrPassword() {

pageObject.EnterOnlyUserName(validUserName);

}

@When("I shall not see submit button")

public void CheckSubmitButtonDisplay() {

Assert.assertEquals(true, pageObject.isSubmitButtonEnabled());

}

//Shall create a separate Page Object Class for below:

public WebDriver getDriver(String deviceType) throws MalformedURLException {

switch (deviceType) {

case "web":

return new ChromeDriver();

case "android":

return new AndroidDriver<>(new URL("http://localhost:4723/wd/hub"), getAndroidDesiredCapabilities());

case "ios":

return new IOSDriver<>(new URL("http://localhost:4723/wd/hub"), getIOSDesiredCapabilities());

default:

return null;

}

}

public PageObjectSelector getPageObject() {

if (PLATFORM.equals("web")) {

pageObject = new WebObjectRepo(driver);

} else {

pageObject = new MobileObjectRepo(driver);

}

return pageObject;

}

//Android

private static DesiredCapabilities getAndroidDesiredCapabilities() {

DesiredCapabilities caps = new DesiredCapabilities();

caps.setCapability(MobileCapabilityType.AUTOMATION\_NAME, "UiAutomator2");

caps.setCapability("appActivity", "com.covid.rox.MainActivity");

caps.setCapability("appPackage", "com.covid.rox");

caps.setCapability("deviceName", "emulator-5554");

caps.setCapability("platformVersion", "9");

caps.setCapability("platformName", "Android");

caps.setCapability("app", System.getProperty("user.dir") + "/src/test/resources/app/covid-rox.apk");

return caps;

}

//IOS

private static DesiredCapabilities getIOSDesiredCapabilities() {

DesiredCapabilities caps = new DesiredCapabilities();

caps.setCapability(MobileCapabilityType.AUTOMATION\_NAME, "XCUITest");

caps.setCapability("bundleId", "com.covid.rox");

caps.setCapability("deviceName", "iPhone XS");

caps.setCapability("platformVersion", "13.2");

caps.setCapability("platformName", "iOS");

return caps;

}

}

package covidrox.stepDefinations;

import io.appium.java\_client.MobileDriver;

import io.appium.java\_client.MobileElement;

import io.appium.java\_client.pagefactory.AndroidFindBy;

import io.appium.java\_client.pagefactory.AppiumFieldDecorator;

import io.appium.java\_client.pagefactory.iOSXCUITFindBy;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.support.PageFactory;

import java.time.Duration;

public class MobileObjectRepo implements PageObjectSelector{

@AndroidFindBy(id = "android\_username")

@iOSXCUITFindBy(id = "ios\_username")

MobileElement userName;

@AndroidFindBy(id = "android\_password")

@iOSXCUITFindBy(id = "ios\_password")

MobileElement password;

@AndroidFindBy(id = "android\_error\_message")

@iOSXCUITFindBy(id = "ios\_error\_message")

MobileElement errorMessage;

@AndroidFindBy(id = "android\_login\_button")

@iOSXCUITFindBy(id = "ios\_login\_button")

MobileElement loginButton;

MobileObjectRepo(WebDriver driver){

PageFactory.initElements(new AppiumFieldDecorator((MobileDriver)driver, Duration.ofSeconds(5)), this);

}

@Override

public void LoginWithCredentials(String userName, String password) {

this.userName.sendKeys(userName);

this.password.sendKeys(password);

this.loginButton.click();

}

@Override

public boolean isSuccessLogin() {

return !errorMessage.isDisplayed();

}

@Override

public void EnterOnlyUserName(String userName) {

this.userName.sendKeys(userName);

}

@Override

public boolean isSubmitButtonEnabled() {

return loginButton.isEnabled();

}

}

package covidrox.stepDefinations;

public interface PageObjectSelector {

void LoginWithCredentials(String userName,String password);

boolean isSuccessLogin();

void EnterOnlyUserName(String userName);

boolean isSubmitButtonEnabled();

}

package covidrox.stepDefinations;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import io.appium.java\_client.pagefactory.AndroidFindBy;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

public class WebObjectRepo implements PageObjectSelector{

@FindBy(id = "web\_username")

WebElement userName;

@FindBy(id = "web\_password")

WebElement password;

@AndroidFindBy(id = "web\_error\_message")

WebElement errorMessage;

@AndroidFindBy(id = "web\_login\_button")

WebElement loginButton;

WebDriver driver;

WebObjectRepo(WebDriver driver){

this.driver=driver;

PageFactory.initElements(driver,this);

}

@Override

public void LoginWithCredentials(String username, String password) {

userName.sendKeys(username);

this.password.sendKeys(password);

this.loginButton.click();

}

@Override

public boolean isSuccessLogin() {

return !errorMessage.isDisplayed();

}

@Override

public void EnterOnlyUserName(String userName) {

this.userName.sendKeys(userName);

}

@Override

public boolean isSubmitButtonEnabled() {

return loginButton.isEnabled();

}

}

1. Write a short function to retrieve all key values in the below JSON.
2. {
3. "total\_rows": 184,
4. "offset": 0,
5. "rows": [
6. {
7. "id": "SIT/DRA/2020/001",
8. "key": "SIT/DRA/2020/001",
9. "value": {
10. "rev": "645-2f3611a220ac5cc86186764304f2e4b5"
11. }
12. },
13. {
14. "id": "SIT/DRA/2020/002",
15. "key": "SIT/DRA/2020/002",
16. "value": {
17. "rev": "630-19697a00a20857b46406c9ed55fa75da"
18. },
19. "counter1": [
20. {
21. "id": "1234",
22. "counter2": [
23. {
24. "id": "1234"
25. }
26. ]
27. }
28. ]
29. }
30. ],
31. "date": "12/12/2001"
32. }

Solution :

Below method will print all Key values from above Json:  
public static void RetrieveAllKeyValueFromJson(String jsonString){  
 JSONObject jsonObject = new JSONObject(jsonString);  
 Iterator<String> keys = jsonObject.keys();  
 while(keys.hasNext()) {  
 String key = keys.next();  
 System.*out*.println(key);  
 if(jsonObject.get(key) instanceof JSONArray) {  
 JSONArray array = (JSONArray) jsonObject.get(key);  
 JSONObject object = (JSONObject) array.get(0);  
 Iterator<String> innerKeys = object.keys();  
 while(innerKeys.hasNext()) {  
 String innerKey = innerKeys.next();  
 System.*out*.println(innerKey);  
 }  
 }  
 }  
}

*Text

Description automatically generated*