

1)

Feature: do the following test on REQRES site

Scenario: Perform operation on end points and get the request and response

Given user is on REQRES web site

When user click on each end point and get request and response

Then user perform Rest Assured operation

2) StepDefination

```
package quickacademy.MavenJava;
```

```
import java.util.Arrays;
```

```
import java.util.List;
```

```
import org.junit.Assert;
```

```
import org.openqa.selenium.By;
```

```
import org.openqa.selenium.JavascriptExecutor;
```

```
import org.openqa.selenium.WebDriver;
```

```
import org.openqa.selenium.WebElement;
```

```
import org.openqa.selenium.chrome.ChromeDriver;
```

```
import cucumber.api.java.en.Given;
```

```
import cucumber.api.java.en.Then;
```

```
import cucumber.api.java.en.When;
```

```
public class TechMahindra {
```

```
    WebDriver driver;
```

```
    JavascriptExecutor jse = (JavascriptExecutor)driver;
```

```
// public static void main(String[] args) throws InterruptedException {
```

```
    @Given("^user is on REQRES web site$")
```

```
    public void user_is_on_REQRES_web_site() throws InterruptedException
```

```
    {
```

```
        System.setProperty("webdriver.chrome.driver", "C:\\Users\\SHANIR\\Documents\\chromeexe\\chromedriver_win32\\chromedriver.exe");
```

```
        WebDriver driver = new ChromeDriver();
```

```
        // Implicit Wait
```

```
// driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);
```

```
driver.get("https://reqres.in/");
```

```
Thread.sleep(2000);
```

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

```
Thread.sleep(2000);
```

```
// Scroll down
```

```
// JavascriptExecutor jse = (JavascriptExecutor)driver;
```

```
jse.executeScript("window.scrollTo(0,500)");
```

```
}
```

```

// 001 UI Spec
@When("^user click on each end point and get request and response$")
public void When_user_click_on_each_end_point_and_get_request_and_response()
{
    String names[] = {"LIST USERS","SINGLE USER", "SINGLE USER NOT FOUND", "LIST <RESOURCE
>", "SINGLE <RESOURCE>"};

    Thread.sleep(3000);
    int j=0;

    List<WebElement> links = driver.findElements(By.tagName("a"));
    System.out.println("Total Links on the page: " + links.size());
    for(int i=0; i<links.size(); i++)
    {

        String name = links.get(i).getText();
        String name1 = name.trim();

        //Convert Array to ArrayList for easy search
        Thread.sleep(2000);
        List b = Arrays.asList(names);
        if(b.contains(name1))
        {
            j++;
            Thread.sleep(3000);
            driver.findElements(By.xpath("//*[@data-key='try-link']")).get(i).click();

            Thread.sleep(2000);
            String request1 = driver.findElement(By.cssSelector("p.request-title")).getText();
            System.out.println(request1);
            Thread.sleep(2000);
            String response1 = driver.findElement(By.cssSelector("p[class='response-title']")).getText();
            System.out.println(response1);

            if(j==names.length)
            {
                break;
            }
        }
    }

// 001B - Support page
@Then("^user perform support page operation$")
public void user_perform_support_page_operation()
{
    // Scroll down
    jse.executeScript("window.scrollTo(0,1000)");
    driver.findElement(By.cssSelector("a[href='support']")).click();

    boolean oneTimePayment = driver.findElement(By.xpath("//*[@id='supportForm']/div/label")).isDisplaye
d();
    Assert.assertTrue(oneTimePayment);

    boolean monthlySupport = driver.findElement(By.xpath("//*[@for='supportRecurring']")).isDisplayed();

```

```

Assert.assertTrue(monthlySupport);

// Upgrade details

driver.findElement(By.cssSelector("button#trigger-pro")).click();
driver.findElement(By.id("mce-EMAIL")).sendKeys("abc@xyz.com");
driver.findElement(By.cssSelector("button")).click();
String confirmMsg = driver.findElement(By.cssSelector(".bodyContent.rounded6")).getText();
System.out.println(confirmMsg);
}
}

package test;

import static io.restassured.RestAssured.given;
import static org.hamcrest.Matchers.equalTo;

import files.TechMahindraPayload;
import io.restassured.RestAssured;

public class TectMahindraAPI {

    public static void main(String[] args) {

        RestAssured.baseURI= "https://reqres.in/api/users/2";
        given().log().all().header("Content-Type","application/json; charset=utf-8")
            .body(TechMahindraPayload.create())
            .when().post("api/users")
            .then().log().all().assertThat().statusCode(201).header("Server", "cloudflare");

    }
}

package files;

public class TechMahindraPayload {
    public static String create()
    {
        return "{\r\n" +
            "  \"name\": \"morpheus\",\r\n" +
            "  \"job\": \"leader\"\r\n" +
            "}";
    }
}
}

```