**Automation Test**

1. **API Automation Test**
2. GET -> <https://jsonplaceholder.cypress.io/posts->>To make sure this end-point have a correct data-type below:

* userId -> Integer
* id -> Integer
* title -> String
* body -> String

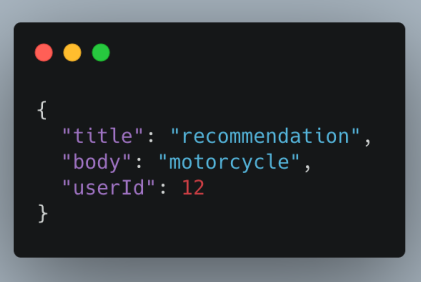
Test scenario:

|  |
| --- |
| @Api  Feature: Open Cypress API Site    @GetApi    Scenario:      Given Access URI      Then Compare    @PostApi    Scenario:      Given Access post URI      Then Compare result |

Automation test script:

|  |
| --- |
| require 'uri'  require 'net/http'  expectedOutput = [Integer, Integer, String, String]  actualOutput = []  Given("Access URI") do    uri = URI("https://jsonplaceholder.cypress.io/posts")    res = Net::HTTP.get\_response(uri) #hit uri    data = JSON.parse(res.body) #diubah/diparsing ke bentuk json    actualOutput.push(data[0]['userId'].class) #ngambil tipe data fari field userID, id, title, body lalu dipush ke actual output    actualOutput.push(data[0]['id'].class)    actualOutput.push(data[0]['title'].class)    actualOutput.push(data[0]['body'].class)  end  Then("Compare") do #membandingkan expected (userID, id) dengan actualOutput    expect(actualOutput).to eq(expectedOutput)  end |

1. POST-> <https://jsonplaceholder.cypress.io/posts> -> To make sure this end-point get a correct response compare with the input data:



Automation test script:

|  |
| --- |
| require 'uri'  require 'net/http'  expectedOutput = ['recomendation', 'motorcycle', 12]  actualOutput = []  Given("Access post URI") do    uri = URI('https://jsonplaceholder.cypress.io/posts')    res = Net::HTTP.post\_form(uri, 'title'=>'recomendation', 'body'=>'motorcycle', 'userId'=>12)    data = JSON.parse(res.body)    actualOutput.push(data['title'])    actualOutput.push(data['body'])    actualOutput.push(data['userId'])  end  Then("Compare result") do    expect(actualOutput).to eq(expectedOutput)  end |

1. **Web and Mobile Automation Test**
2. Create automation test script to test 1 of these scenarios using Bukalapak website:

- Register

- Login

- Add to Cart

- Search Product

Test scenario for script automation:

|  |
| --- |
| @Bukalapak    Feature: Completing the register, login, search, and add to chart process      @Register      Scenario:        Given I am access Bukalapak site        When I go to register page        Then Compare url      @LoginSuccess      Scenario:        Given I am access Bukalapak site        When I go to login page        Then Input true password        Then Compare after login      @SearchSuccess      Scenario:        Given I am access Bukalapak site        When I go to login page        Then Input true password        Then Go to search        Then Compare after search      @AddToCartSuccess      Scenario:        Given I am access Bukalapak site        When I go to login page        Then Input true password        Then Go to search        Then I add to cart        Then Compare after cart |

Automation test script:

|  |
| --- |
| require 'selenium-webdriver'  require 'rubygems'  require 'rspec'  driver = Selenium::WebDriver.for :chrome  link = "https://www.bukalapak.com/"  email = "gita.nddp@gmail.com"  namaLengkap = "Gita Nadapdap"  password = "qwerty030"  Given("I am access Bukalapak site") do    driver.navigate.to link  end  When("I go to register page") do    driver.find\_element(:xpath, '//\*[@id="vm\_\_white-header-dweb"]/section/header/div[3]/div/div/div[2]/div/a[1]/p').click    driver.find\_element(:xpath, '/html/body/main/div/div[2]/section/div[1]/div[1]/div/input').send\_keys email    driver.find\_element(:xpath, '/html/body/main/div/div[2]/section/div[2]/button').click    sleep(5)    driver.find\_element(:xpath, '/html/body/main/section/div[4]/div[2]/div/button[1]').click    sleep(5)    driver.find\_element(:xpath, '//\*[@id="TFAJS-1633491529932"]/section/div/div[2]/div[2]/div/div/div/div[2]/button').click    sleep(5)    driver.find\_element(:xpath, '/html/body/main/section/div/div[1]/div/input').send\_keys namaLengkap    driver.find\_element(:xpath, '/html/body/main/section/div/div[2]/div/input').send\_keys password    driver.find\_element(:xpath, '/html/body/main/section/div/button').click    sleep(2)  end  Then("Compare url") do    url = driver.current\_url    puts url    expect(url).to eq("https://accounts.bukalapak.com/register?comeback=https%3A%2F%2Fwww.bukalapak.com%2F&from=nav\_header")  end  When("I go to login page") do    driver.find\_element(:xpath, '//\*[@id="vm\_\_white-header-dweb"]/section/header/div[3]/div/div/div[2]/div/a[2]/p').click    driver.find\_element(:xpath, '//\*[@id="user\_identity\_textfield"]').send\_keys email    sleep(2)    driver.find\_element(:xpath, '//\*[@id="submit\_button"]').click    sleep(2)  end  Then ("Input true password") do    driver.find\_element(:xpath, '//\*[@id="input-password"]').send\_keys password    driver.find\_element(:xpath, '//\*[@id="btn-login"]').click    sleep(3)  end  Then("Compare after login") do    url = driver.current\_url    puts url    expect(url).to eq("https://www.bukalapak.com/?flash=you\_login#")  end    Then("Go to search") do    driver.find\_element(:xpath, '//\*[@id="v-omnisearch\_\_input"]').send\_keys 'earphone jbl c100si original'    driver.find\_element(:xpath, '//\*[@id="v-omnisearch"]/button').click  end  Then("I add to cart") do    driver.find\_element(:xpath, '//\*[@id="product-explorer-container"]/div/div[1]/div[2]/div/div[2]/div[3]/div/div[2]/div/div/div[2]').click    driver.find\_element(:xpath, '//\*[@id="section-main-product"]/div[2]/div[5]/div/button[2]').click  end  Then("Compare after search") do    url = driver.current\_url    expect(url) == ("https://www.bukalapak.com/products?from=omnisearch&from\_keyword\_history=false&search%5Bkeywords%5D=earphone%20jbl%20c100si%20original&search\_source=omnisearch\_keyword&source=navbar")  end  Then("Compare after cart") do    url = driver.current\_url    puts url    expect(url).to eq("https://www.bukalapak.com/p/handphone/headset-earphone/1n6xija-jual-headset-handsfree-earphone-jbl-c100si-original?from=list-product&pos=1&keyword=earphone%20jbl%20c100si%20original&funnel=omnisearch&product\_owner=normal\_seller&cf=1&ssa=1&sort\_origin=relevansi&search\_sort\_default=true&promoted=1")  end |

1. Create and automation test (positive and negative)for apps with 1 of these scenarios:

* Register
* Login

main.java

|  |
| --- |
| import io.appium.java\_client.android.AndroidDriver; import io.appium.java\_client.android.AndroidElement; import io.appium.java\_client.remote.MobileCapabilityType; import org.openqa.selenium.remote.DesiredCapabilities;  import java.net.MalformedURLException; import java.net.URL;  public class main {  public static void main(String[] args) throws MalformedURLException, InterruptedException {    DesiredCapabilities desiredCapabilities = new DesiredCapabilities();   desiredCapabilities.setCapability(MobileCapabilityType.*DEVICE\_NAME*, "e963e017");  desiredCapabilities.setCapability("platformName", "android");  desiredCapabilities.setCapability("appPackage", "com.loginmodule.learning");  desiredCapabilities.setCapability("appActivity", ".activities.LoginActivity");   AndroidDriver<AndroidElement> driver= new AndroidDriver<AndroidElement>(new URL("http://127.0.0.1:4723/wd/hub"),desiredCapabilities);   scriptAutomationLoginTest loginTest = new scriptAutomationLoginTest();  scriptAutomationRegisterTest registerTest = new scriptAutomationRegisterTest();   String email = "gita.nddp@gmail.com";  String password = "gita";  String name = "gita";  String confirmPassword = "gita";   //positive testcase  registerTest.RegisterTest(driver,email,password,confirmPassword, name); // loginTest.LoginTest(driver,email,password);   } } |

scriptAutomationRegisterTest.java

|  |
| --- |
| import io.appium.java\_client.MobileElement; import io.appium.java\_client.android.AndroidDriver; import io.appium.java\_client.android.AndroidElement; import org.testng.Assert;  import java.net.MalformedURLException;  public class scriptAutomationRegisterTest {  public void RegisterTest(AndroidDriver<AndroidElement> driver,  String email,  String password,  String confirmPassword,  String name) throws MalformedURLException, InterruptedException {    MobileElement el1 = (MobileElement) driver.findElementById("com.loginmodule.learning:id/textViewLinkRegister");  el1.click();  Thread.*sleep*(2000);  MobileElement el2 = (MobileElement) driver.findElementById("com.loginmodule.learning:id/textInputEditTextName");  el2.sendKeys(name);  MobileElement el3 = (MobileElement) driver.findElementById("com.loginmodule.learning:id/textInputEditTextEmail");  el3.sendKeys(email);  MobileElement el4 = (MobileElement) driver.findElementById("com.loginmodule.learning:id/textInputEditTextPassword");  el4.sendKeys(password);  MobileElement el5 = (MobileElement) driver.findElementById("com.loginmodule.learning:id/textInputEditTextConfirmPassword");  el5.sendKeys(confirmPassword);  MobileElement el6 = (MobileElement) driver.findElementById("com.loginmodule.learning:id/appCompatButtonRegister");  el6.click();   Thread.*sleep*(5000);  MobileElement el9 = (MobileElement) driver.findElementById("com.loginmodule.learning:id/appCompatTextViewLoginLink");  el9.click();   } } |

scriptAutomationLoginTest.java

|  |
| --- |
| import io.appium.java\_client.MobileElement; import io.appium.java\_client.android.AndroidDriver; import io.appium.java\_client.android.AndroidElement; import org.testng.Assert;  import java.net.MalformedURLException;  public class scriptAutomationLoginTest {  public void LoginTest(AndroidDriver<AndroidElement> driver, String email, String password) throws MalformedURLException, InterruptedException {   Thread.*sleep*(1000);  MobileElement el3 = (MobileElement) driver.findElementById("com.loginmodule.learning:id/textInputEditTextEmail");  el3.sendKeys(email);  MobileElement el5 = (MobileElement) driver.findElementById("com.loginmodule.learning:id/textInputEditTextPassword");  el5.sendKeys(password);  MobileElement el6 = (MobileElement) driver.findElementById("com.loginmodule.learning:id/appCompatButtonLogin");  el6.click();   Thread.*sleep*(2000);  MobileElement el7 = (MobileElement) driver.findElementByXPath("/hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.view.ViewGroup/android.widget.FrameLayout[2]/android.widget.LinearLayout/android.support.v7.widget.LinearLayoutCompat/android.widget.TextView[1]");  Assert.*assertEquals*(el7.getText(), "Android NewLine Learning" );  } } |