Please complete the following automation exercises using your automation test tool of choice.

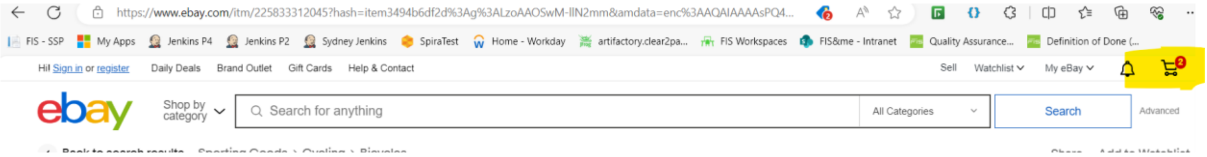
Please structure your solution as you would do in a work environment / setting.

UI Automation

On ebay.com website, please automate the following test scenario

Scenario 1 – Verify item can be added to Cart

1. Open browser
2. Navigate to ebay.com
3. Search for ‘book’
4. Click on the first book in the list
5. In the item listing page, click on ‘Add to cart’
6. verify the cart has been updated and displays the number of items in the cart as shown below in yellow.



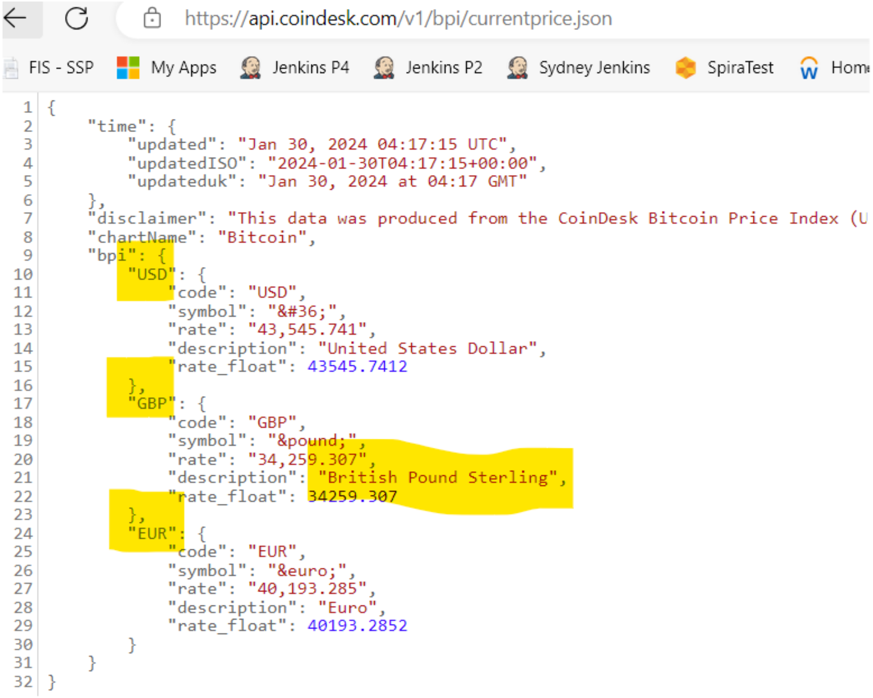
Ans-

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

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

 import org.openqa.selenium.support.ui.ExpectedConditions;

 import org.openqa.selenium.support.ui.WebDriverWait;


public class EbayAddToCart { public static void main(String[] args)


{

System.setProperty("webdriver.chrome.driver", "/path/to/chromedriver");

WebDriver driver = new ChromeDriver();

//Step 1-Open Browser & Navigate to ebay.com

driver.get("https://www.ebay.com");
driver.manage().window().maximize();


// Search for Book

WebDriverWait wait = new WebDriverWait(driver, 10);

WebElement searchBox = wait.until(ExpectedConditions.presenceOfElementLocated(By.id("gh-ac")));

searchBox.sendKeys("book");

WebElement searchButton = driver.findElement(By.id("gh-btn"));

searchButton.click();


//Click on the first book in the list

WebElement firstBook =
wait.until(ExpectedConditions.presenceOfElementLocated(By.xpath("(//li[@class='s-item'])[1]//a")));

firstBook.click();


// In the item listing page, click on ‘Add to cart’

WebElement addToCartButton =
wait.until(ExpectedConditions.elementToBeClickable(By.id("isCartBtn_btn")));

addToCartButton.click();


//Verify the cart has been updated and displays the number of items in the cart as shown below in
yellow.


String cartCount = cartCountElement.getText();

if ("1".equals(cartCount))

 { System.out.println("Item successfully added to cart. Cart count: " + cartCount); }

else { System.out.println("Error: Expected cart count to be 1, but found " + cartCount); }

 driver.quit(); } }
```

API Automation

On the following endpoint - api.coindesk.com/v1/bpi/currentprice.json, automate the following

1. Send the GET request
2. Verify the response contains
   a. There are 3 BPIs
      i. USD
      ii. GBP
      iii. EUR
   b. The GBP ‘description’ equals ‘British Pound Sterling’.



```
{
    "time": {
        "updated": "Jan 30, 2024 04:17:15 UTC",
        "updatedISO": "2024-01-30T04:17:15+00:00",
        "updateduk": "Jan 30, 2024 at 04:17 GMT"
    },
    "disclaimer": "This data was produced from the CoinDesk Bitcoin Price Index (U
    "chartName": "Bitcoin",
    "bpi": {
        "USD": {
            "code": "USD",
            "symbol": "&#36;",
            "rate": "43,545.741",
            "description": "United States Dollar",
            "rate_float": 43545.7412
        },
        "GBP": {
            "code": "GBP",
            "symbol": "&pound;",
            "rate": "34,259.307",
            "description": "British Pound Sterling",
            "rate_float": 34259.307
        },
        "EUR": {
            "code": "EUR",
            "symbol": "&euro;",
            "rate": "40,193.285",
            "description": "Euro",
            "rate_float": 40193.2852
        }
    }
}
```

Ans-

```
import org.apache.http.HttpResponse;

import org.apache.http.client.HttpClient;

 import org.apache.http.client.methods.HttpGet;

import org.apache.http.impl.client.HttpClientBuilder;

import org.apache.http.util.EntityUtils;

import org.json.JSONObject;

import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.*;
```

```
public class CoinDeskAPITest {
@Test public void testCoinDeskAPI() throws Exception {
// Send the GET request
HttpClient client = HttpClientBuilder.create().build();
HttpGet request = new HttpGet("https://api.coindesk.com/v1/bpi/currentprice.json");
HttpResponse response = client.execute(request);

// Get the response body as a string
String responseBody = EntityUtils.toString(response.getEntity());

//Verify the response contains
JSONObject jsonResponse = new JSONObject(responseBody);
JSONObject bpi = jsonResponse.getJSONObject("bpi");
assertTrue(bpi.has("USD"), "USD BPI is missing");
assertTrue(bpi.has("GBP"), "GBP BPI is missing");
assertTrue(bpi.has("EUR"), "EUR BPI is missing");
assertEquals(3, bpi.length(), "Expected 3 BPIs, but found " + bpi.length());

// The GBP ‘description’ equals ‘British Pound Sterling’.
JSONObject gbp = bpi.getJSONObject("GBP");
String gbpDescription = gbp.getString("description");
assertEquals("British Pound Sterling", gbpDescription, "GBP description is incorrect");
System.out.println("API test passed!");
} }
```