

## Task 01 Solve:

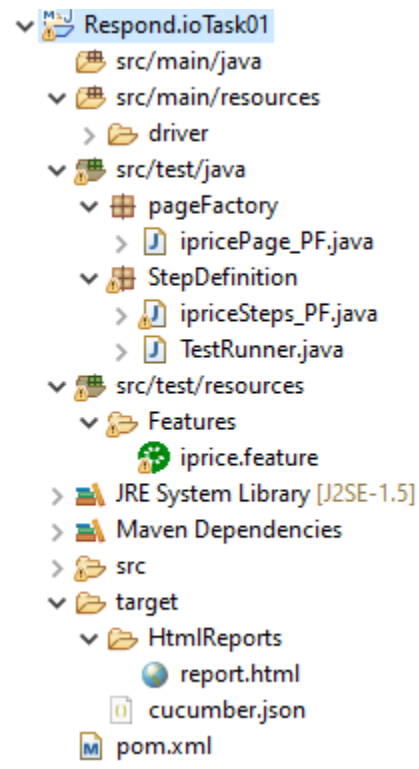
For UI Automation, I have used Selenium Maven with BDD and Page object model framework reusability and redundancy. Also I have used Page Factory technique. **It is used for initialization of Page objects or to instantiate the Page object itself.**

### Folder Structure:

Page Factory Folder → Have all the Page classes

StepDefinition Folder → Have all the Step Definition classes

Feature Folder → Have all the features



**Target folder** → have the HTML report and .json report for passed / failed test cases

### How to execute:

There are two ways to execute, 1) Run .feature file. 2) Run the Test Runner file, it will generate test cases report in .html and .json format.

→ For reporting I have used Junit reports.

<p>Execution summary</p> <p>1 scenarios</p> <p>passed</p>	<p>Duration 1m 6.978s</p> <p>Implementation cucumber-jvm - 6.4.0</p> <p>Runtime Eclipse Adoptium - 17.0.2</p> <p>OS Windows 10</p> <p>CPU amd64</p>	<input type="text" value="Some text or @tags"/> <p>You can use either plain text for the search or <a href="#">cucumber tag expressions</a> to filter the output.</p>
---	---	---

file:///C:/Users/Raza/eclipse-workspace/Respond.ioTask01/src/test/resources/Features/iprice.feature

**Feature:** Test the iprice functionality

**Scenario Outline:** Test the functionality of iprice

- Given browser is open
- And user is on iprice home page and validate it
- When user navigate to laptops page and validate it
- And user select the brand value to be Dell
- Then validate result based on selected filter
- Then user navigate to dresses page
- And user click on Price sorting until the indicator indicates
- Then Validate that the results are sorted in descending order of Price
- Then user navigate to home page
- And user Search for <item>
- And user click on Search button
- Then Validate that the result matches the search criteria

Examples:

## Task Solution 02:

I have used Cypress tool for API automation task.

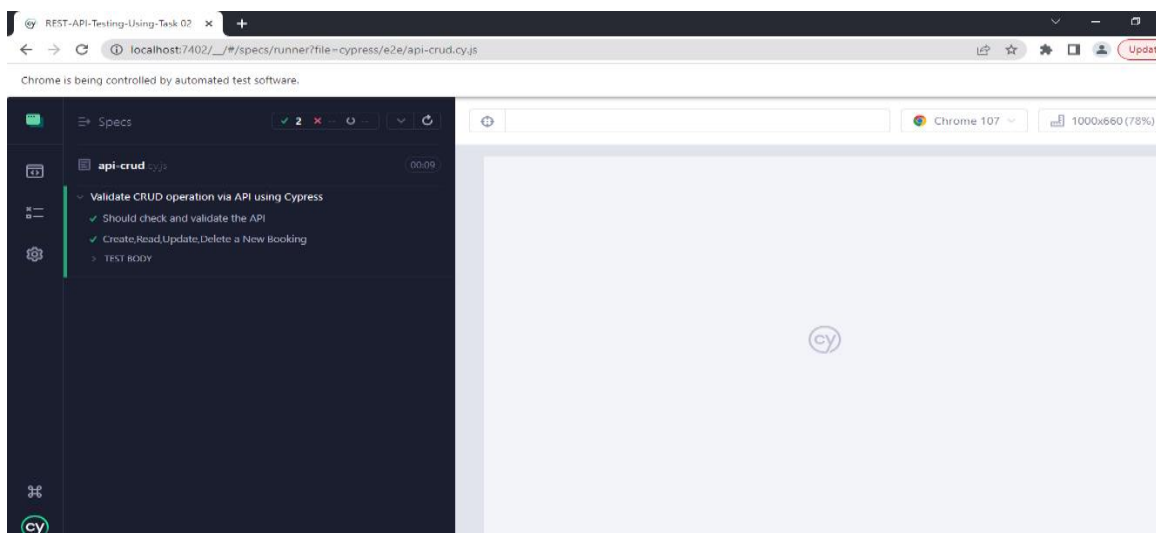
Api-crud.cy.js → file describe all the test cases in it. Fixture file have all the test data with key value pair.

### Execution of test:

On terminal write command npm run test.

Video folder have execution video

### Output:



**Task 03 Solution:**

1)

```
SELECT email, phone from Contacts where (  
    email IS NOT NULL  
    AND  
    phone IS NOT NULL  
);
```

**2)**

```
SELECT groupId, COUNT(*)  
FROM Users  
GROUP BY groupId;
```

**Second Method:**

```
SELECT group.id, COUNT(*) FROM Users  
INNER JOIN Users ON users.group_id=groups.id  
GROUP BY groups.id
```

3)

```
SELECT Customers.customerName, Orders.ordId, Orders.orderDate  
FROM Orders, Customer  
WHERE Customer.id = Orders.customerId;
```

**Thanks!**