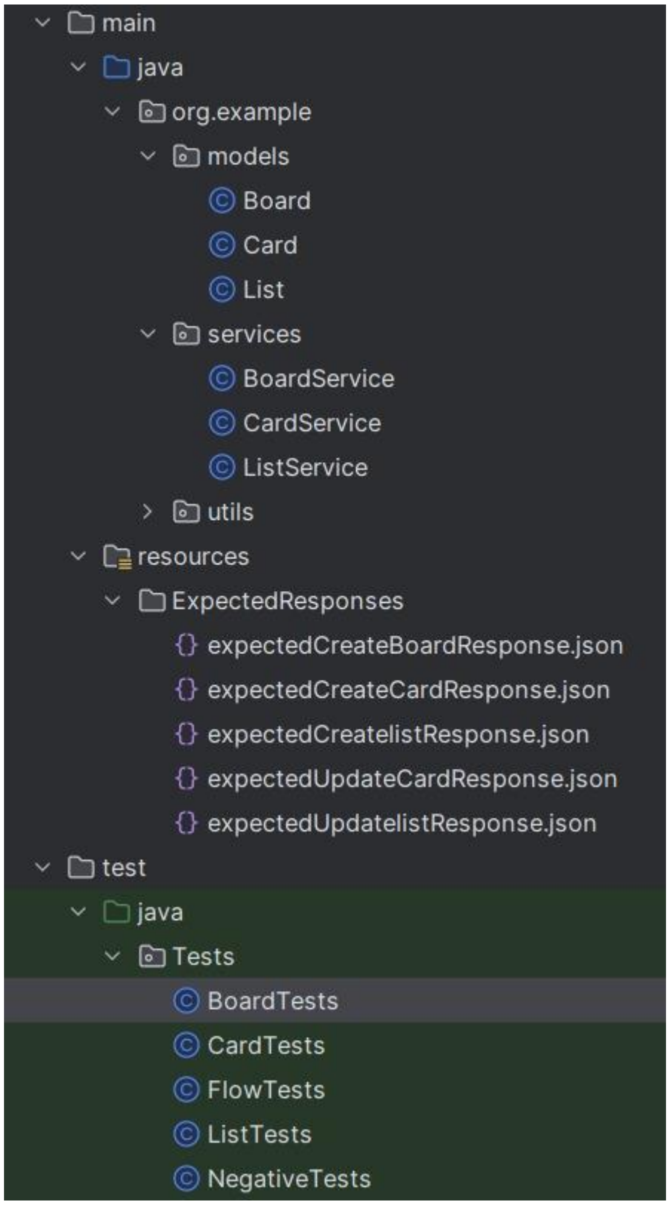
Automated API Testing of Trello API using Rest Assured

This project aims to design, execute, and automate comprehensive API tests for Trello using Rest Assured. By setting up a structured Java-based testing framework and utilizing TestNG for automation, we ensure the reliability and functionality of Trello's API endpoints.

➢ <u>Environment Setup:</u>

- We used Java project using maven as a build tool

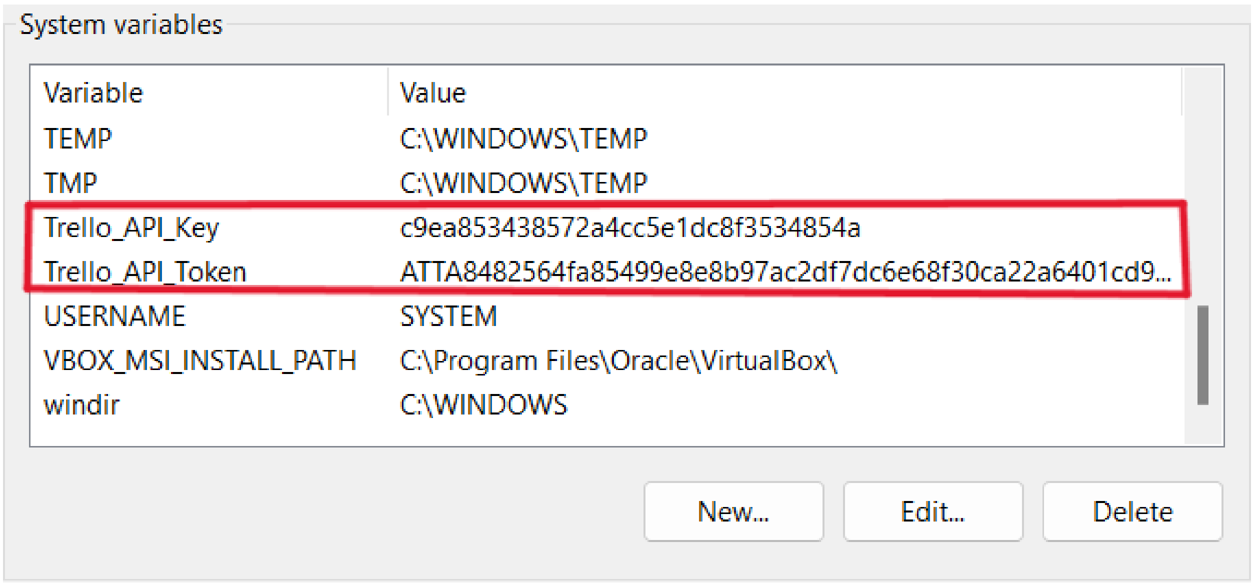


```
main
  java
    org.example
      models
        Board
        Card
        List
      services
        BoardService
        CardService
        ListService
      utils
  resources
    ExpectedResponses
      expectedCreateBoardResponse.json
      expectedCreateCardResponse.json
      expectedCreatelistResponse.json
      expectedUpdateCardResponse.json
      expectedUpdatelistResponse.json
test
  java
    Tests
      BoardTests
      CardTests
      FlowTests
      ListTests
      NegativeTests
```

- Added the necessary dependencies for rest assured, TestNG, Hamcrest and Allure in pom.xml file.

➢ Framework Setup:

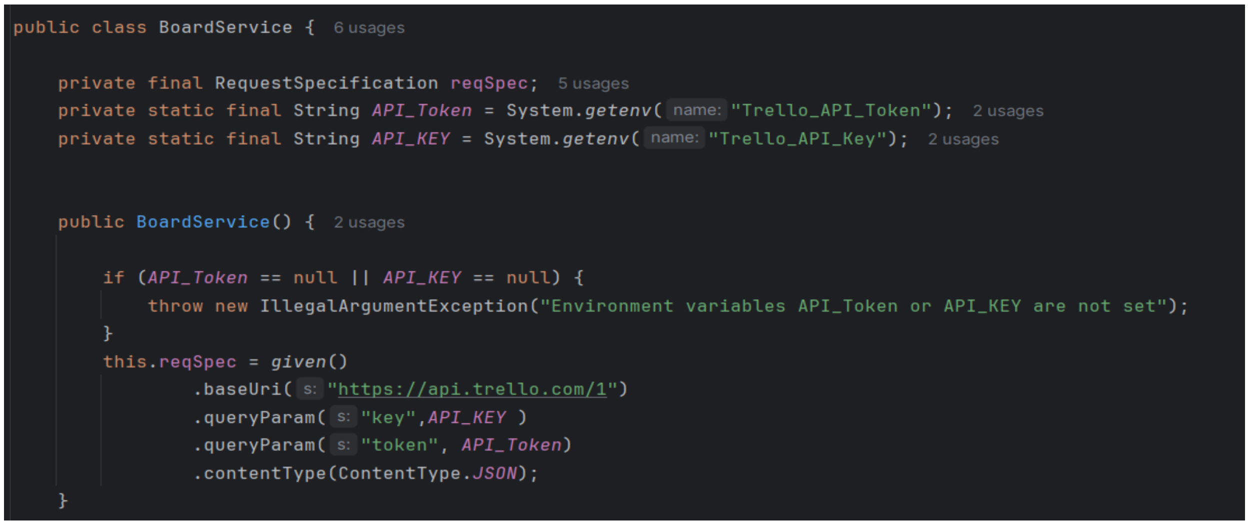
- We configured our system environment variables and put our API_key and API_token so anyone who has its own api key and token configured in his system environment variables can run the project successfully.
- Here is a screenshot for our API_key and token.



System variables

Variable	Value
TEMP	C:\WINDOWS\TEMP
TMP	C:\WINDOWS\TEMP
Trello_API_Key	c9ea853438572a4cc5e1dc8f3534854a
Trello_API_Token	ATTA8482564fa85499e8e8b97ac2df7dc6e68f30ca22a6401cd9...
USERNAME	SYSTEM
VBOX_MSI_INSTALL_PATH	C:\Program Files\Oracle\VirtualBox\
windir	C:\WINDOWS

- Here is a screenshot of how we implemented this in our code.

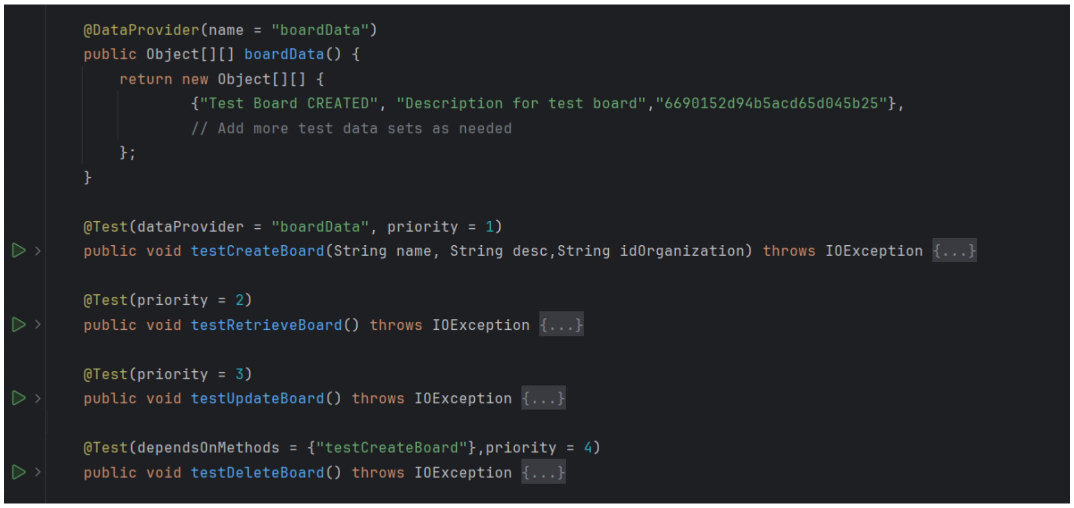


```java
public class BoardService {  6 usages

    private final RequestSpecification reqSpec;  5 usages
    private static final String API_Token = System.getenv( name: "Trello_API_Token");  2 usages
    private static final String API_KEY = System.getenv( name: "Trello_API_Key");  2 usages


    public BoardService() {  2 usages

        if (API_Token == null || API_KEY == null) {
            throw new IllegalArgumentException("Environment variables API_Token or API_KEY are not set");
        }
        this.reqSpec = given()
                .baseUri( s: "https://api.trello.com/1")
                .queryParam( s: "key",API_KEY )
                .queryParam( s: "token", API_Token)
                .contentType(ContentType.JSON);
    }
```

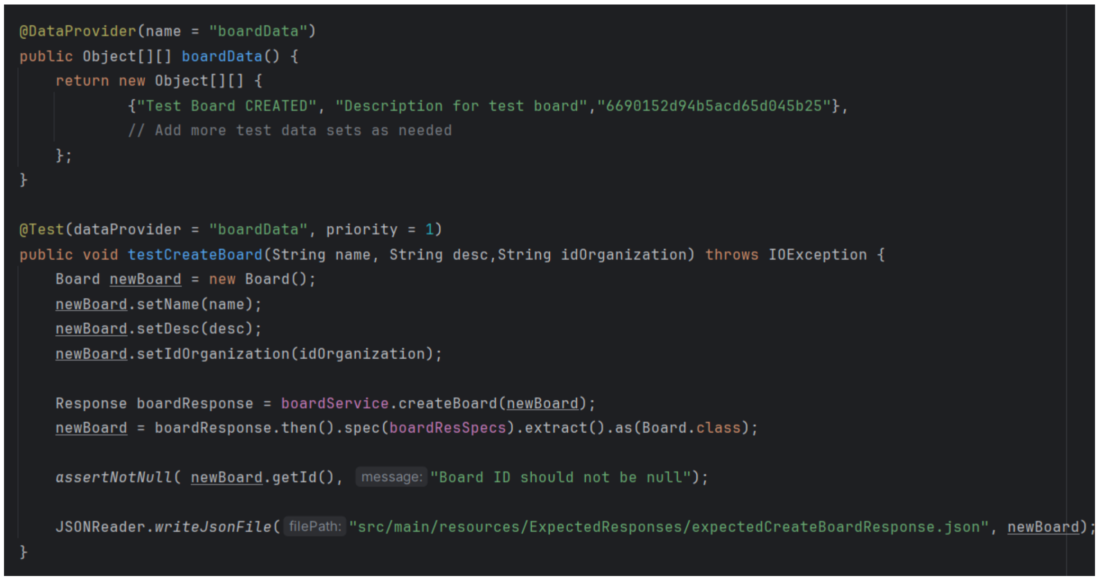
➢ Test Case Design:

We have done different test scenarios to cover

- Board management:

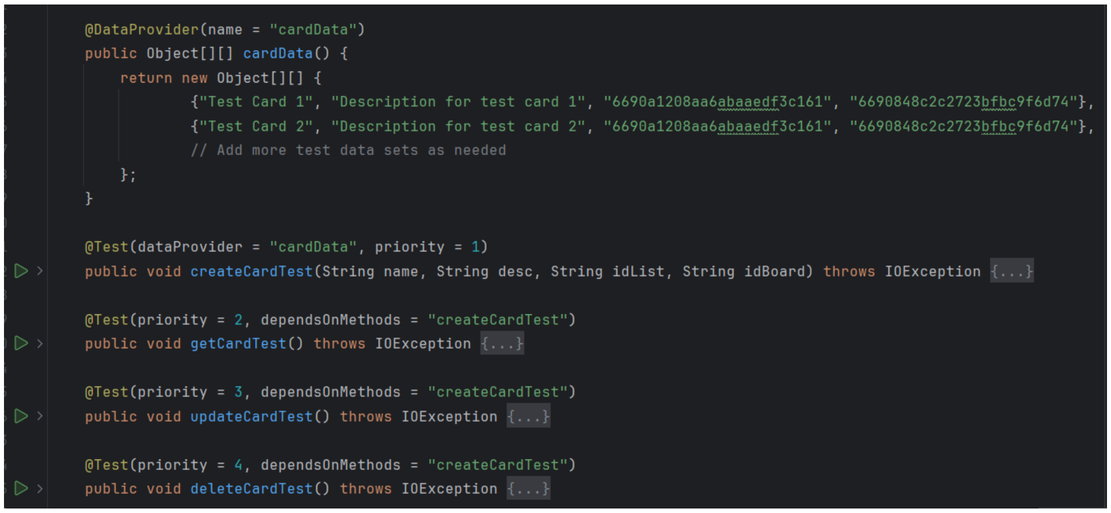


```java
@DataProvider(name = "boardData")
public Object[][] boardData() {
    return new Object[][] {
            {"Test Board CREATED", "Description for test board","6690152d94b5acd65d045b25"},
            // Add more test data sets as needed
    };
}

@Test(dataProvider = "boardData", priority = 1)
public void testCreateBoard(String name, String desc,String idOrganization) throws IOException {...}

@Test(priority = 2)
public void testRetrieveBoard() throws IOException {...}

@Test(priority = 3)
public void testUpdateBoard() throws IOException {...}

@Test(dependsOnMethods = {"testCreateBoard"},priority = 4)
public void testDeleteBoard() throws IOException {...}
```

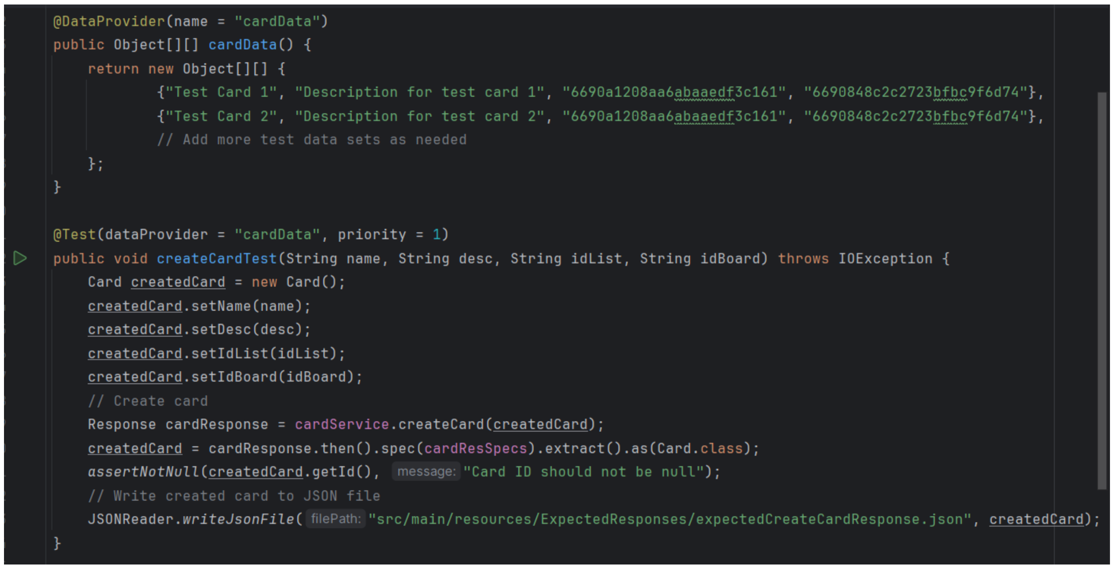


```java
@DataProvider(name = "boardData")
public Object[][] boardData() {
    return new Object[][] {
            {"Test Board CREATED", "Description for test board","6690152d94b5acd65d045b25"},
            // Add more test data sets as needed
    };
}

@Test(dataProvider = "boardData", priority = 1)
public void testCreateBoard(String name, String desc,String idOrganization) throws IOException {
    Board newBoard = new Board();
    newBoard.setName(name);
    newBoard.setDesc(desc);
    newBoard.setIdOrganization(idOrganization);

    Response boardResponse = boardService.createBoard(newBoard);
    newBoard = boardResponse.then().spec(boardResSpecs).extract().as(Board.class);

    assertNotNull( message: newBoard.getId(), "Board ID should not be null");

    JSONReader.writeJsonFile( filePath: "src/main/resources/ExpectedResponses/expectedCreateBoardResponse.json", newBoard);
}
```

- Card management:

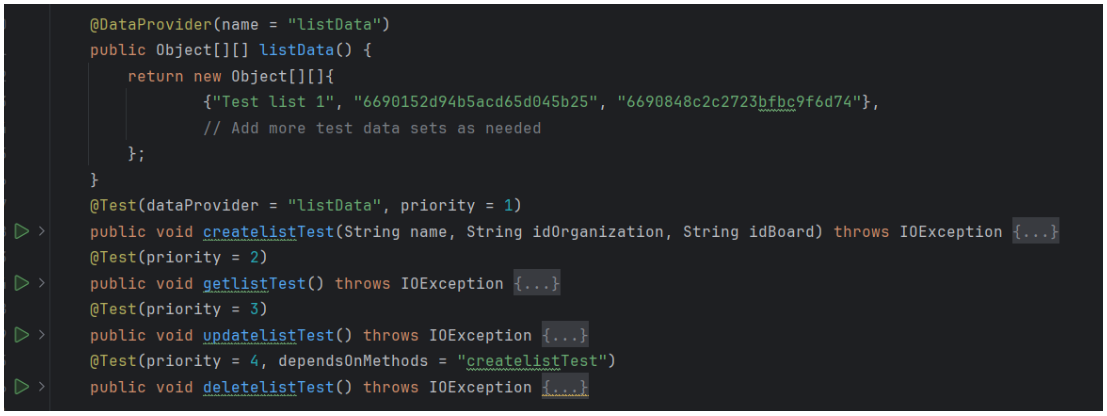


```java
@DataProvider(name = "cardData")
public Object[][] cardData() {
    return new Object[][] {
            {"Test Card 1", "Description for test card 1", "6690a1208aa6abaaedf3c161", "6690848c2c2723bfbc9f6d74"},
            {"Test Card 2", "Description for test card 2", "6690a1208aa6abaaedf3c161", "6690848c2c2723bfbc9f6d74"},
            // Add more test data sets as needed
    };
}

@Test(dataProvider = "cardData", priority = 1)
public void createCardTest(String name, String desc, String idList, String idBoard) throws IOException {...}

@Test(priority = 2, dependsOnMethods = "createCardTest")
public void getCardTest() throws IOException {...}

@Test(priority = 3, dependsOnMethods = "createCardTest")
public void updateCardTest() throws IOException {...}

@Test(priority = 4, dependsOnMethods = "createCardTest")
public void deleteCardTest() throws IOException {...}
```

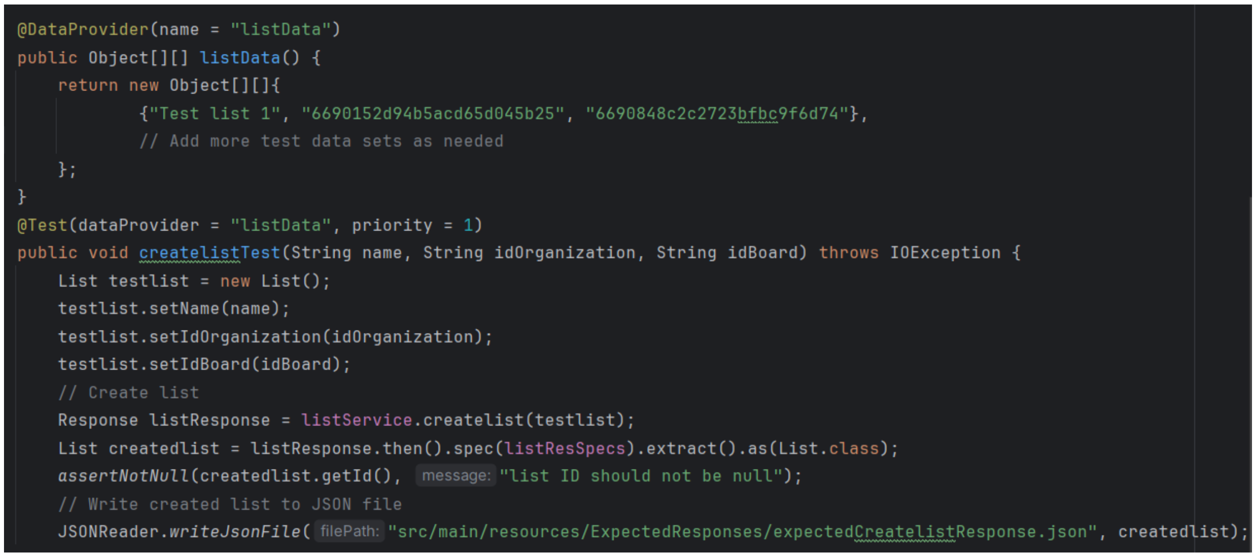


```java
@DataProvider(name = "cardData")
public Object[][] cardData() {
    return new Object[][] {
            {"Test Card 1", "Description for test card 1", "6690a1208aa6abaaedf3c161", "6690848c2c2723bfbc9f6d74"},
            {"Test Card 2", "Description for test card 2", "6690a1208aa6abaaedf3c161", "6690848c2c2723bfbc9f6d74"},
            // Add more test data sets as needed
    };
}

@Test(dataProvider = "cardData", priority = 1)
public void createCardTest(String name, String desc, String idList, String idBoard) throws IOException {
    Card createdCard = new Card();
    createdCard.setName(name);
    createdCard.setDesc(desc);
    createdCard.setIdList(idList);
    createdCard.setIdBoard(idBoard);
    // Create card
    Response cardResponse = cardService.createCard(createdCard);
    createdCard = cardResponse.then().spec(cardResSpecs).extract().as(Card.class);
    assertNotNull(createdCard.getId(), message: "Card ID should not be null");
    // Write created card to JSON file
    JSONReader.writeJsonFile( filePath: "src/main/resources/ExpectedResponses/expectedCreateCardResponse.json", createdCard);
}
```

- Lists management:

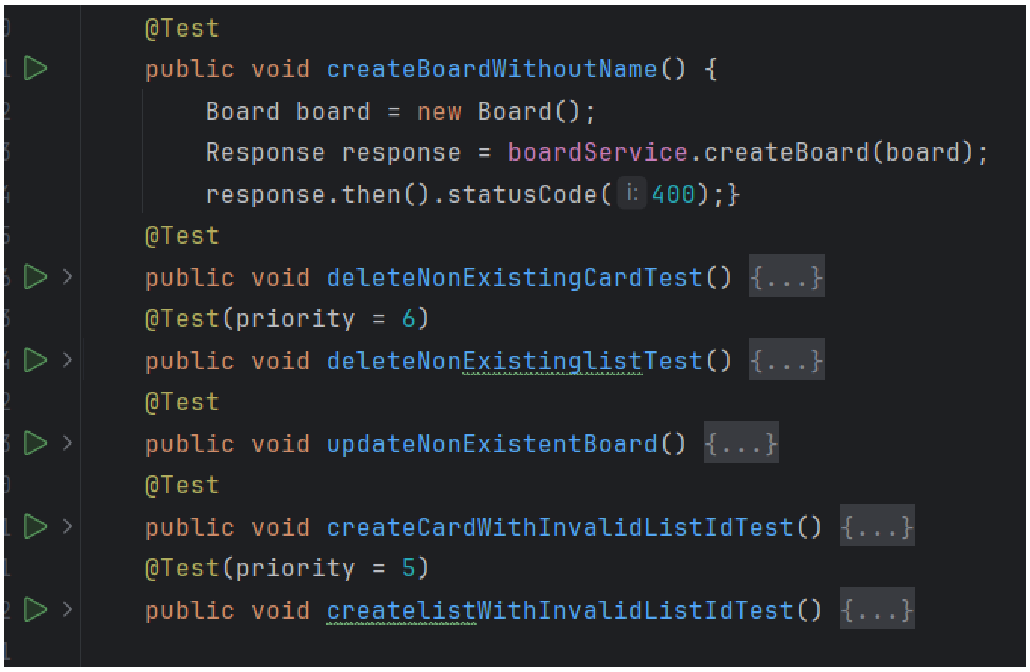


```java
@DataProvider(name = "listData")
public Object[][] listData() {
    return new Object[][]{
            {"Test list 1", "6690152d94b5acd65d045b25", "6690848c2c2723bfbc9f6d74"},
            // Add more test data sets as needed
    };
}
@Test(dataProvider = "listData", priority = 1)
public void createlistTest(String name, String idOrganization, String idBoard) throws IOException {...}
@Test(priority = 2)
public void getlistTest() throws IOException {...}
@Test(priority = 3)
public void updatelistTest() throws IOException {...}
@Test(priority = 4, dependsOnMethods = "createlistTest")
public void deletelistTest() throws IOException {...}
```



```java
@DataProvider(name = "listData")
public Object[][] listData() {
    return new Object[][]{
            {"Test list 1", "6690152d94b5acd65d045b25", "6690848c2c2723bfbc9f6d74"},
            // Add more test data sets as needed
    };
}
@Test(dataProvider = "listData", priority = 1)
public void createlistTest(String name, String idOrganization, String idBoard) throws IOException {
    List testlist = new List();
    testlist.setName(name);
    testlist.setIdOrganization(idOrganization);
    testlist.setIdBoard(idBoard);
    // Create list
    Response listResponse = listService.createlist(testlist);
    List createdlist = listResponse.then().spec(listResSpecs).extract().as(List.class);
    assertNotNull(createdlist.getId(),  message: "list ID should not be null");
    // Write created list to JSON file
    JSONReader.writeJsonFile( filePath: "src/main/resources/ExpectedResponses/expectedCreatelistResponse.json", createdlist);
```

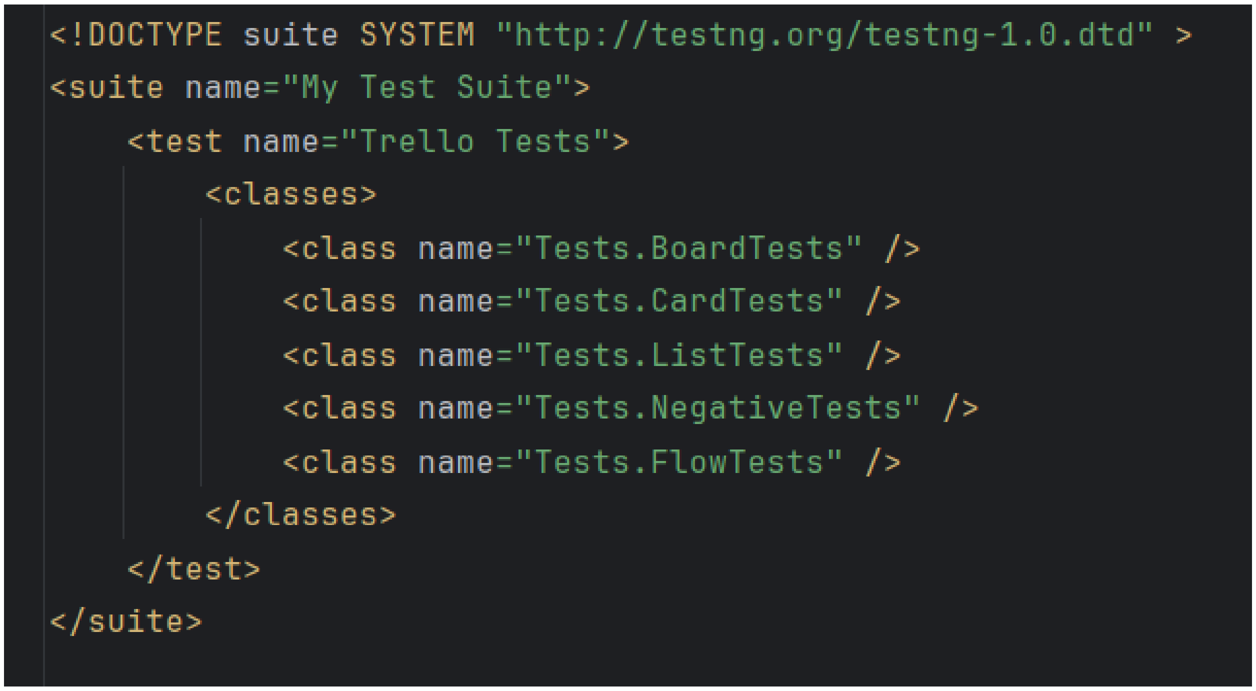
- Negative tests with invalid inputs:



```java
@Test
public void createBoardWithoutName() {
    Board board = new Board();
    Response response = boardService.createBoard(board);
    response.then().statusCode( i: 400);}
@Test
public void deleteNonExistingCardTest() {...}
@Test(priority = 6)
public void deleteNonExistinglistTest() {...}
@Test
public void updateNonExistentBoard() {...}
@Test
public void createCardWithInvalidListIdTest() {...}
@Test(priority = 5)
public void createlistWithInvalidListIdTest() {...}
```

➢ Automation Testing:

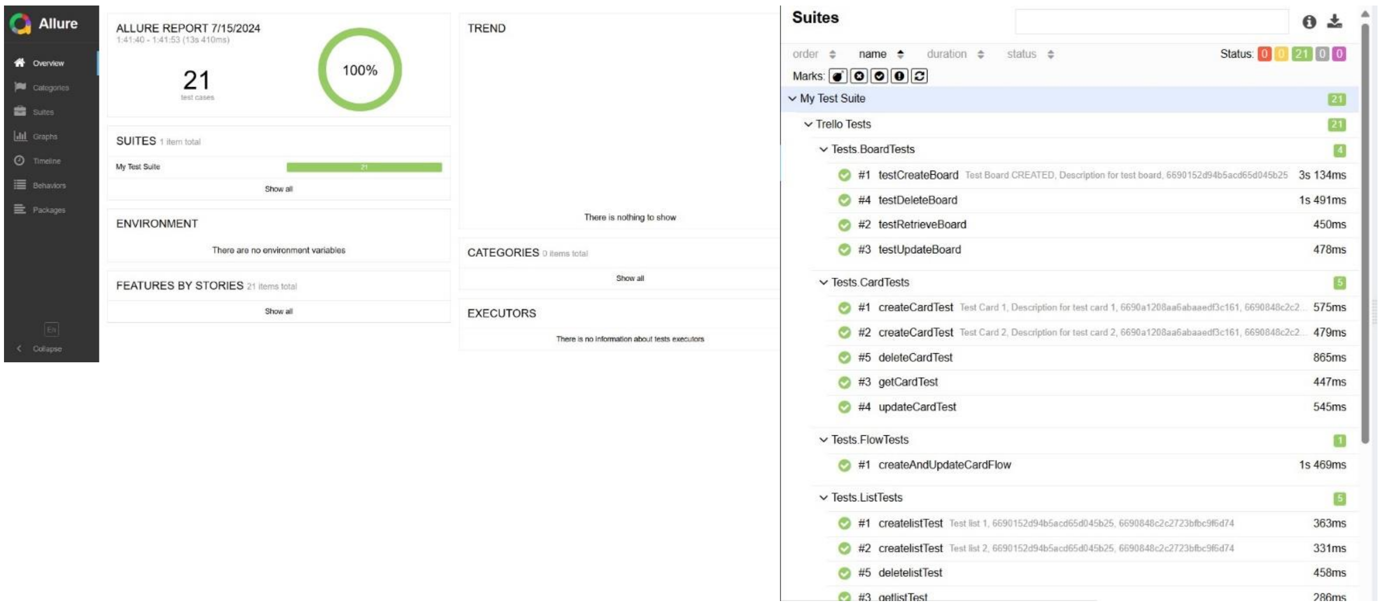
We have made a testng.xml file to use as a Test Runner so we can run the test as configured in the file.



```xml
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd" >
<suite name="My Test Suite">
    <test name="Trello Tests">
        <classes>
            <class name="Tests.BoardTests" />
            <class name="Tests.CardTests" />
            <class name="Tests.ListTests" />
            <class name="Tests.NegativeTests" />
            <class name="Tests.FlowTests" />
        </classes>
    </test>
</suite>
```

➢ Generating Reports:

We used allure to generate an HTML report to see the test results in details.



ALLURE REPORT 7/15/2024
1:41:40 - 1:41:53 (13s 410ms)
21 test cases
100%

SUITES 1 item total
My Test Suite 21
Show all

ENVIRONMENT
There are no environment variables

FEATURES BY STORIES 21 items total
Show all

TREND
There is nothing to show

CATEGORIES 0 items total
Show all

EXECUTORS
There is no information about tests executors

```
Suites
order   name   duration   status                 Status: 0 0 21 0 0
Marks:
∨ My Test Suite                                                      21
  ∨ Trello Tests                                                     21
    ∨ Tests.BoardTests                                                4
      #1 testCreateBoard  Test Board CREATED, Description for test board, 6690152d94b5acd65d045b25   3s 134ms
      #4 testDeleteBoard                                        1s 491ms
      #2 testRetrieveBoard                                         450ms
      #3 testUpdateBoard                                           478ms
    ∨ Tests.CardTests                                                 5
      #1 createCardTest  Test Card 1, Description for test card 1, 6690a1208aa6abaaedf3c161, 6690848c2c2...  575ms
      #2 createCardTest  Test Card 2, Description for test card 2, 6690a1208aa6abaaedf3c161, 6690848c2c2...  479ms
      #5 deleteCardTest                                            865ms
      #3 getCardTest                                               447ms
      #4 updateCardTest                                            545ms
    ∨ Tests.FlowTests                                                 1
      #1 createAndUpdateCardFlow                                1s 469ms
    ∨ Tests.ListTests                                                 5
      #1 createlistTest  Test list 1, 6690152d94b5acd65d045b25, 6690848c2c2723bfbc9f6d74   363ms
      #2 createlistTest  Test list 2, 6690152d94b5acd65d045b25, 6690848c2c2723bfbc9f6d74   331ms
      #5 deletelistTest                                            458ms
      #3 getlistTest                                               286ms
```