**MOBILE PLAN** **REST** **API TEST CASES**

**Subject: The list of test cases are listed below:**

**Test case id: 01**

**Test Description:** Get the Mobile plan from the Database.

**Precondition:** The Mobile plan Rest API code running in the spring tool suite on Port: 8080,

Postman is working properly.

**Testing Steps:**

1. Open Postman Tool
2. Create a New HTTP request in the tool
3. Select the GET HTTP method to create Mobile plan data in postman.
4. Enter the URL along with Port:8080 and project name http://localhost:8080/mp in the postman tool
5. Click on the ‘Send’ button
6. View the response.

**Excepted Result:**

* HTTP Response code 201 should be Created
* True

 {

        "id": 1,

        "name": "Airtel",

        "description": "High speed network",

        "validity": 28,

        "price": 299

    },

    {

        "id": 2,

        "name": "VI",

        "description": "2GB Per day",

        "validity": 56,

        "price": 399

    },

    {

        "id": 3,

        "name": "jio",

        "description": "2GB Per day and unlimited calls",

        "validity": 84,

        "price": 799

    },

    {

        "id": 4,

        "name": "BSNL",

        "description": "Get network every corner",

        "validity": 24,

        "price": 150

    }

**Actual Result:**

* Success Code: 201
* Displayed Successfully.

 {

        "id": 1,

        "name": "Airtel",

        "description": "High speed network",

        "validity": 28,

        "price": 299

    },

    {

        "id": 2,

        "name": "VI",

        "description": "2GB Per day",

        "validity": 56,

        "price": 399

    },

    {

        "id": 3,

        "name": "jio",

        "description": "2GB Per day and unlimited calls",

        "validity": 84,

        "price": 799

    },

    {

        "id": 4,

        "name": "BSNL",

        "description": "Get network every corner",

        "validity": 24,

        "price": 150

    }

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Test case id: 02**

**Test Description:** Get only specific Mobile plan details presented in the Database.

**Precondition**: The Mobile plan Rest API code running in the spring tool suite on Port: 8080,

Postman is working properly.

**Testing Steps:**

1. Open Postman Tool
2. Create a New HTTP request in the tool
3. Select the Get HTTP method to get data in postman
4. Enter the URL along with Port:8080, project

name, and a specific ID http://localhost:8080/mp/1 in the postman tool

1. Select the ‘raw’ and ‘JSON’ output body in postman.
2. Click on the ‘Send’ button
3. View the response

**Excepted Result:**

* HTTP Response code 200 (OK)
* True -> (Specific record will encounter of given ID)

{

    "id": 1,

    "name": "Airtel",

    "description": "High speed network",

    "validity": 28,

    "price": 299

}

**Actual Result:**

* Success Code: 200 (OK)
* Executed Successfully
* Record with ID: 1 will be returned in JSON format.

{

    "id": 1,

    "name": "Airtel",

    "description": "High speed network",

    "validity": 28,

    "price": 299

}

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Test case id: 03**

**Test Description:** Get only specific Mobile plan details that are not present in the Database

**Precondition:** The Bookshelf Rest API code running in the spring tool suite on Port: 8080,

Status: Postman is working properly.

**Testing Steps:**

1. Open Postman Tool
2. Create a New HTTP request in the tool
3. Select the Get HTTP method to get data in postman
4. Enter the URL along with Port:8080, project name, and a specific ID http://localhost:8080/mp/1000 in the postman tool
5. Select the ‘raw’ and ‘JSON’ output body in postman.
6. Click on the ‘Send’ button
7. View the response

**Excepted Result:**

* HTTP Response code 404 is displayed .
* False.
* Not found Message should be displayed.

**Actual Result:**

* Success Code: HTTP Response code 404 is displayed
* False.
* Not found Message displayed.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Test case id: 04**

**Test Description:** Adding a Mobile plan object, in the Database.

**Precondition:** The Mobile plan Rest API code running in the spring tool suite on Port: 8080,

Postman is working properly.

**Testing Steps:**

1. Open Postman Tool

2. Create a New HTTP request in the tool.

3. Select the Post HTTP method to create Mobile plan data in postman.

4. Enter the URL along with Port:8080 and project name http://localhost:8080/mp in the postman tool.

5. Select the message body to be sent as a request in postman.

6. Select ‘raw’ and ‘JSON’ in the request message body in postman.

7. Write the Mobile plan data in the message body in ‘JSON’ format Data look like

{

    "id": 3,

    "name": "jio”.

    "description": "2GB Per day and unlimited calls",

    "validity": 84,

    "price": 799

}

8. Click on the ‘Send’ button.

9. View the response.

**Excepted Result:**

* HTTP Response code 200 (OK)
* True -> (All records will be posted in the database).

**Actual Result:**

* Success Code: 200 (OK)
* Executed Successfully
* All Records will be posted in the database.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Test case id: 05**

**Test Description:** Adding a Mobile plan object in a database the plan is already present.

**Precondition:** The Mobile plan Rest API code running in the spring tool suite on Port: 8080, Postman is working properly.

**Testing Steps:**

1. Open Postman Tool

2. Create a New HTTP request in the tool.

3. Select the Post HTTP method to create Mobile plan data in postman.

4. Enter the URL along with Port:8080 and project name http://localhost:8080/mp in the postman tool.

5. Select the message body to be sent as a request in postman.

6. Select ‘raw’ and ‘JSON’ in the request message body in postman.

7. Write the already present Mobile plan data in the message body in ‘JSON’ format Data look like this.

{

    "id": 3,

    "name": "jio",

    "description": "2GB Per day and unlimited calls",

    "validity": 84,

    "price": 799

}

8. Click on the ‘Send’ button.

9. View the response.

**Excepted Result:**

* HTTP Response code 404 (Not found)
* False,
* The message should be displayed, ERROR: DUPLICATE ENTRY NOT ALLOW

**Actual Result:**

* code 404 (Not found)
* False
* Message displayed, ERROR: DUPLICATE ENTRY NOT ALLOW

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Test case id:** **06**

**Test Description:** Update Mobile plan details that are present in the Database.

**Precondition:** The Mobile plan Rest API code running in the spring tool suite on Port: 8080,

Postman is working properly.

**Testing Steps:**

1. Open Postman Tool
2. Create a New HTTP request in the tool
3. Select the PATCH HTTP method to get data in postman
4. Enter the URL along with Port:8080 and project name http://localhost:8080/mp in the postman tool
5. Change the existing data fields that need to change.
6. Click on the ‘Send’ button
7. View the response

**Excepted Result:**

* HTTP Response code 200 (OK)
* True -> (Record exists and should be updated)

**Actual Result:**

* Success Code: 200 (OK)
* Executed and Updated Successfully.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Test case id:** **07**

**Test Description:** Update Mobile plan details that are not present in the Database.

**Precondition:** The Mobile plan Rest API code running in the spring tool suite on Port: 8080,

Postman is working properly.

**Testing Steps:**

1. Open Postman Tool
2. Create a New HTTP request in the tool
3. Select the PATCH HTTP method to get data in postman
4. Enter the URL along with Port:8080 and project name http://localhost:8080/mp in the postman tool
5. Change the data fields that are Not present in the database.

6. Click on the ‘Send’ button

7. View the response.

**Excepted Result:**

* HTTP Response code 404 (Not Found)
* False -> (Error message should be displayed).

**Actual Result:**

* HTTP Response 404 (Not Found)
* False->ERROR: Data with an id -> does not exist.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Test case id**: **08**

**Test Description**: Delete only specified Mobile plan details presented in the database

**Precondition**: The Mobile plan Rest API code running in the spring tool suite on Port: 8080,

Status: Postman is working properly.

Testing Steps:

1. Open Postman Tool
2. Create a New HTTP request in the tool
3. Select the Get HTTP method to get data in postman
4. Enter the URL along with Port:8080 and project name with specific ID http://localhost:8080/mp/6 in the postman tool.
5. Click on the ‘Send’ button
6. View the response

**Excepted Result:**

* HTTP Response code 200 (OK)
* True -> (Specific record is deleted of given ID)

**Actual Result:**

* Success Code: 200 (OK)
* Executed Successfully
* Record with ID: 6 deleted

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Test case id: 09**

**Test Description**: Delete specific book details which are not present in the HashMap

**Precondition**: The Bookshelf Rest API code running in the spring tool suite on Port: 8080,

Status: Postman is working properly.

Testing Steps:

1. Open Postman Tool
2. Create a New HTTP request in the tool
3. Select the Get HTTP method to get data in postman
4. Enter the URL along with Port:8080, project name, and specify ID http://localhost:8080/mp /10 in the postman tool
5. Select ‘raw’ and ‘JSON’ in the request output body in postman
6. Click on the ‘Send’ button
7. View the response

Excepted Result:

* HTTP Response code 404 (Not Found)
* False-> (Error message should be displayed).

Actual Result:

* Success Code: 404 (Not Found)
* False
* ID Not Found

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Test case id: 10**

**Test Description:** Delete the Mobile plan from the Database.

**Precondition:** The Mobile plan Rest API code running in the spring tool suite on Port: 8080,

Postman is working properly.

**Testing Steps:**

1. Open Postman Tool
2. Create a New HTTP request in the tool
3. Select the Delete HTTP method to delete Mobile plan data in postman.
4. Enter the URL along with Port:8080 and project name http://localhost:8080/mp in the postman tool
5. Click on the ‘Send’ button
6. View the response.

**Excepted Result:**

* HTTP Response code 201 should be Created
* True

**Actual Result:**

* Success Code: 201
* All plans Successfully deleted

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**