

SUMMARY OF PROJECT

Title: Journey of Customer Banking registration with Rest APIs.

Project Scope: Goal of the project is to create REST APIs for the Customer Registration in banking domain using Spring MVC.

Project Description: I have created a Spring MVC maven project which consists of three APIs to demonstrate the working of Customer registration.

- 1) First API will give combination of Customer details, Account List and Nominee Relationship data based on the Customer Id. If Customer is already registered, then API will return existing customer registration details.
- 2) Second API will save the new Customer registration details. Also it will return validation message if customer is already registered based on the customer Id.
- 3) Third API will retrieve all the registered requests.

Package Details: In this project I have created 5 packages naming Controller, service, Dao , Model, Util.

- **Controller package:** It has ErrorController for displaying messages as per error code and RequestController as frontend controller which consists registration apis
- **Service package:** Includes ServiceException class to handle customized exceptions at controller and service layer. Also consists RequestServiceImpl class
- **Dao Package:** Includes DaoException class to handle customized exceptions at dao layer and RequestDao class to fetch and store the data in mysql database using jdbc template

- **Model package:** It consist Request, Account, Customer POJO classes
- **Util package:** It consists of Constant class to declare constants, MessageLogger class to log messages and PayloadStatus enum class
- **Resources folder:** System.properties to configure database details, Sql.properties consists of queries and log4j.xml to configure logs settings

Technologies used in project: Java 8, Spring Framework 5.2.4, MySql, Spring MVC, REST APIs

Tools used - STS, Postman(To test APIs).

REST APIs: Following are the request and response JSON details of the APIs after testing them in the Postman. Below I have added exact result which reflected in Postman console while running it as per the above APIs.

1) READ API to fetch customer, account and nominee relationship data against customer id:

Url: <http://localhost:8080/spring-mvc/requests/?action=read&customerId=1>

Method: POST

Response JSON:

Success Response

```
{
  "success": true,
  "msg": "Request data retrieved successfully for customer id = 1",
  "status": 200,
  "result": {
    "nomineeRelationship": [
      {
        "id": "1",
        "description": "Mother"
      },
      {
        "id": "2",
        "description": "Father"
      }
    ],
    "name": "Bishop",
    "accounts": [
      {
        "accountNo": "10000090",
        "micrCode": "879890K",
        "ifscCode": "KKBK09888",
        "customerId": "1"
      },
      {
        "accountNo": "800098900",
        "micrCode": "879890K",
        "ifscCode": "KKBK09888",
        "customerId": "1"
      }
    ]
  },
  "totalAvailableRecords": 0,
  "header": null,
  "broadcast": null
}
```

Error response:

```
{
  "success": false,
  "msg": "Could not fetch request details",
  "status": -1,
  "result": "Could not fetch request details",
  "totalAvailableRecords": 0,
  "header": null,
  "broadcast": null
}
```

2) CREATE API to save registration details in request table :

Url: <http://localhost:8080/spring-mvc/requests/?action=create>

Method: POST

Request Body:

```
{
  "bankName": "Citibank",
  "branchName": "Mumbai",
  "micrCode": "600037002",
  "ifscCode": "CITI0000003",
  "accountNo": "34645645",
  "customerId": "100231",
  "title": "Mr",
  "name": "John Snow",
  "gender": "M",
  "dob": "10/09/1998",
  "address": "Mumbai",
  "mobileNo": "9655567890",
  "emailId": "john@gmail.com",
  "aadharNo": null,
  "nomineeName": "Jack Snow",
  "nomineeAadharNo": null,
  "nomineeRelationship": "father",
  "isNomineeMinor": "Yes",
  "gaurdianName": "Jill",
  "pensionAmt": "1000",
  "monthlyContribution": "100",
  "age": 25,
  "contributionPeriod": "6",
  "panNumber": "AGPAY4562R"
}
```

Success response:

```
{
  "success": true,
  "msg": "Request data saved successfully",
  "status": 200,
  "result": {
    "requestId": "1"
  },
  "totalAvailableRecords": 0,
  "header": null,
}
```

```
    "broadcast": null
}
```

Error response:

```
{
  "success": false,
  "msg": "Request is already registered for this customer",
  "status": -1,
  "result": "Request is already registered for this customer",
  "totalAvailableRecords": 0,
  "header": null,
  "broadcast": null
}
```

3) READ API to fetch all registered customer requests

Url: <http://localhost:8080/spring-mvc/requests/all?action=read>

Method: GET

Response Json:

Success Response:

```
{
  "success": true,
  "msg": "Requests retrieved successfully ",
  "status": 200,
  "result": [
    {
      "id": 1,
      "bankName": "Citibank",
      "branchName": "Mumbai",
      "micrCode": "600037002",
      "ifscCode": "CITI0000003",
      "accountNo": "34645645",
      "customerId": "100231",
      "title": "Mr",
      "name": "John Snow",
      "gender": "M",
      "dob": "10/09/1998",
      "address": "Mumbai",
      "mobileNo": "9655567890",
      "emailId": "john@gmail.com",
      "aadharNo": null,
      "nomineeName": "Jack Snow",
      "nomineeAadharNo": null,
      "nomineeRelationship": "father",
      "isNomineeMinor": "Yes",
      "gaurdianName": "Jill",
      "pensionAmt": "1000",
      "monthlyContribution": "100",
      "age": null,
      "contributionPeriod": "6",
      "panNumber": "AGPAY4562R"
    }
  ],
}
```

```
"totalAvailableRecords": 0,  
"header": null,  
"broadcast": null  
}
```

Error Response:

```
{  
  "success": false,  
  "msg": " Application encountered invalid parameters. Please contact  
your system administrator to report this issue at web server.",  
  "status": 400,  
  "result": "",  
  "totalAvailableRecords": 0,  
  "header": null,  
  "broadcast": null  
}
```

Conclusion: As per my previous experience in Banking Domain I have made a simple Replica of Customer Registration by using above three APIs.