**Itinerary-Planning**

**Auth-Service**

TABLE OF CONTENTS

[1. Introduction 2](#_Toc204703403)

[2. Tech Stack 3](#_Toc204703404)

[3. Data Dictionary 4](#_Toc204703405)

[3.1.1 UserVerification 4](#_Toc204703406)

[4. REST API Endpoints 5](#_Toc204703407)

[4.1.1 /register 6](#_Toc204703408)

[4.1.2 /login 8](#_Toc204703409)

[4.1.3 /verify/{id}/{token} 10](#_Toc204703410)

[5.Resilience4j Configuration 11](#_Toc204703411)

[5.1.1. Circuit Breaker 11](#_Toc204703412)

[5.1.2 Rate Limiter 11](#_Toc204703413)

[6. Project Structure 12](#_Toc204703414)

# 1. Introduction

The Auth Serviceis a core microservice in a banking ecosystem responsible for user registration, login. reset password, mobile number verification and email verification

It exposes RESTful APIs for internal/external communication and interacts with other services via Kafka and RestTemplate.

**Key Features**  
  
1. Registration  
2 . Email Verification  
3. Login  
4. Reset Password  
5. Mobile Number Verification

# 2. Tech Stack

|  |  |
| --- | --- |
| Technology | Version |
| JDK | 17 |
| Maven | 3.6.3 |
| Spring Boot | 3.5.3 |
| PostgreSQL | 17.5 |
| Kafka | 7.4.3 |
| Zookeeper | 7.4.3 |
| Zipkin | 2.23 |
| KeyCloak | 24.0.3 |

# 3. Data Dictionary

Database: users

|  |  |  |
| --- | --- | --- |
| No | Table Name | Description |
| 1 | UserVerification | Holds user email verification details |
| 2 | Users | Hold user details. |
| 3 | User\_OTP | Hold mobile OTP details. |

## 3.1.1 UserVerification

**Table Description:** The UserVerification table stores user email verification data, including user ID, email, token details, verification status, and timestamps. It supports managing and validating user registrations.

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Description | Sample Value |
| userId | String | Primary key; references the user being verified | johndoe |
| email | String | Email address of the user | [john.doe@example.com](mailto:john.doe@example.com) |
| mobile | String | Mobile number of the user | +1234567890 |
| token | String | Verification token (e.g. for email/mobile verification) | a7f9e3b2c0f14367 |
| type | String | Type of verification (e.g. "EMAIL", "MOBILE") | EMAIL |
| isCheck | Boolean | Indicates whether verification was checked | true |
| isVerified | boolean | Indicates if the user is fully verified | true |
| isVerifiedMobile | boolean | Indicates if the mobile number is verified | false |
| otp | Integer | OTP sent for verification | 849302 |
| sendAt | LocalDateTime | Date and time the verification/OTP was sent | 2025-08-21T13:50:00 |
| expiryDate | LocalDateTime | Expiry time of the verification or OTP | 2025-08-21T14:20:00 |

## 3.1.2 Users

**Table Description:**  
The users table stores core user profile data, including usernames, names, contact details, account status, and role information. It supports authentication, authorization, and general user management features within the system.

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Description | Sample Value |
| username | String | Unique identifier for user (used as primary key) | johndoe |
| uid | String | User's unique ID (possibly for internal use) | 123e4567-e89b-12d3-a456 |
| roles | String | Roles assigned to the user | ADMIN,USER |
| firstName | String | User's first name | John |
| lastName | String | User's last name | Doe |
| email | String | Unique email address of the user | [john.doe@example.com](mailto:john.doe@example.com) |
| phoneNumber | String | Unique phone number of the user | +1234567890 |
| gender | String (Enum) | User's gender | MALE |
| isActive | boolean | Indicates if the user account is active | true |
| isVerifiedMobile | boolean | Indicates if the mobile number is verified | true |
| isVerifiedEmail | boolean | Indicates if the email address is verified | false |
| otp | List | One-to-many relation with UserOtp entity | [Otp1, Otp2, ...] |

## 3.1.3 OTP

The UserOtp table stores One-Time Password (OTP) records sent to users for authentication or verification purposes. It tracks when the OTP was sent, its expiration, and whether it was verified. It maintains a many-to-one relationship with the user who received the OTP.

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Description | Sample Value |
| id | Long | Primary key; unique ID for the OTP record | 101 |
| OTP | String | One-Time Password sent to the user | 849302 |
| sentDate | LocalDateTime | Date and time when the OTP was sent | 2025-08-21T14:30:00 |
| expiryDate | LocalDateTime | Date and time when the OTP will expire | 2025-08-21T14:35:00 |
| verificationTime | LocalDateTime | Date and time when the OTP was verified | 2025-08-21T14:32:10 |
| user | User | Reference to the user who received the OTP | johndoe (or User object ref) |

# 4. REST API Endpoints

**Base URL:** http://localhost:8222/api/v1/auth

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Method | Endpoint | Description |
| 1 | POST | /register | Registers a new user with validation. Returns error if input is invalid. |
| 2 | POST | /login | Authenticates user credentials and returns a token on success. |
| 3 | GET | /verify/{id}/{token} | Verifies user account using ID and verification token. |
| 4 | POST | /verify/mobile/{otp} | Verifies the user's mobile number using the provided OTP and Authorization token. |
| 5 | GET | |  | | --- | |  |  |  | | --- | | /username | | Retrieves the username of the authenticated user from the Authorization header. |
| 6 | POST | /reset/{username} | Sends a password reset link to the user identified by the given username. |
| 7 | POST | /reset/{uid}/{token} | Resets the user's password using the UID, reset token, and new password details. |

## 4.1.1 /register

This endpoint allows users to add an interest rate associated with a specific loan type.

|  |  |
| --- | --- |
| Field | Value |
| Method | POST |
| URL | http://localhost:8222/api/v1/auth/register |
| Authorization | - |
| Auth Role | - |
| Request Body | JSON |
| DTO | RegisterRequest |
| Service Dependency | Yes |

**Service Dependency**:  
The /register endpoint depends on Keycloak for user identity management and authentication.

* **External Service**: Keycloak
* **Purpose**:  
  To create a new user in the Keycloak identity provider as part of the registration process.  
  After successful creation, additional user data may be stored in the local database.

**Body** raw (JSON)

{

username: string,

email: string,

password: string,

firstName: string,

lastName: string

}

example

**Example**

{

**"username":"dhruv",**

**"email":"dhruvpatel22082020@gmail.com",**

**"password":"123123",**

**"firstName":"Dhruv",**

**"lastName":"Patel"**

**}**

**Response**

{

**status : string**

**message : string**

**object : any**

}

## 4.1.2 /login

This endpoint is used to authenticate users by verifying their credentials. Upon successful authentication, the server will provide a response that typically includes a token or user information, which can be used for subsequent requests that require authentication.

|  |  |
| --- | --- |
| Field | Value |
| Method | POST |
| URL | http://localhost:8222/api/v1/auth/login |
| Authorization | - |
| Auth Role | - |
| Request Body | JSON |
| DTO | RegisterRequest |
| Service Dependency | Yes |

**Service Dependency**:  
The /register endpoint depends on Keycloak for user identity management and authentication.

* **External Service**: Keycloak
* **Purpose**:  
  To create a new user in the Keycloak identity provider as part of the registration process.  
  After successful creation, additional user data may be stored in the local database.

**Body** raw (JSON)

{

**username : string,**

**password : string**

}

* example

**Example**

{

    "username": "dhruv",

    "password": "123123"

}

}

**Response**

{

**status : string**

**message : string**

**object : any**

}

**Response Example**

{

"status": "200",

"message": "Login Success",

"object": {

"token": "eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJQMFdGMy1yTG41azh3NnFxYVVEQWo5S1pxcTQwTlA2OHdOd1lwVFZ0Q3VjIn0.eyJleHAiOjE3NTM3ODIxNzAsImlhdCI6MTc1Mzc4MTU3MCwianRpIjoiMTQ4Y2Q0MTYtMTRhOS00hbmFnZS1heaWrnjX2ttdqYALmlYWRtRsBv42jEkFIubGq29LKBnwZ5QOA8-bcOVEyLIUgC5nzLtd1xL3SFqwQDVxnPn9vevua8k3o1ok2C\_5Qh2j8DNyNAKGL7Nanm0Y56O91MTdUPQhq8UO0LFZ7vhRQYI6dMVJNhP7mA"

}

}

## 4.1.3 /verify/{id}/{token}

This endpoint is used to verify a user's authentication status based on a unique user identifier and a verification token.

|  |  |
| --- | --- |
| Field | Value |
| Method | GET |
| URL | http://localhost:8222/api/v1/auth/verify/{id}/{token} |
| Authorization | - |
| Auth Role | - |
| Request Body | - |
| Request Parameter | **Id:** user-id  **token:** unique token  **Example:**  **id:** 5cc6f578-8646-4957-81eb-69043c6b4bf6  **token:** 890cde12-8782-40d9-a681-bfc559231881 |
| Service Dependency | No |

**Response**

{

**status : string**

**message : string**

**object : any**

}

**Response Example**

{

"status": "204 NO\_CONTENT",

"message": "email verified",

"object": null

}

## 4.1.4 /verify/mobile/{otp}

This endpoint is used to verify a user's mobile number using the provided OTP and authorization token.

|  |  |
| --- | --- |
| Field | Value |
| Method | POST |
| URL | http://localhost:8222/api/v1/auth/verify/mobile/{otp} |
| Authorization | Required (Authorization header with bearer token) |
| Auth Role | User |
| Request Body | - |
| Request Parameter | otp: OTP code sent to user's phone |
| Example | otp: 849302 |
| Service Dependency | No |

**Response Example**

{

"status": "204 NO\_CONTENT",

"message": "mobile verified",

"object": null

}

4.1.5 /usernameThis endpoint is used to retrieve the username of the currently authenticated user using their token.

|  |  |
| --- | --- |
| Field | Value |
| Method | **GET** |
| URL | **http://localhost:8222/api/v1/auth/username** |
| Authorization | **Required (Authorization header with bearer token)** |
| Auth Role | **User** |
| Request Body | **-** |
| Request Parameter | **-** |
| Example | **Header: Authorization: Bearer eyJhbGciOi...** |
| Service Dependency | **Yes – depends on helperService.getUserName(...)** |

**Response Example**

John\_jow

## 4.1.3 /reset/{username}

This endpoint sends a password reset link to the registered email address of the given username.

|  |  |
| --- | --- |
| Field | Value |
| Method | POST |
| URL | http://localhost:8222/api/v1/auth/reset/{username} |
| Authorization | - |
| Auth Role | - |
| Request Body | - |
| Request Parameter | username: Registered username |
| Example | username: john\_doe |
| Service Dependency | No |

## 4.1.7 /reset/{uid}/{token}

This endpoint is used to reset the user's password using a verification token and unique user ID.

|  |  |
| --- | --- |
| Field | Value |
| Method | POST |
| URL | http://localhost:8222/api/v1/auth/reset/{uid}/{token} |
| Authorization | - |
| Auth Role | - |
| Request Body | NewPasswordDTO (JSON with new password fields) |
| Request Parameter | uid: user identifiertoken: reset token |
| Example | uid: 5cc6f578-8646-4957-81eb-69043c6b4bf6<br>token`: 890cde12-8782-40d9-a681-bfc559231881 |
| Service Dependency | Yes – depends on authService.resetPassword(...) |

**Request Body:**

{

    "password":"abc@1234",

    "confirmPassword":"abc@1234"

}

## 

**Response Body:**

{

"status": "204 NO\_CONTENT",

"message": "password reset success",

"object": null

}

# 5.Resilience4j Configuration

## 5.1.1. Circuit Breaker

|  |  |  |
| --- | --- | --- |
| Property | Value | Description |
| register-health-indicator | true | Enables health check for this circuit breaker. |
| event-consumer-buffer-size | 10 | Size of the buffer for circuit breaker events. |
| failure-rate-threshold | 50 | Percentage of failed calls to trip the circuit breaker. |
| minimum-number-of-calls | 5 | Minimum number of calls before calculating failure rate. |
| automatic-transition-from-open-to-half-open-enabled | true | Enables automatic transition to half-open state. |
| wait-duration-in-open-state | 7s | Wait time before transitioning from open to half-open state. |
| permitted-number-of-calls-in-half-open-state | 3 | Number of test calls permitted in half-open state. |
| sliding-window-size | 10 | Size of the sliding window. |
| sliding-window-type | COUNT\_BASED | Sliding window is based on number of calls. |

## 5.1.2 Rate Limiter

|  |  |  |
| --- | --- | --- |
| Property | Value | Description |
| limit-refresh-period | 5s | Duration after which available calls are refreshed. |
| limit-for-period | 10 | Maximum number of calls allowed during each refresh period. |
| timeout-duration | 1s | Maximum time to wait for a permit before failing the call. |

# 6. Project Structure

Auth/

├── src/

│ ├── main/

│ │ ├── java/

│ │ │ └── com/

│ │ │ └── com.ltineraryplanning.authservice/

│ │ │ ├── config/

│ │ │ ├── controller/

│ │ │ ├── service/

│ │ │ ├── repo/

│ │ │ ├── model/

│ │ │ ├── dto/

│ │ │ ├── kafka/

│ │ │ ├── Validation/

│ │ │ └── AuthServiceApplication.java

│ └── resources/

│ ├── application.yml

│ ├── static/

│ └── templates/

├── test/

│ └── java/

│ └── com/

│ └── com.bank.web.app.auth/

│ ├── controller/

│ ├── service/

│ └── Auth