

# Bank-Web-App

## Auth-Service

### TABLE OF CONTENTS

1. Introduction .....	2
2. Tech Stack .....	3
3. Data Dictionary .....	4
3.1.1 UserVerification .....	4
4. REST API Endpoints .....	5
4.1.1 /register .....	6
4.1.2 /login .....	8
4.1.3 /verify/{id}/{token} .....	10
5. Resilience4j Configuration.....	11
5.1.1. Circuit Breaker .....	11
5.1.2 Rate Limiter .....	11
6. Project Structure .....	12

# 1. Introduction

The Auth Service is a core microservice in a banking ecosystem responsible for user registration, login and 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

## 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

#### 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	Unique identifier of the user	"user_123"
email	String	User's email address	" <a href="#">user@example.com</a> "
token	String	Verification token sent to the user	"abc123xyzTOKEN"
isVerified	boolean	Indicates whether the user has been verified	true
sendAt	LocalDateTime	Timestamp when the verification token was sent	2024-07-01T10:00:00
expiryDate	LocalDateTime	Token expiration timestamp	2024-07-02T10:00:00

## 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	GET	/heartbeat	Health check endpoint to confirm service is alive.

### 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

```
{  
  "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

```
{  
  "username": "dhruv",  
  "password": "123123"  
}
```



## Response

```
{
  status : string
  message : string
  object : any
}
```

### Response Example

```
{
  "status": "200",
  "message": "Login Success",
  "object": {
    "token":
"eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6IChJQMFdGMMy1yTG41azh3NnFhYVVEQWw0
5S1pxcTQwTlA2OHdOd1lwVFZ0Q3VjIn0.eyJleHAiOiJlE3NTM3ODIxNzAsImthdCI6MTc1Mzc4MT
U3MCwianRpljoiMTQ4Y2Q0MTYtMTRhOS00hbmFnZS1heaWVhbnRjX2ttdqYALmYWRtRsBv42jEkF
lubGq29LKBnWZ5QOA8-
bcOVEyLIUgC5nzLtd1xL3SFqwQDVxnPn9vevua8k3o1ok2C_5Qh2j8DNyNAKGL7Nanm0Y560
91MTdUPQhq8UO0LFZ7vhRQYI6dMVJNhP7mA"
  }
}
```

### 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	<b>Id:</b> user-id <b>token:</b> unique token <b>Example:</b> <b>id:</b> 5cc6f578-8646-4957-81eb-69043c6b4bf6 <b>token:</b> 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  
}
```

## 5. Resilience4j Configuration

### 5.1.1. Circuit Breaker

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

### 5.1.2 Rate Limiter

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

## 6. Project Structure

```
Auth/
├── src/
│   ├── main/
│   │   ├── java/
│   │   │   └── com/
│   │   │       └── com.bank.web.app.auth/
│   │   │           ├── config/
│   │   │           ├── controller/
│   │   │           ├── service/
│   │   │           ├── repo/
│   │   │           ├── model/
│   │   │           ├── dto/
│   │   │           ├── kafka/
│   │   │           ├── Validation/
│   │   │           └── AuthApplication.java
│   └── resources/
│       ├── application.yml
│       ├── static/
│       └── templates/
└── test/
    ├── java/
    │   └── com/
    │       └── com.bank.web.app.auth/
    │           ├── controller/
    │           ├── service/
    │           └── Auth
```