PERSONAL DETAILS

NAME: NANSUBUGA JOYCE EUZEBIA

Email : njeuzebia@gmail.com

Phone: 0758720304/0780365538

Repository Url: https://github.com/euzebia/accounts_manager

BRIEF DESCRIPTION OF THE DESIGN AND EVIDENCE IN SCREEN SHOTS

Software description

Software name: accounts_manager

A real-world use case considered for implementation is an off-the-shelf software developed by a third-party software vendor/system owner, that is used in banks to help bank staff (personal banker) create bank accounts, modify accounts based on customer preferences and managing other account details.

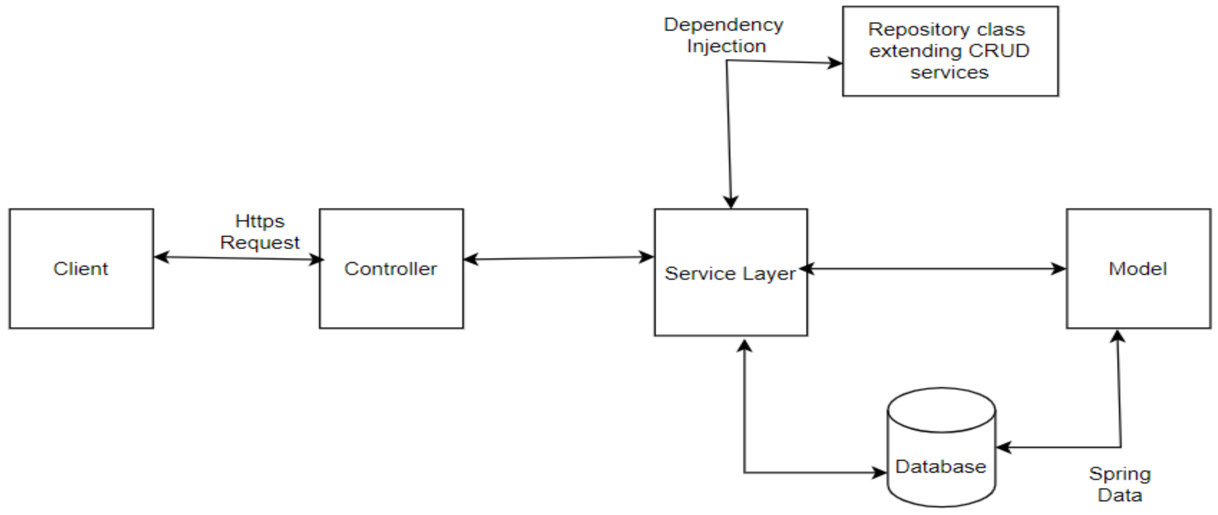
The software assumes that three user types interact with it i.e., bank staff, bank customers and system owners.

The system owners over see all activities taking place in the system, bank staff are responsible for managing customer accounts and for customers they may only interact with the system in scenarios when they want to view their personal information.

The document is broken down into various sections each detailing various software development areas where handled; some of them include but not limited to database design, system architecture, security, application performance, etc..

1. <u>Api architecture</u>

The api design follows a layered architecture as illustrated below.



Explanation;

1. A client makes an http request i.e GET,POST,etc.. and is forwarded to the controller.
2. The controller maps the request and calls the service layer logic.
3. In the service layer, data validation, database calls and other business logic take place from there
4. The database acts as a central repository for the data used by the api/application.

2. <u>Database Design</u>

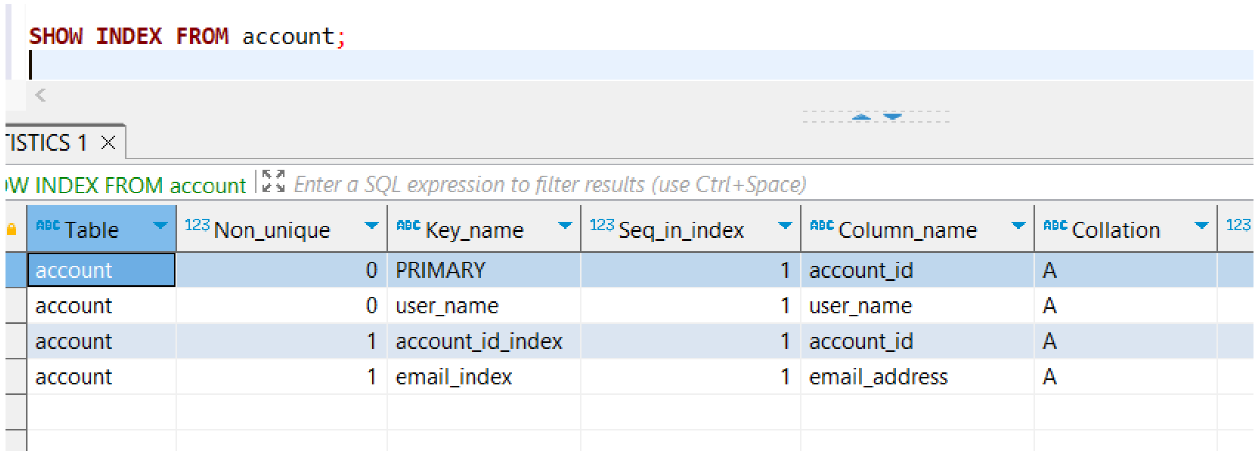
a. Data normalization has been used to eliminate data redundancy as some properties are shared by more than one table.

b. A primary key has been added to each table to uniquely identify each record in the table.

c. Stored procedures and views have been to retrieve data from related tables in order to reduce on the traffic between the api and the db. Instead of sending multiple lengthy SQL statements, the api only sends sends the name and parameters of the stored procedure and details are retrieved once found

d. Use of indices have been on columns like username and email address column in the accounts tables as it is one of the tables on which many search queries are done to retrieve accounts, as data grows, search queries tend to slow down.

Illustration



```
SHOW INDEX FROM account;
```

ISTICS 1

W INDEX FROM account | Enter a SQL expression to filter results (use Ctrl+Space)

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	123
account	0	PRIMARY	1	account_id	A	
account	0	user_name	1	user_name	A	
account	1	account_id_index	1	account_id	A	
account	1	email_index	1	email_address	A	

e. User_name and email are unique as they part of details used to uniquely identify a user in the system.

f. Foreign keys have been used to link tables that are related to each other.

g. Different types of users ie Type A,Type B,Type C have been categoried as bank staff,Customer and Vendor respectively.

h. The institution in which these users belong have been categorized by type ie the institution in which bank staff and Customer belong has been categorized as bank, type C as Third party software vendor (vendor of the software).

i. Type C user(vendor) creates a bank staff and assigns them a under the institution bank.

j. A bank staff creates customer accounts and modifies customers' info based on customer preferences.

k. Customers have been categorized according to customer category ie corporate and ordinary user.

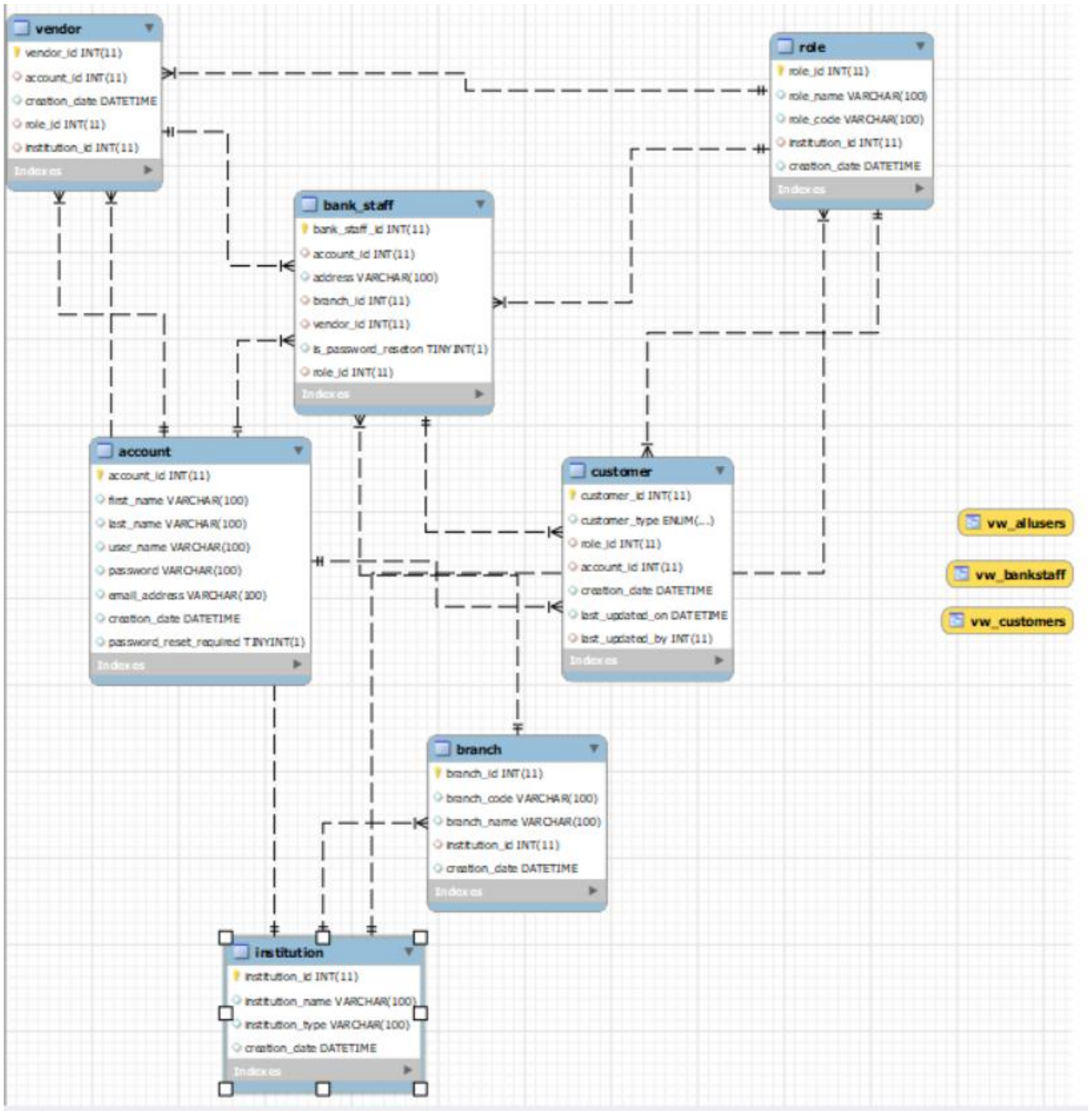
l. Institution bank has a branch attached to it.

m. A vendor creates a bank staff who is responsible for creating customer accounts.

n. Database views have been used to group data related to the various user types

The entity relationship diagram showing how different entities talk to each other can be seen below;

ENTITY RELATIONSHIP DIAGRAM



3. Security

- Api security

The api has basic authentication there by unauthorized are prevented from accessing the endpoint;

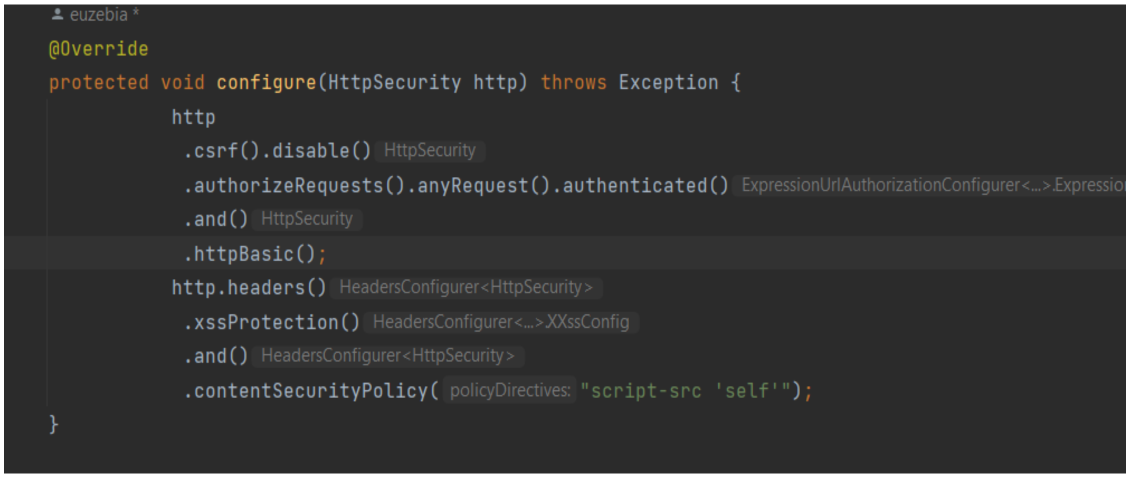
Authentication type : Basic

Username: api_user

Password: t5p4krGSJyil2hWB

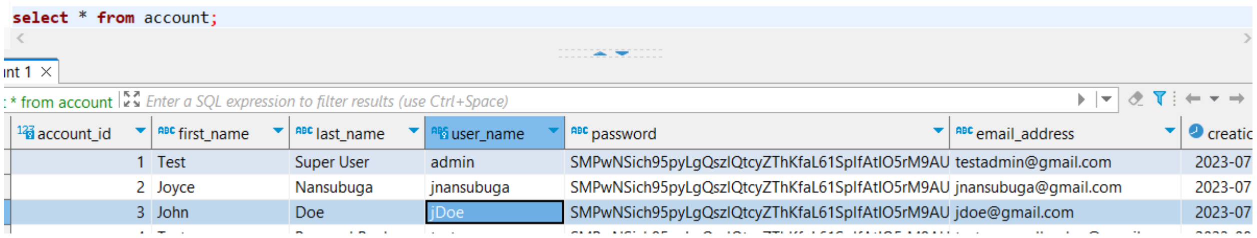
- Validating post requests to the endpoint

  Measures have been put to handle xss and csrf vulnerabilities as seen below;



```
euzebia *
@Override
protected void configure(HttpSecurity http) throws Exception {
        http
          .csrf().disable() HttpSecurity
          .authorizeRequests().anyRequest().authenticated() ExpressionUrlAuthorizationConfigurer<...>.Expressio
          .and() HttpSecurity
          .httpBasic();
        http.headers() HeadersConfigurer<HttpSecurity>
          .xssProtection() HeadersConfigurer<...>.XXssConfig
          .and() HeadersConfigurer<HttpSecurity>
          .contentSecurityPolicy( policyDirectives: "script-src 'self'");
}
```

- Encyption of user passwords while data is at rest,this prevents malicious users from identifying user passwords;



```
select * from account;
```

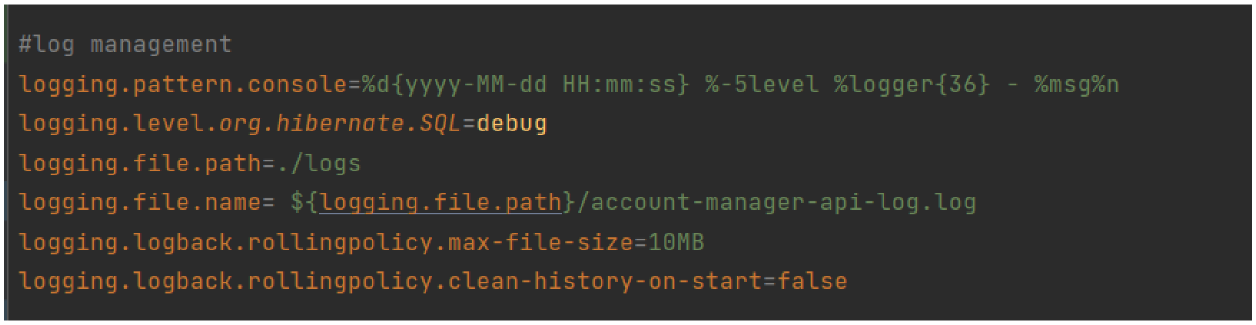
nt 1

* from account | Enter a SQL expression to filter results (use Ctrl+Space)

	account_id	first_name	last_name	user_name	password	email_address	creatio
	1	Test	Super User	admin	SMPwNSich95pyLgQszlQtcyZThKfaL61SplfAtlO5rM9AU	testadmin@gmail.com	2023-07
	2	Joyce	Nansubuga	jnansubuga	SMPwNSich95pyLgQszlQtcyZThKfaL61SplfAtlO5rM9AU	jnansubuga@gmail.com	2023-07
	3	John	Doe	jDoe	SMPwNSich95pyLgQszlQtcyZThKfaL61SplfAtlO5rM9AU	jdoe@gmail.com	2023-07

4. Log Management

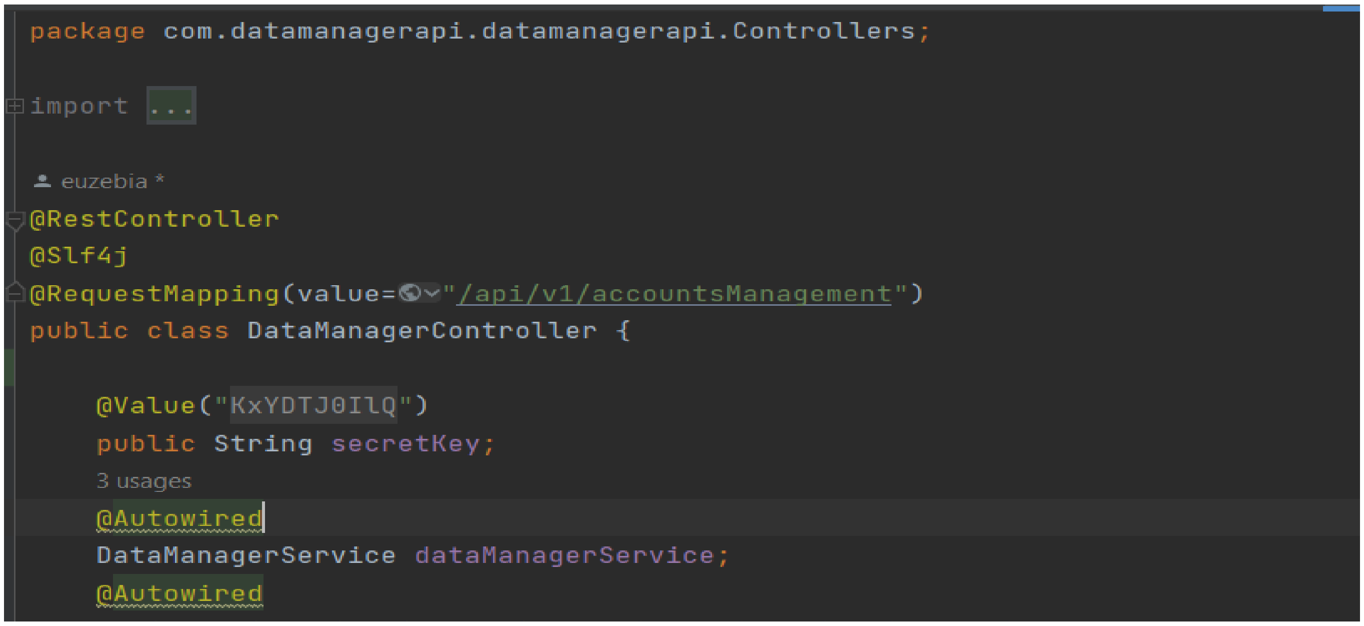
   This helps in easy monitoring of the application performance i.e if the application is catching any errors during execution, errors can easily be identified as they are logged on file during execution. As seen below, the file log reached 10MB,it is zipped,rolled and a new one created basing on configurations below;



```
#log management
logging.pattern.console=%d{yyyy-MM-dd HH:mm:ss} %-5level %logger{36} - %msg%n
logging.level.org.hibernate.SQL=debug
logging.file.path=./logs
logging.file.name= ${logging.file.path}/account-manager-api-log.log
logging.logback.rollingpolicy.max-file-size=10MB
logging.logback.rollingpolicy.clean-history-on-start=false
```

5. Api versioning

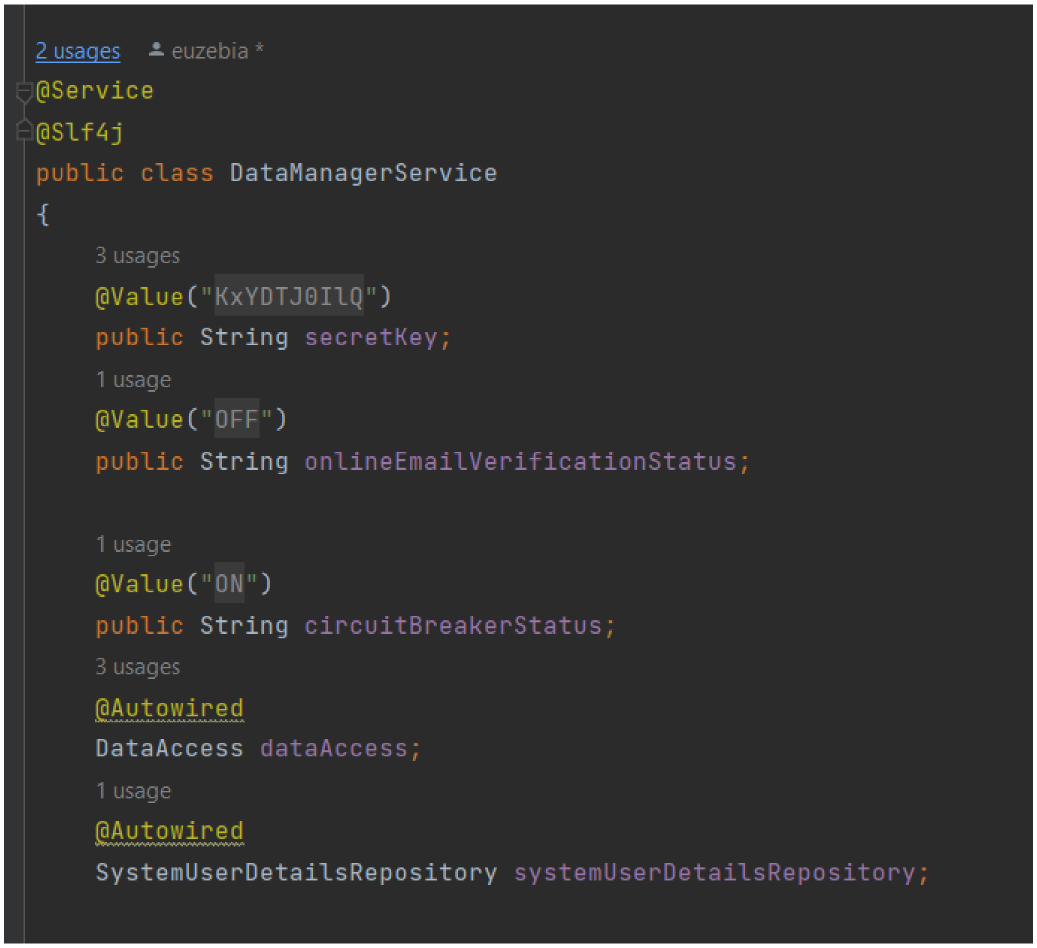
   It helps to ensure stability and reliability, if you don't properly version apis this can cause a big negative effect on products already in production. With this, developers can easily maintain and make additions to existing software without breaking changes.



```
package com.datamanagerapi.datamanagerapi.Controllers;

import ...

euzebia *
@RestController
@Slf4j
@RequestMapping(value="/api/v1/accountsManagement")
public class DataManagerController {

    @Value("KxYDTJ0IlQ")
    public String secretKey;
    3 usages
    @Autowired
    DataManagerService dataManagerService;
    @Autowired
```

6. Dependency injection

   It helps to ensure loose coupling between classes. @Autowired annotation injects the DataAccess class into the ManagerDataService as shown below;

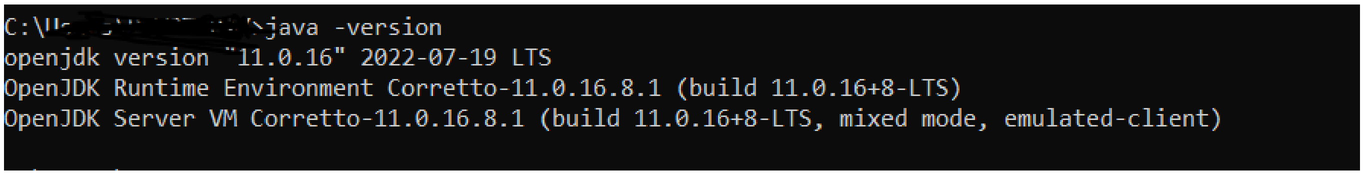


```
2 usages   euzebia *
@Service
@Slf4j
public class DataManagerService
{
        3 usages
        @Value("KxYDTJ0IlQ")
        public String secretKey;
        1 usage
        @Value("OFF")
        public String onlineEmailVerificationStatus;

        1 usage
        @Value("ON")
        public String circuitBreakerStatus;
        3 usages
        @Autowired
        DataAccess dataAccess;
        1 usage
        @Autowired
        SystemUserDetailsRepository systemUserDetailsRepository;
```

Running the developed application;

Pre-requisites

- Mysql and php installed (mysql version 5.1.3,PHP Version: 7.4.28)
- Java installed,to confirm if all is set run the command below;

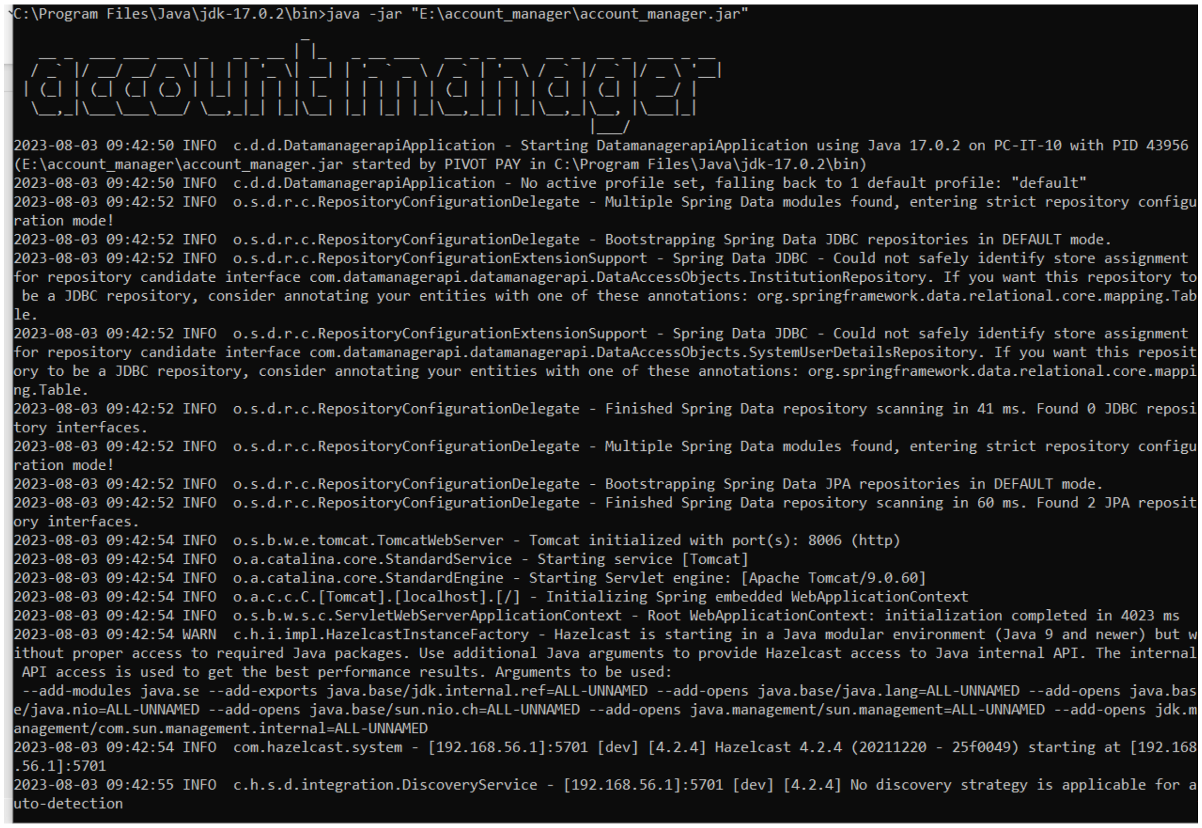


```
C:\U[redacted]>java -version
openjdk version "11.0.16" 2022-07-19 LTS
OpenJDK Runtime Environment Corretto-11.0.16.8.1 (build 11.0.16+8-LTS)
OpenJDK Server VM Corretto-11.0.16.8.1 (build 11.0.16+8-LTS, mixed mode, emulated-client)
```

- After you have successfully closed the application, navigate to the \accounts_manager\build\libs to locate the .jar file to run;

- ❖ On your machine(windows) for our example,navigate to where the jdk was installed and enter the commands below to start the application;

java -jar “the file path to .jar on your machine”



```
C:\Program Files\Java\jdk-17.0.2\bin>java -jar "E:\account_manager\account_manager.jar"

2023-08-03 09:42:50 INFO  c.d.d.DatamanagerapiApplication - Starting DatamanagerapiApplication using Java 17.0.2 on PC-IT-10 with PID 43956
(E:\account_manager\account_manager.jar started by PIVOT PAY in C:\Program Files\Java\jdk-17.0.2\bin)
2023-08-03 09:42:50 INFO  c.d.d.DatamanagerapiApplication - No active profile set, falling back to 1 default profile: "default"
2023-08-03 09:42:52 INFO  o.s.d.r.c.RepositoryConfigurationDelegate - Multiple Spring Data modules found, entering strict repository configu
ration mode!
2023-08-03 09:42:52 INFO  o.s.d.r.c.RepositoryConfigurationDelegate - Bootstrapping Spring Data JDBC repositories in DEFAULT mode.
2023-08-03 09:42:52 INFO  o.s.d.r.c.RepositoryConfigurationExtensionSupport - Spring Data JDBC - Could not safely identify store assignment
for repository candidate interface com.datamanagerapi.datamanagerapi.DataAccessObjects.InstitutionRepository. If you want this repository to
 be a JDBC repository, consider annotating your entities with one of these annotations: org.springframework.data.relational.core.mapping.Tab
le.
2023-08-03 09:42:52 INFO  o.s.d.r.c.RepositoryConfigurationExtensionSupport - Spring Data JDBC - Could not safely identify store assignment
for repository candidate interface com.datamanagerapi.datamanagerapi.DataAccessObjects.SystemUserDetailsRepository. If you want this reposit
ory to be a JDBC repository, consider annotating your entities with one of these annotations: org.springframework.data.relational.core.mappi
ng.Table.
2023-08-03 09:42:52 INFO  o.s.d.r.c.RepositoryConfigurationDelegate - Finished Spring Data repository scanning in 41 ms. Found 0 JDBC reposi
tory interfaces.
2023-08-03 09:42:52 INFO  o.s.d.r.c.RepositoryConfigurationDelegate - Multiple Spring Data modules found, entering strict repository configu
ration mode!
2023-08-03 09:42:52 INFO  o.s.d.r.c.RepositoryConfigurationDelegate - Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2023-08-03 09:42:52 INFO  o.s.d.r.c.RepositoryConfigurationDelegate - Finished Spring Data repository scanning in 60 ms. Found 2 JPA reposit
ory interfaces.
2023-08-03 09:42:54 INFO  o.s.b.w.e.tomcat.TomcatWebServer - Tomcat initialized with port(s): 8006 (http)
2023-08-03 09:42:54 INFO  o.a.catalina.core.StandardService - Starting service [Tomcat]
2023-08-03 09:42:54 INFO  o.a.catalina.core.StandardEngine - Starting Servlet engine: [Apache Tomcat/9.0.60]
2023-08-03 09:42:54 INFO  o.a.c.c.C.[Tomcat].[localhost].[/] - Initializing Spring embedded WebApplicationContext
2023-08-03 09:42:54 INFO  o.s.b.w.s.c.ServletWebServerApplicationContext - Root WebApplicationContext: initialization completed in 4023 ms
2023-08-03 09:42:54 WARN  c.h.i.impl.HazelcastInstanceFactory - Hazelcast is starting in a Java modular environment (Java 9 and newer) but w
ithout proper access to required Java packages. Use additional Java arguments to provide Hazelcast access to Java internal API. The internal
 API access is used to get the best performance results. Arguments to be used:
 --add-modules java.se --add-exports java.base/jdk.internal.ref=ALL-UNNAMED --add-opens java.base/java.lang=ALL-UNNAMED --add-opens java.bas
e/java.nio=ALL-UNNAMED --add-opens java.base/sun.nio.ch=ALL-UNNAMED --add-opens java.management/sun.management=ALL-UNNAMED --add-opens jdk.m
anagement/com.sun.management.internal=ALL-UNNAMED
2023-08-03 09:42:54 INFO  com.hazelcast.system - [192.168.56.1]:5701 [dev] [4.2.4] Hazelcast 4.2.4 (20211220 - 25f0049) starting at [192.168
.56.1]:5701
2023-08-03 09:42:55 INFO  c.h.s.d.integration.DiscoveryService - [192.168.56.1]:5701 [dev] [4.2.4] No discovery strategy is applicable for a
uto-detection
```

Here the application has started running listening for traffic on port 8006 as seen above,we can start the tests.

<u>Choosing a client to test with</u>

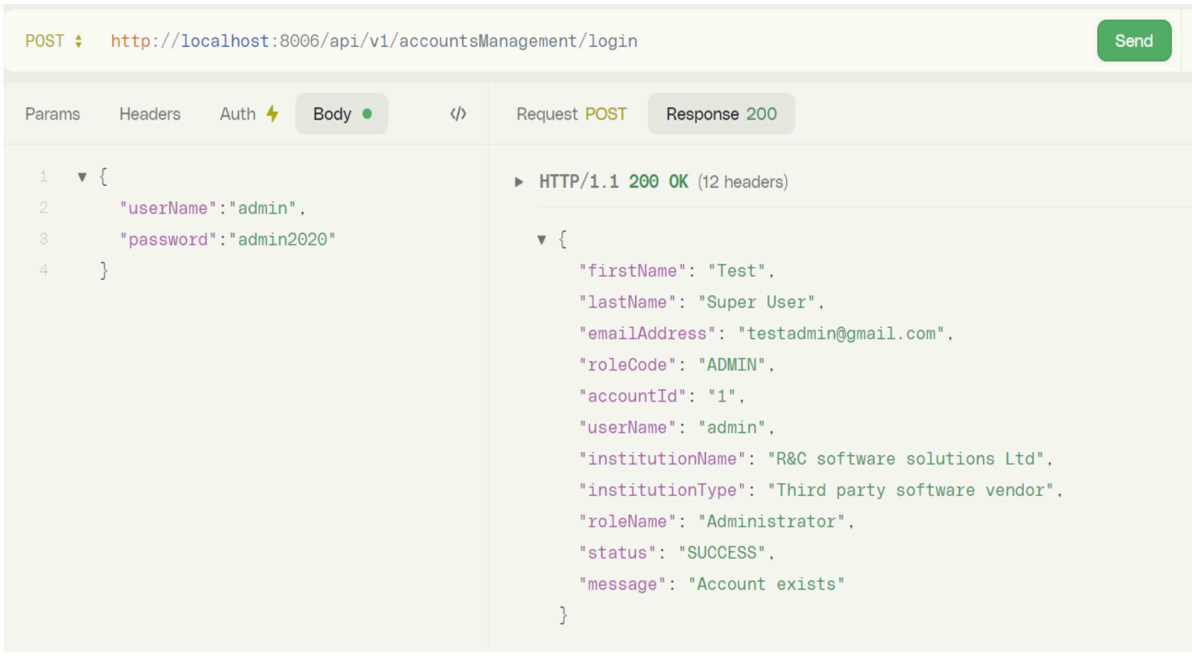
- ➢ For this case HTTPie is acting as our client. It can downloaded from https://httpie.io/download.

Test scenarios

- ➢ Login: All users access the application from one central point

  User Type;

  1. Vendor (Username: admin, Password: admin2020)

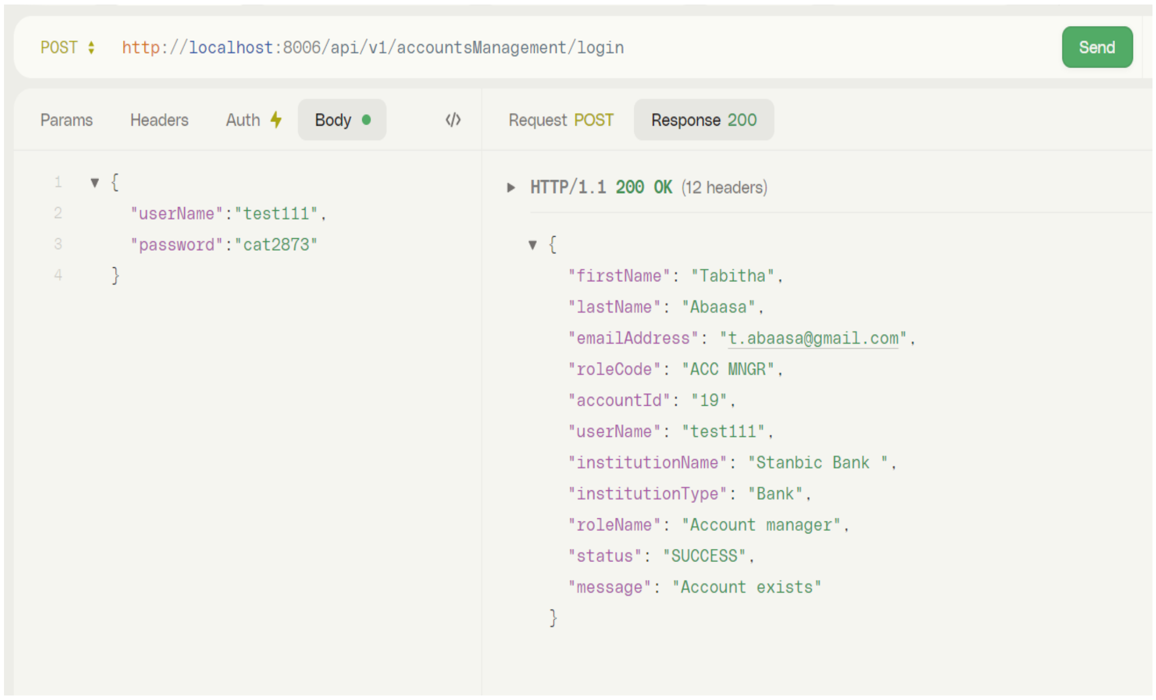

POST http://localhost:8006/api/v1/accountsManagement/login Send

Params Headers Auth Body Request POST Response 200

```
{
  "userName":"admin",
  "password":"admin2020"
}
```

```
HTTP/1.1 200 OK (12 headers)
{
  "firstName": "Test",
  "lastName": "Super User",
  "emailAddress": "testadmin@gmail.com",
  "roleCode": "ADMIN",
  "accountId": "1",
  "userName": "admin",
  "institutionName": "R&C software solutions Ltd",
  "institutionType": "Third party software vendor",
  "roleName": "Administrator",
  "status": "SUCCESS",
  "message": "Account exists"
}
```

2. Bank staff (username:test111,password: cat2873)

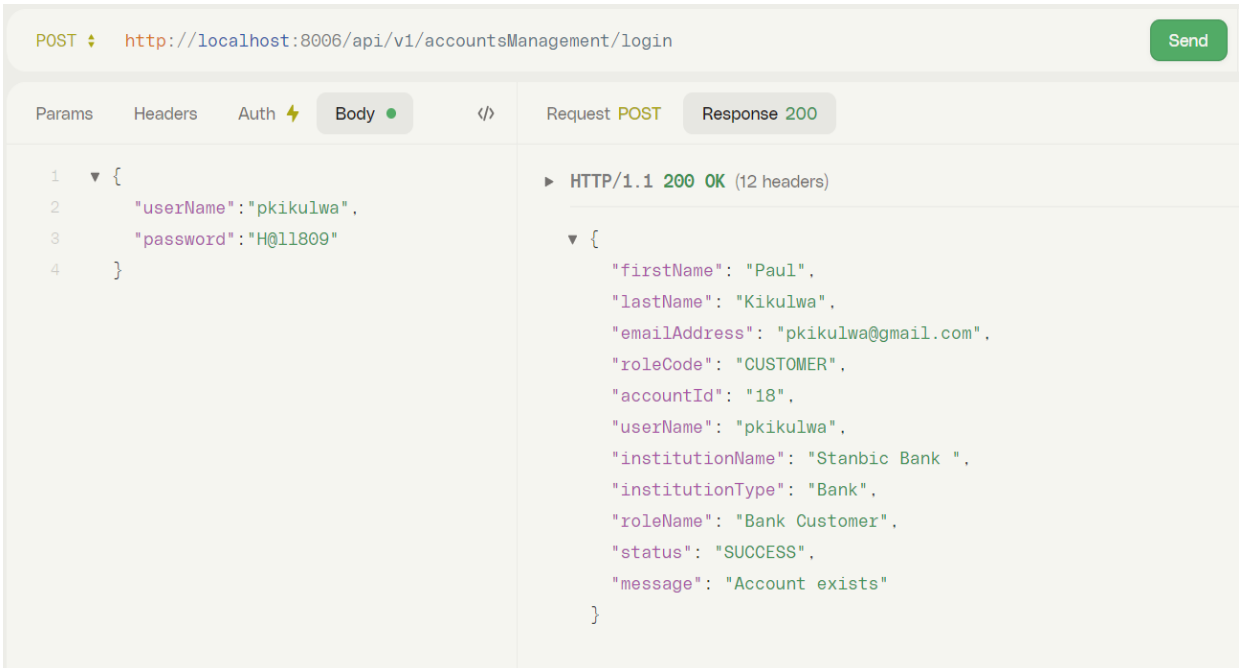

POST http://localhost:8006/api/v1/accountsManagement/login Send

Params Headers Auth Body Request POST Response 200

```
{
  "userName":"test111",
  "password":"cat2873"
}
```

```
HTTP/1.1 200 OK (12 headers)
{
  "firstName": "Tabitha",
  "lastName": "Abaasa",
  "emailAddress": "t.abaasa@gmail.com",
  "roleCode": "ACC MNGR",
  "accountId": "19",
  "userName": "test111",
  "institutionName": "Stanbic Bank ",
  "institutionType": "Bank",
  "roleName": "Account manager",
  "status": "SUCCESS",
  "message": "Account exists"
}
```

3. Customer



POST http://localhost:8006/api/v1/accountsManagement/login Send

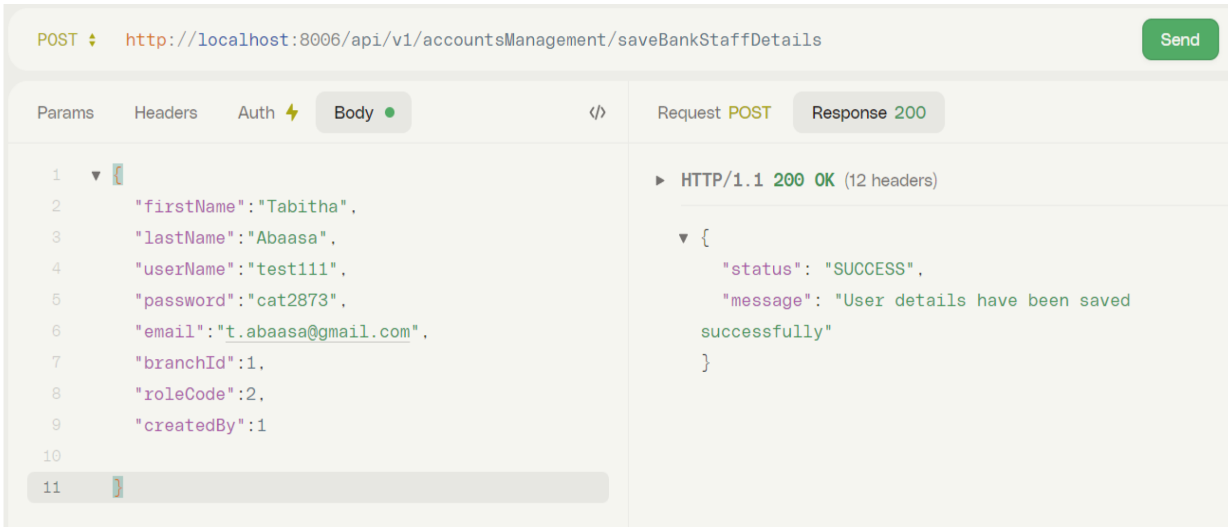
Params Headers Auth Body | Request POST Response 200

```
{
    "userName":"pkikulwa",
    "password":"H@1l809"
}
```

```
HTTP/1.1 200 OK (12 headers)

{
    "firstName": "Paul",
    "lastName": "Kikulwa",
    "emailAddress": "pkikulwa@gmail.com",
    "roleCode": "CUSTOMER",
    "accountId": "18",
    "userName": "pkikulwa",
    "institutionName": "Stanbic Bank ",
    "institutionType": "Bank",
    "roleName": "Bank Customer",
    "status": "SUCCESS",
    "message": "Account exists"
}
```

All the three user types have been able to logon from one central point.

➢ Save Bank Staff Details



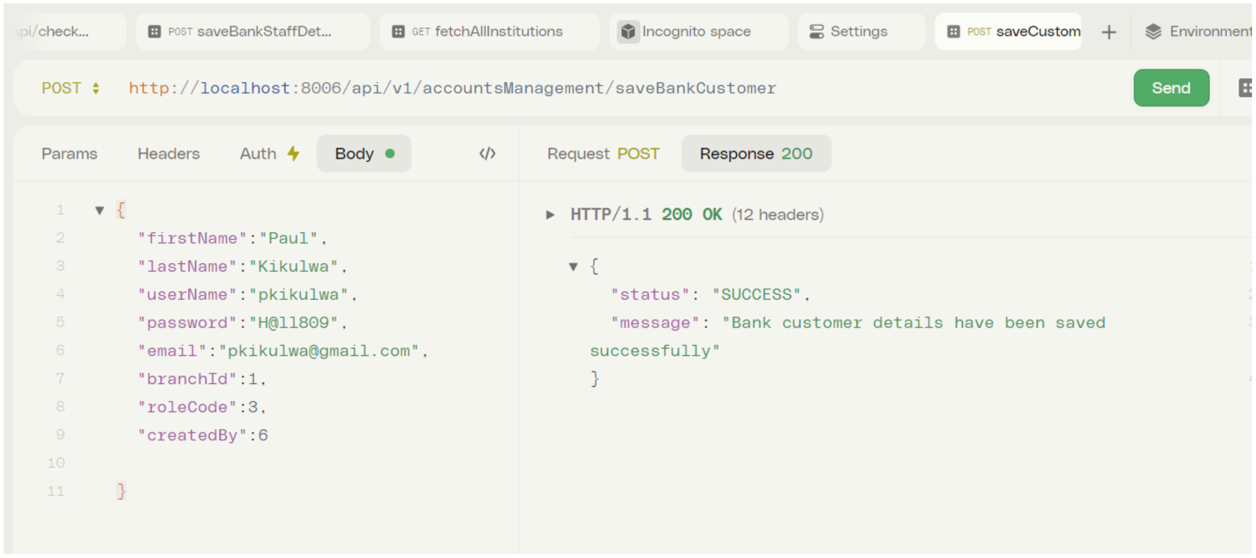
POST http://localhost:8006/api/v1/accountsManagement/saveBankStaffDetails Send

Params Headers Auth Body | Request POST Response 200

```
{
    "firstName":"Tabitha",
    "lastName":"Abaasa",
    "userName":"test111",
    "password":"cat2873",
    "email":"t.abaasa@gmail.com",
    "branchId":1,
    "roleCode":2,
    "createdBy":1

}
```

```
HTTP/1.1 200 OK (12 headers)

{
    "status": "SUCCESS",
    "message": "User details have been saved
successfully"
}
```

➢ Save Customer Details



```
POST http://localhost:8006/api/v1/accountsManagement/saveBankCustomer
```

```
{
    "firstName":"Paul",
    "lastName":"Kikulwa",
    "userName":"pkikulwa",
    "password":"H@11809",
    "email":"pkikulwa@gmail.com",
    "branchId":1,
    "roleCode":3,
    "createdBy":6

}
```

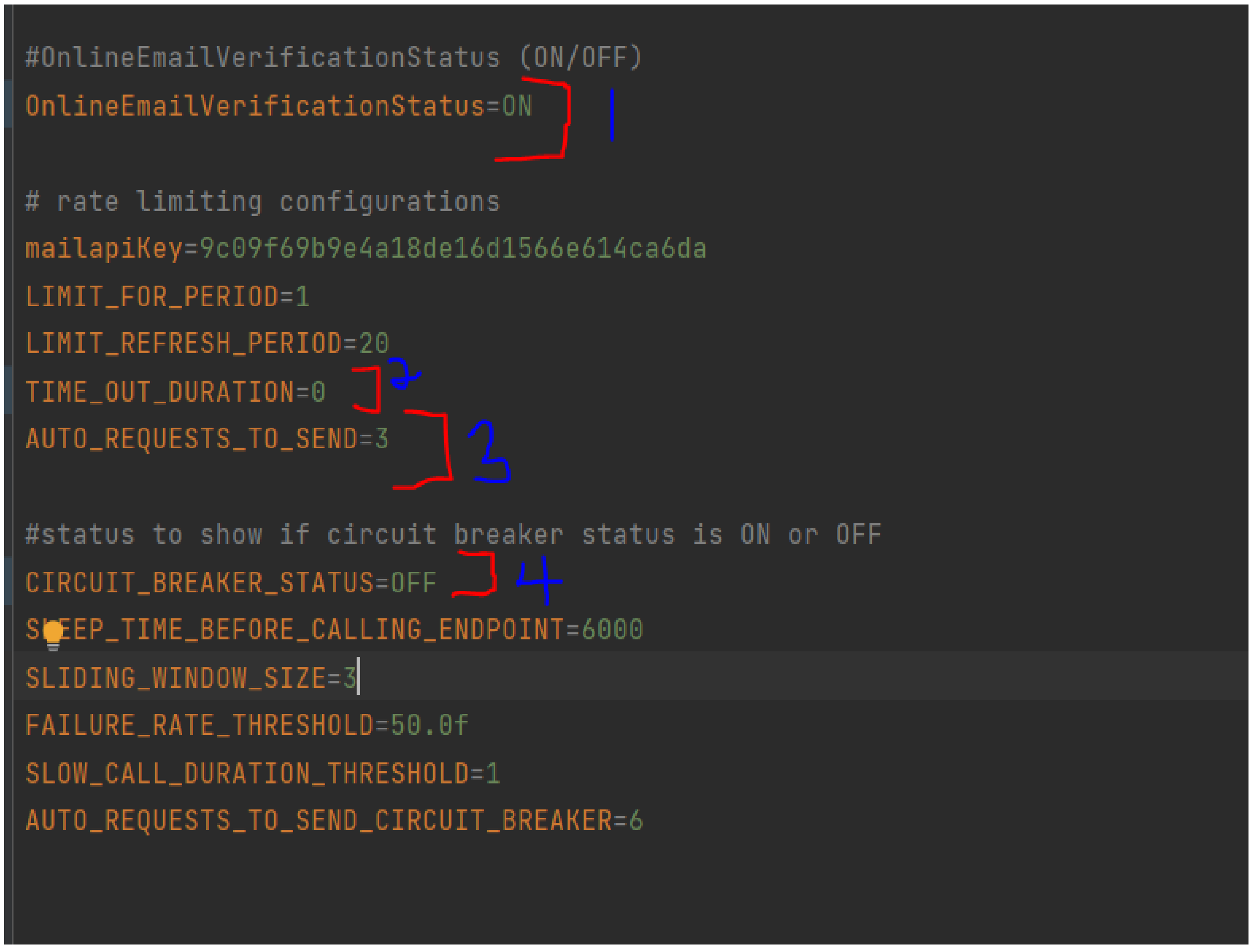
```
HTTP/1.1 200 OK (12 headers)
{
  "status": "SUCCESS",
  "message": "Bank customer details have been saved
successfully"
}
```

➢ Save Bank Staff Details with Rate Limiting Enabled

To illustrate how rate limiting works, a configuration was done in the service on validating user input we check to confirm if the supplied email is valid, for this case the service can either choose to validate the email locally using the regex expression or online using a third party endpoint that provides online email verification. This implementation focuses on client side rate limiting as the emails are issued basing on a given service plan. This example illustrates how to limit the number of requests sent to the third party service provider or saving the cost that would be incurred if the service is elasting. Here when a give request limit is exceeded,the requests are throttled thus preventing them from being sent out.

Configs to enable rate limiting to take place.

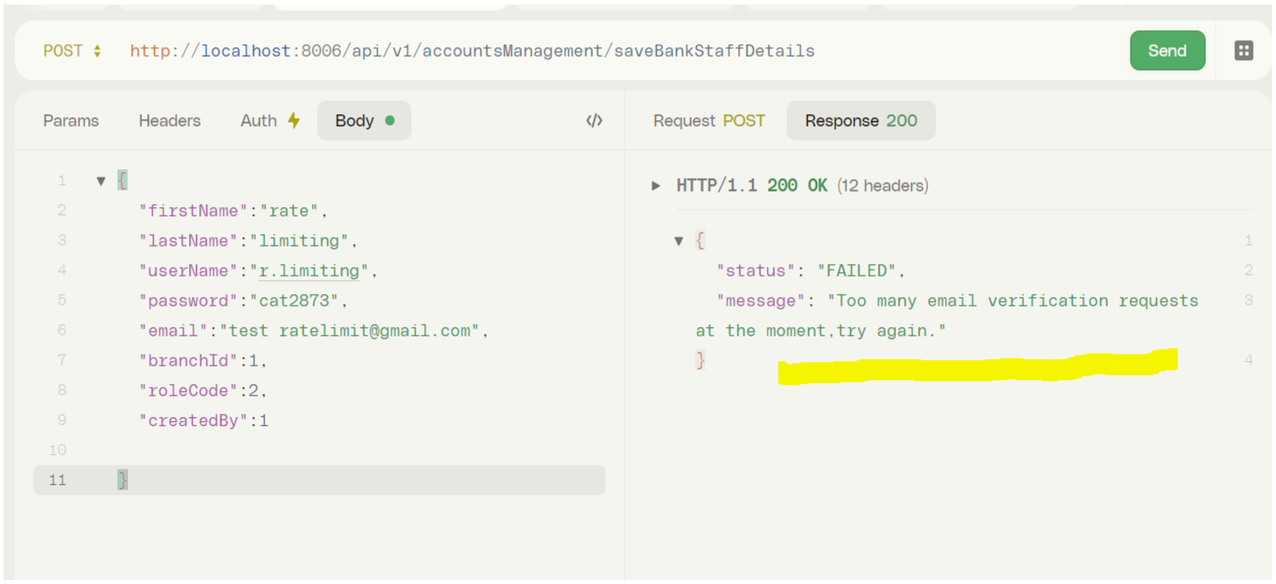
- **Set the email verification service to be online i.e validate at the third party**



```
#OnlineEmailVerificationStatus (ON/OFF)
OnlineEmailVerificationStatus=ON

# rate limiting configurations
mailapiKey=9c09f69b9e4a18de16d1566e614ca6da
LIMIT_FOR_PERIOD=1
LIMIT_REFRESH_PERIOD=20
TIME_OUT_DURATION=0
AUTO_REQUESTS_TO_SEND=3

#status to show if circuit breaker status is ON or OFF
CIRCUIT_BREAKER_STATUS=OFF
SLEEP_TIME_BEFORE_CALLING_ENDPOINT=6000
SLIDING_WINDOW_SIZE=3
FAILURE_RATE_THRESHOLD=50.0f
SLOW_CALL_DURATION_THRESHOLD=1
AUTO_REQUESTS_TO_SEND_CIRCUIT_BREAKER=6
```

Settings labelled 1,2,3 and 4 should be as shown above, so when a user accesses the client(httpie) and attempts to register a new bank staff, the service checks if email verification is being done online or offline. In this case it is online,we have limited to number of requests acceptable period to 1,2 says if a request comes and a thread is not able to acquire permission don’t wait just stop execution and return error back to client,internally the service automatically generates 3 more requests to send along with this initiated from the client. Here the all together 4 requests will be sent and since configuration 2 is set to 0, an exception of request not permitted will be thrown and a failure message will be sent to the client. If configuration 2 is not 0, the thread will wait and until it gets permission to process the request;

Error returned on client side;

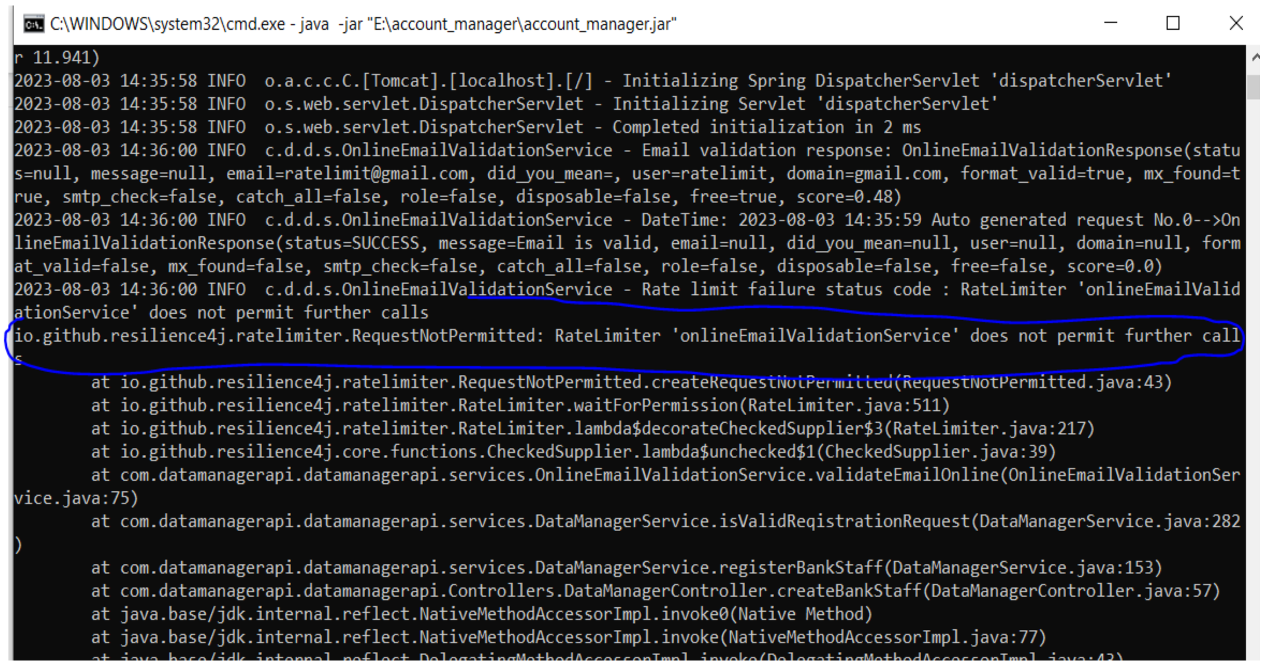


```
POST http://localhost:8006/api/v1/accountsManagement/saveBankStaffDetails
```

```
{
    "firstName":"rate",
    "lastName":"limiting",
    "userName":"r.limiting",
    "password":"cat2873",
    "email":"test ratelimit@gmail.com",
    "branchId":1,
    "roleCode":2,
    "createdBy":1

}
```

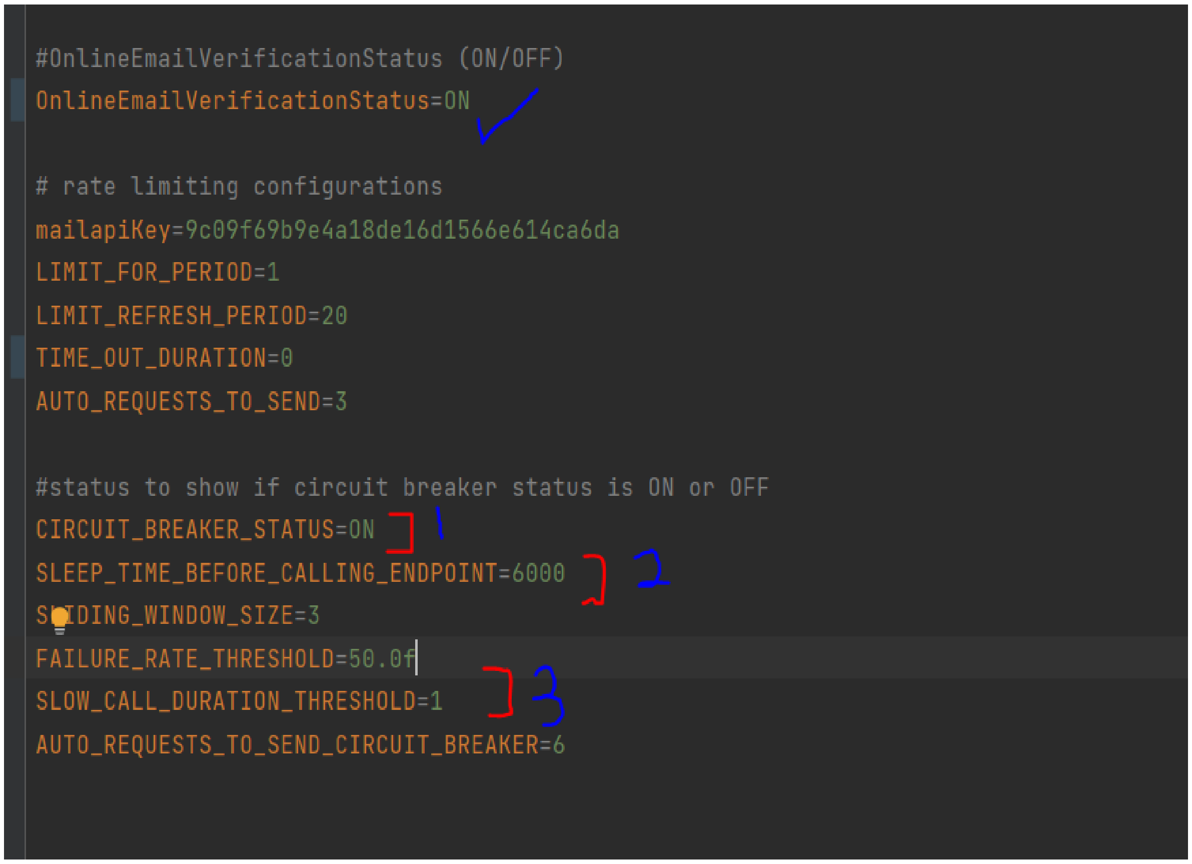
```
HTTP/1.1 200 OK (12 headers)

{
  "status": "FAILED",
  "message": "Too many email verification requests
at the moment.try again."
}
```

Exception caught in the application



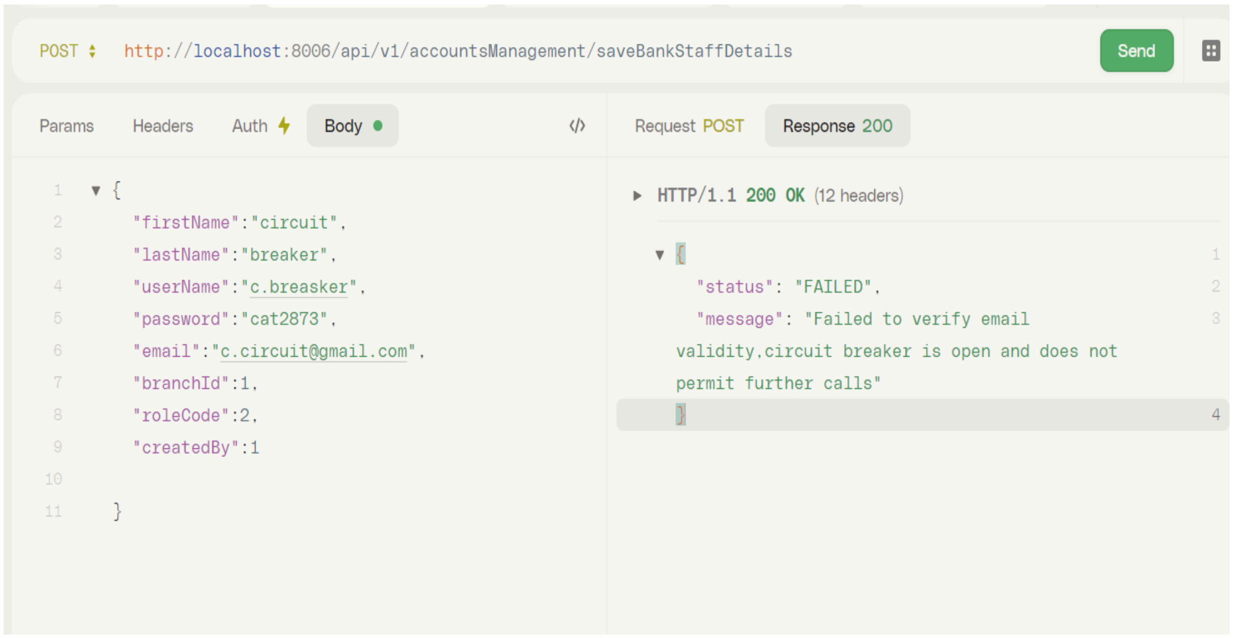
```
C:\WINDOWS\system32\cmd.exe - java  -jar "E:\account_manager\account_manager.jar"
r 11.941)
2023-08-03 14:35:58 INFO  o.a.c.c.C.[Tomcat].[localhost].[/] - Initializing Spring DispatcherServlet 'dispatcherServlet'
2023-08-03 14:35:58 INFO  o.s.web.servlet.DispatcherServlet - Initializing Servlet 'dispatcherServlet'
2023-08-03 14:35:58 INFO  o.s.web.servlet.DispatcherServlet - Completed initialization in 2 ms
2023-08-03 14:36:00 INFO  c.d.d.s.OnlineEmailValidationService - Email validation response: OnlineEmailValidationResponse(statu
s=null, message=null, email=ratelimit@gmail.com, did_you_mean=, user=ratelimit, domain=gmail.com, format_valid=true, mx_found=t
rue, smtp_check=false, catch_all=false, role=false, disposable=false, free=true, score=0.48)
2023-08-03 14:36:00 INFO  c.d.d.s.OnlineEmailValidationService - DateTime: 2023-08-03 14:35:59 Auto generated request No.0-->On
lineEmailValidationResponse(status=SUCCESS, message=Email is valid, email=null, did_you_mean=null, user=null, domain=null, form
at_valid=false, mx_found=false, smtp_check=false, catch_all=false, role=false, disposable=false, free=false, score=0.0)
2023-08-03 14:36:00 INFO  c.d.d.s.OnlineEmailValidationService - Rate limit failure status code : RateLimiter 'onlineEmailValid
ationService' does not permit further calls
io.github.resilience4j.ratelimiter.RequestNotPermitted: RateLimiter 'onlineEmailValidationService' does not permit further call
s
        at io.github.resilience4j.ratelimiter.RequestNotPermitted.createRequestNotPermitted(RequestNotPermitted.java:43)
        at io.github.resilience4j.ratelimiter.RateLimiter.waitForPermission(RateLimiter.java:511)
        at io.github.resilience4j.ratelimiter.RateLimiter.lambda$decorateCheckedSupplier$3(RateLimiter.java:217)
        at io.github.resilience4j.core.functions.CheckedSupplier.lambda$unchecked$1(CheckedSupplier.java:39)
        at com.datamanagerapi.datamanagerapi.services.OnlineEmailValidationService.validateEmailOnline(OnlineEmailValidationSer
vice.java:75)
        at com.datamanagerapi.datamanagerapi.services.DataManagerService.isValidReqistrationRequest(DataManagerService.java:282
)
        at com.datamanagerapi.datamanagerapi.services.DataManagerService.registerBankStaff(DataManagerService.java:153)
        at com.datamanagerapi.datamanagerapi.Controllers.DataManagerController.createBankStaff(DataManagerController.java:57)
        at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:77)
```

➢ Save Bank Staff Details with Circuit Breaker Enabled



```
#OnlineEmailVerificationStatus (ON/OFF)
OnlineEmailVerificationStatus=ON ✓

# rate limiting configurations
mailapiKey=9c09f69b9e4a18de16d1566e614ca6da
LIMIT_FOR_PERIOD=1
LIMIT_REFRESH_PERIOD=20
TIME_OUT_DURATION=0
AUTO_REQUESTS_TO_SEND=3

#status to show if circuit breaker status is ON or OFF
CIRCUIT_BREAKER_STATUS=ON ] 1
SLEEP_TIME_BEFORE_CALLING_ENDPOINT=6000 ] 2
SLIDING_WINDOW_SIZE=3
FAILURE_RATE_THRESHOLD=50.0f
SLOW_CALL_DURATION_THRESHOLD=1 ] 3
AUTO_REQUESTS_TO_SEND_CIRCUIT_BREAKER=6
```

If the email verification status is set to on, the service checks if circuit breaker status is set to on, if so then, a delay of 6 seconds is introduced (for each request a thread first sleeps for 6 seconds before calling the third party api)as seen labelled (2) in the image,
Circuit breaker status is set to On as seen in (1),the time at which the service considers to be getting a slow response is set to 1 as seen in (3) but as explained earlier each requests first waits for 6 seconds before being sent out, The service also supplements on the request with 6 more request (AUTO_REQUESTS_TO_SEND_CIRCUIT_BREAKER=6) here all requests are bound to fail and if the circuit breaker sees that the 3 requests (SLIDING_WINDOW_SIZE=3) previous calls were not successful, the failure rate threshold will be reached therefore the circuit will open and an error will be returned to the client as seen;

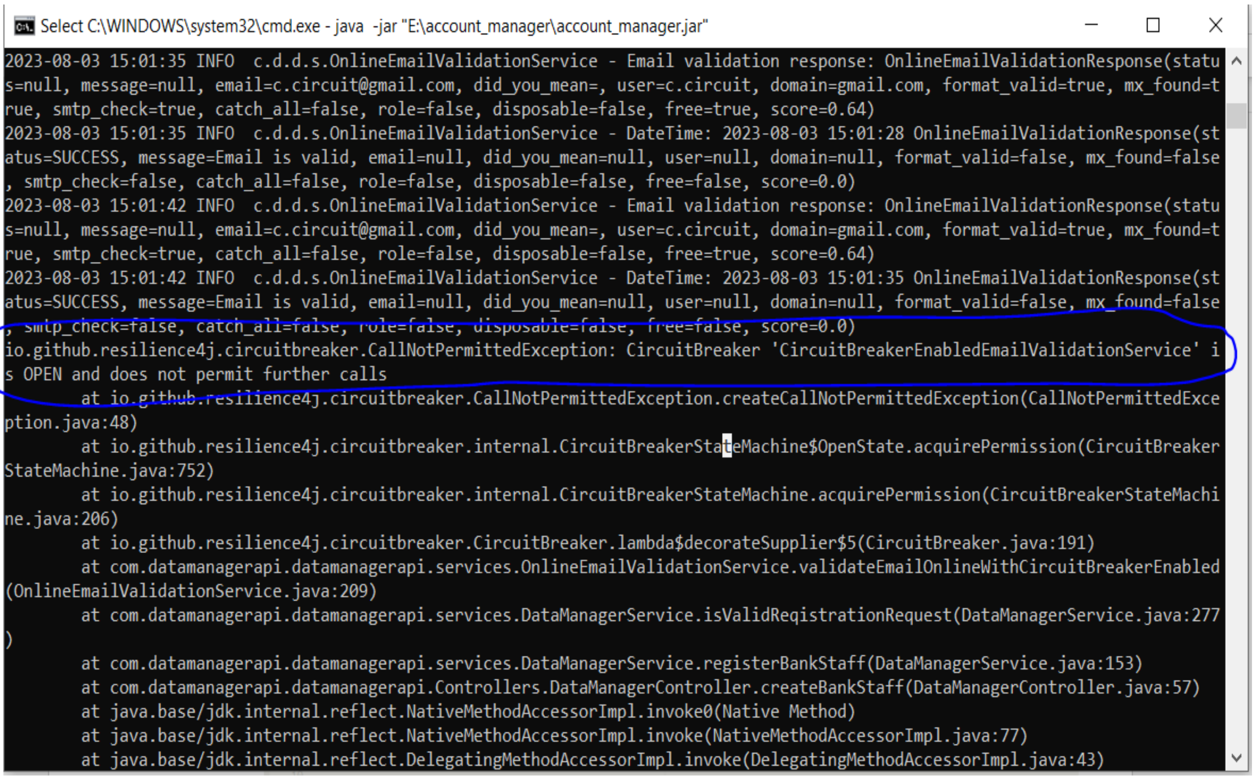


```
POST http://localhost:8006/api/v1/accountsManagement/saveBankStaffDetails
```

```
{
    "firstName":"circuit",
    "lastName":"breaker",
    "userName":"c.breasker",
    "password":"cat2873",
    "email":"c.circuit@gmail.com",
    "branchId":1,
    "roleCode":2,
    "createdBy":1

}
```

```
HTTP/1.1 200 OK (12 headers)

{
  "status": "FAILED",
  "message": "Failed to verify email
validity.circuit breaker is open and does not
permit further calls"
}
```

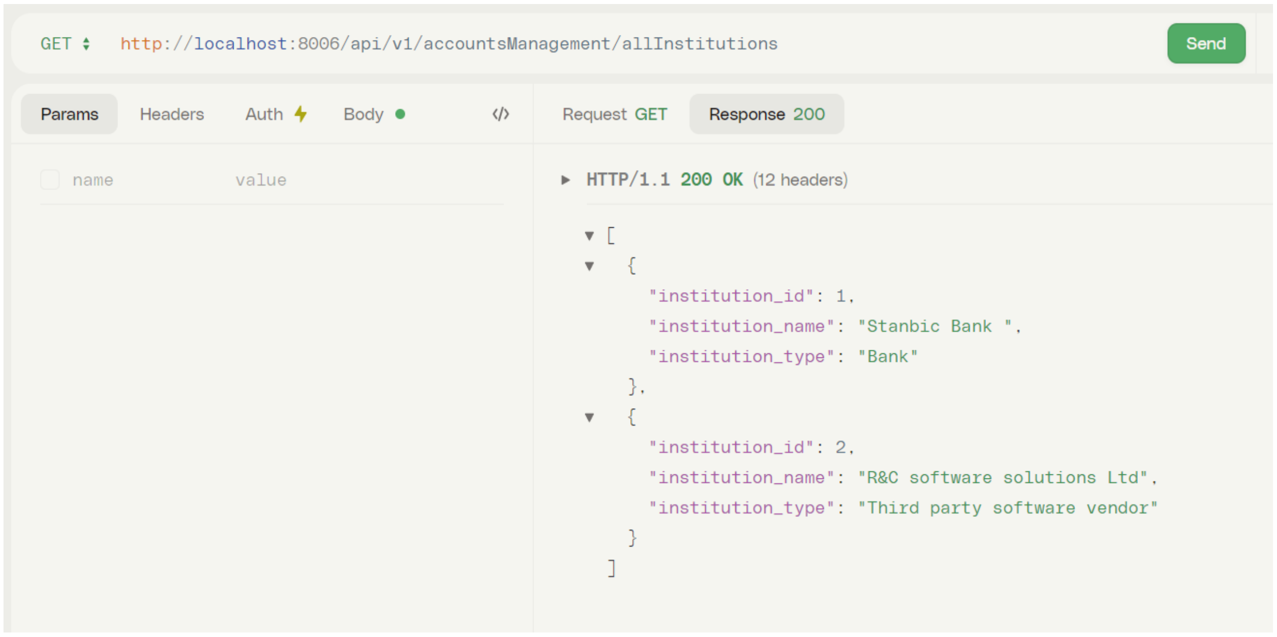
Error in application


Select C:\WINDOWS\system32\cmd.exe - java -jar "E:\account_manager\account_manager.jar"

```
2023-08-03 15:01:35 INFO  c.d.d.s.OnlineEmailValidationService - Email validation response: OnlineEmailValidationResponse(statu
s=null, message=null, email=c.circuit@gmail.com, did_you_mean=, user=c.circuit, domain=gmail.com, format_valid=true, mx_found=t
rue, smtp_check=true, catch_all=false, role=false, disposable=false, free=true, score=0.64)
2023-08-03 15:01:35 INFO  c.d.d.s.OnlineEmailValidationService - DateTime: 2023-08-03 15:01:28 OnlineEmailValidationResponse(st
atus=SUCCESS, message=Email is valid, email=null, did_you_mean=null, user=null, domain=null, format_valid=false, mx_found=false
, smtp_check=false, catch_all=false, role=false, disposable=false, free=false, score=0.0)
2023-08-03 15:01:42 INFO  c.d.d.s.OnlineEmailValidationService - Email validation response: OnlineEmailValidationResponse(statu
s=null, message=null, email=c.circuit@gmail.com, did_you_mean=, user=c.circuit, domain=gmail.com, format_valid=true, mx_found=t
rue, smtp_check=true, catch_all=false, role=false, disposable=false, free=true, score=0.64)
2023-08-03 15:01:42 INFO  c.d.d.s.OnlineEmailValidationService - DateTime: 2023-08-03 15:01:35 OnlineEmailValidationResponse(st
atus=SUCCESS, message=Email is valid, email=null, did_you_mean=null, user=null, domain=null, format_valid=false, mx_found=false
, smtp_check=false, catch_all=false, role=false, disposable=false, free=false, score=0.0)
io.github.resilience4j.circuitbreaker.CallNotPermittedException: CircuitBreaker 'CircuitBreakerEnabledEmailValidationService' i
s OPEN and does not permit further calls
        at io.github.resilience4j.circuitbreaker.CallNotPermittedException.createCallNotPermittedException(CallNotPermittedExce
ption.java:48)
        at io.github.resilience4j.circuitbreaker.internal.CircuitBreakerStateMachine$OpenState.acquirePermission(CircuitBreaker
StateMachine.java:752)
        at io.github.resilience4j.circuitbreaker.internal.CircuitBreakerStateMachine.acquirePermission(CircuitBreakerStateMachi
ne.java:206)
        at io.github.resilience4j.circuitbreaker.CircuitBreaker.lambda$decorateSupplier$5(CircuitBreaker.java:191)
        at com.datamanagerapi.datamanagerapi.services.OnlineEmailValidationService.validateEmailOnlineWithCircuitBreakerEnabled
(OnlineEmailValidationService.java:209)
        at com.datamanagerapi.datamanagerapi.services.DataManagerService.isValidReqistrationRequest(DataManagerService.java:277
)
        at com.datamanagerapi.datamanagerapi.services.DataManagerService.registerBankStaff(DataManagerService.java:153)
        at com.datamanagerapi.datamanagerapi.Controllers.DataManagerController.createBankStaff(DataManagerController.java:57)
        at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:77)
        at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
```

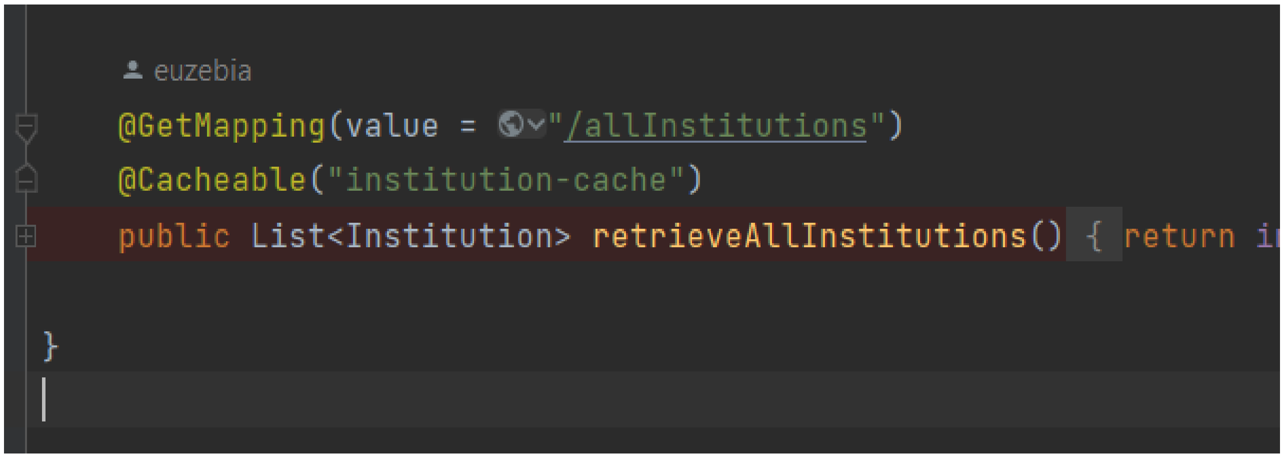
- Retrieval of cached institution details

  For Hazelcast’s support for data caching,it stores frequently data in memory such that it is easily available to services that readily need the data.



```
http://localhost:8006/api/v1/accountsManagement/allInstitutions

HTTP/1.1 200 OK (12 headers)
[
  {
    "institution_id": 1,
    "institution_name": "Stanbic Bank ",
    "institution_type": "Bank"
  },
  {
    "institution_id": 2,
    "institution_name": "R&C software solutions Ltd",
    "institution_type": "Third party software vendor"
  }
]
```

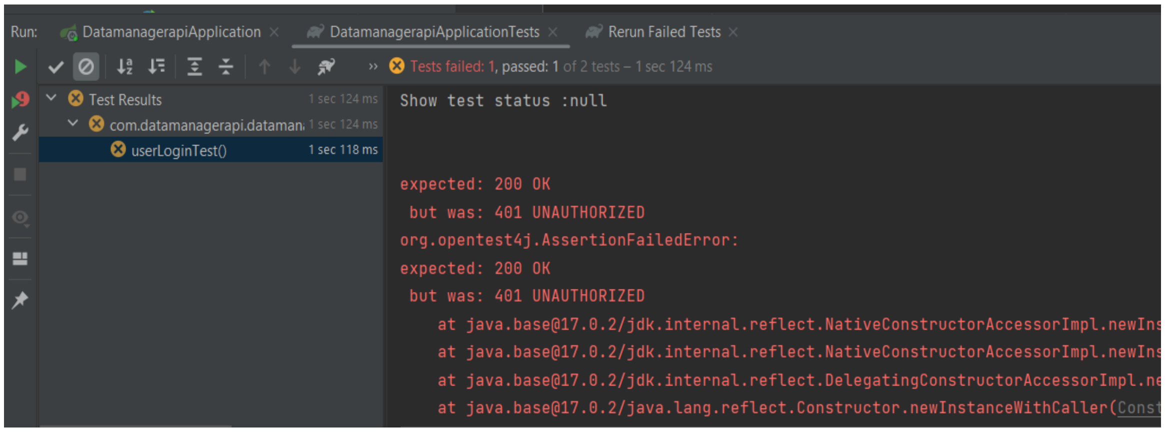
Configuration in code



```
@GetMapping(value = "/allInstitutions")
@Cacheable("institution-cache")
public List<Institution> retrieveAllInstitutions() { return i

}
```

➢ Test driven Development

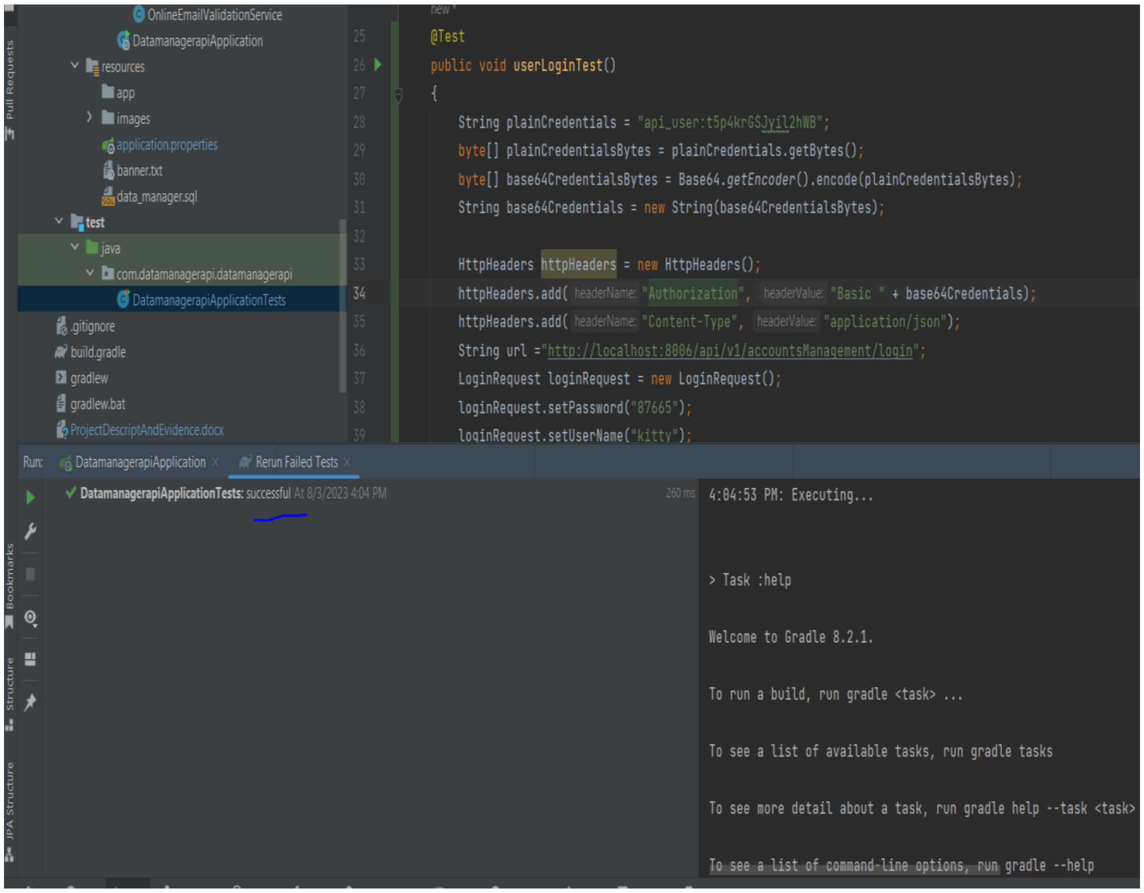
❖ User login test without api credentials being passed, fails with 401 status code



Run: DatamanagerapiApplication × DatamanagerapiApplicationTests × Rerun Failed Tests ×
Tests failed: 1, passed: 1 of 2 tests – 1 sec 124 ms
Test Results 1 sec 124 ms
com.datamanagerapi.datamana 1 sec 124 ms
userLoginTest() 1 sec 118 ms

```
Show test status :null


expected: 200 OK
 but was: 401 UNAUTHORIZED
org.opentest4j.AssertionFailedError:
expected: 200 OK
 but was: 401 UNAUTHORIZED
	at java.base@17.0.2/jdk.internal.reflect.NativeConstructorAccessorImpl.newIns
	at java.base@17.0.2/jdk.internal.reflect.NativeConstructorAccessorImpl.newIns
	at java.base@17.0.2/jdk.internal.reflect.DelegatingConstructorAccessorImpl.ne
	at java.base@17.0.2/java.lang.reflect.Constructor.newInstanceWithCaller(Const
```

❖ After supplied the api credentials in the header, the test passes



OnlineEmailValidationService
DatamanagerapiApplication
resources
app
images
application.properties
banner.txt
data_manager.sql
test
java
com.datamanagerapi.datamanagerapi
DatamanagerapiApplicationTests
.gitignore
build.gradle
gradlew
gradlew.bat
ProjectDescriptAndEvidence.docx
Run: DatamanagerapiApplication × Rerun Failed Tests ×
DatamanagerapiApplicationTests: successful At 8/3/2023 4:04 PM
260 ms

```
@Test
public void userLoginTest()
{
    String plainCredentials = "api_user:t5p4krGSJyil2hWB";
    byte[] plainCredentialsBytes = plainCredentials.getBytes();
    byte[] base64CredentialsBytes = Base64.getEncoder().encode(plainCredentialsBytes);
    String base64Credentials = new String(base64CredentialsBytes);

    HttpHeaders httpHeaders = new HttpHeaders();
    httpHeaders.add( headerName: "Authorization",  headerValue: "Basic " + base64Credentials);
    httpHeaders.add( headerName: "Content-Type",  headerValue: "application/json");
    String url ="http://localhost:8006/api/v1/accountsManagement/login";
    LoginRequest loginRequest = new LoginRequest();
    loginRequest.setPassword("87665");
    loginRequest.setUserName("kitty");
```

```
4:04:53 PM: Executing...


> Task :help

Welcome to Gradle 8.2.1.

To run a build, run gradle <task> ...

To see a list of available tasks, run gradle tasks

To see more detail about a task, run gradle help --task <task>

To see a list of command-line options, run gradle --help
```

END