**ANZ User Account Service REST API**

Technologies used to develop this service:

* Java 8
* Spring boot
* Restful
* H2 database(in memory database)
* Junit
* EhCache
* Swagger2

**Overview:**

1. **Loading Account details & Transaction details When Server Starting Up:**

* Account details will insert into ACCOUNT\_DETAILS Table based on static data file **data.sql** maintained in **resource** (**src/main/resources)** folder When Server Up.
* Transaction details will insert into TRANSACTION\_DETAILS Table based on static data file **data.sql** maintained in **resource** (**src/main/resources)** folder When Server Up.
* User details will insert into USER\_DETAILS Table based on static data file **data.sql** maintained in **resource** (**src/main/resources)** folder When Server Up.

1. **Account Details Fetch service:**

**Based on logged in User Id request parameter below functionalities will perform.**

* **If User related account data is available for given user Id , then will return Account details for given User Id ,otherwise system will return message like** return message like Account Details not available for user::XXXXXX.

1. **Transaction Details Fetch service:**

**Based on logged in user holding Account Number request parameter below functionalities will perform.**

* **If Transaction data is available for given account number then will return Transaction details for given account, otherwise system will return message like** Transaction Details not available for Account::XXXX.

1. **Back To User Account Details service:**

**Service will back to User Account details based on logged in User and fetch user account details form Ehcache.**

**Unit Testing:**

* Account details fetch functionalities and Account details not available test cases are covered by using TestRestTemplate functionality.
* Transaction details fetch functionalities and Transaction details not available test cases are covered by using TestRestTemplate functionality.

**Exception handling:**

Here I have created few custom exception classes to handle edge cases.

Exception#1: If Account details not available for specific user id then throw ***AccountDetailsNotFoundException("***Account Details not available for user")

Exception#2: If transaction details is not available for given account number then throw ***TransactionDetailsNotFoundException***("Transaction Details not available for Account::")

**Logger and Documentation:**

Logger and documentation is cover for very method in all layers.

**Procedure to run the application:**

1. Import project

* Import the project -> Maven Existing Projects.
* Configure JDK as java build path.

1. Build project:

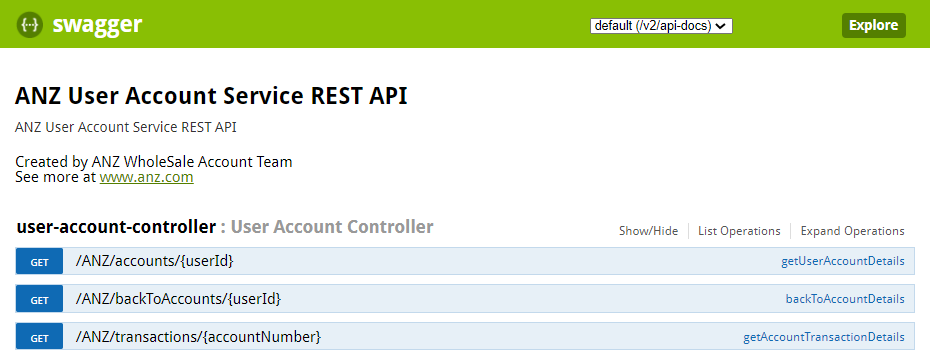
* -> mvn clean install

1. Run project

* -> mvn spring-boot:run(Run as->Spring Boot App)

**Swagger2** Once service up and running, Here exposing 2 services with swagger API.

<http://localhost:8990/swagger-ui.html#/user-account-controller>

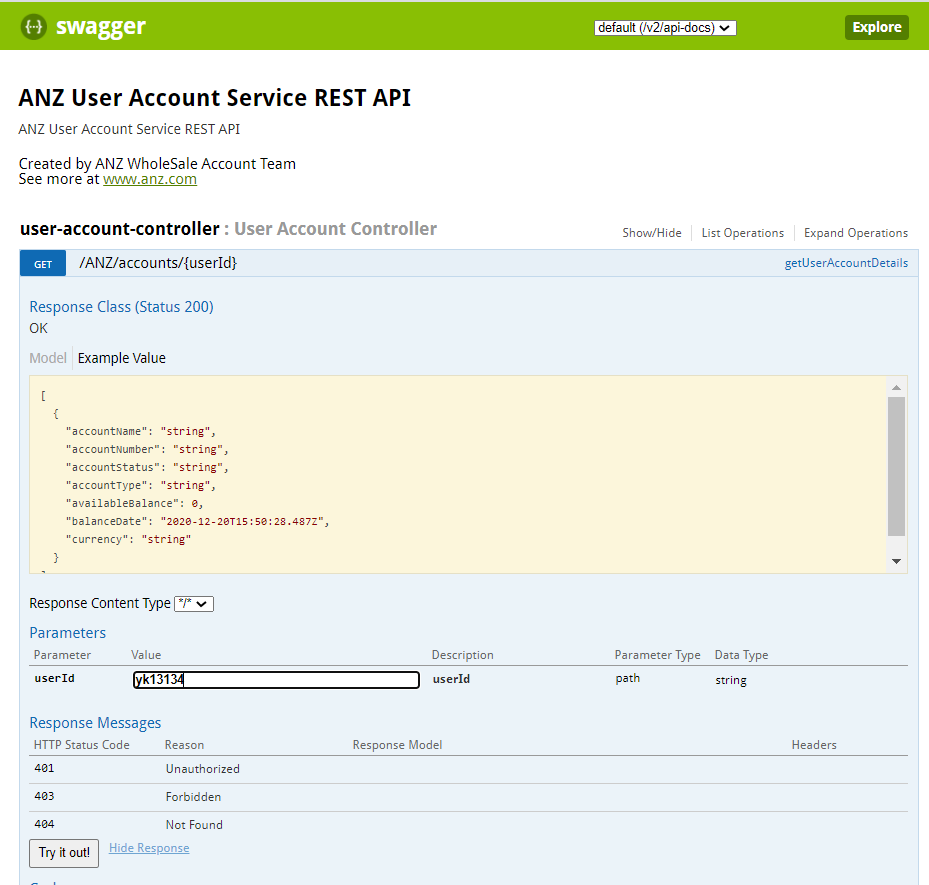


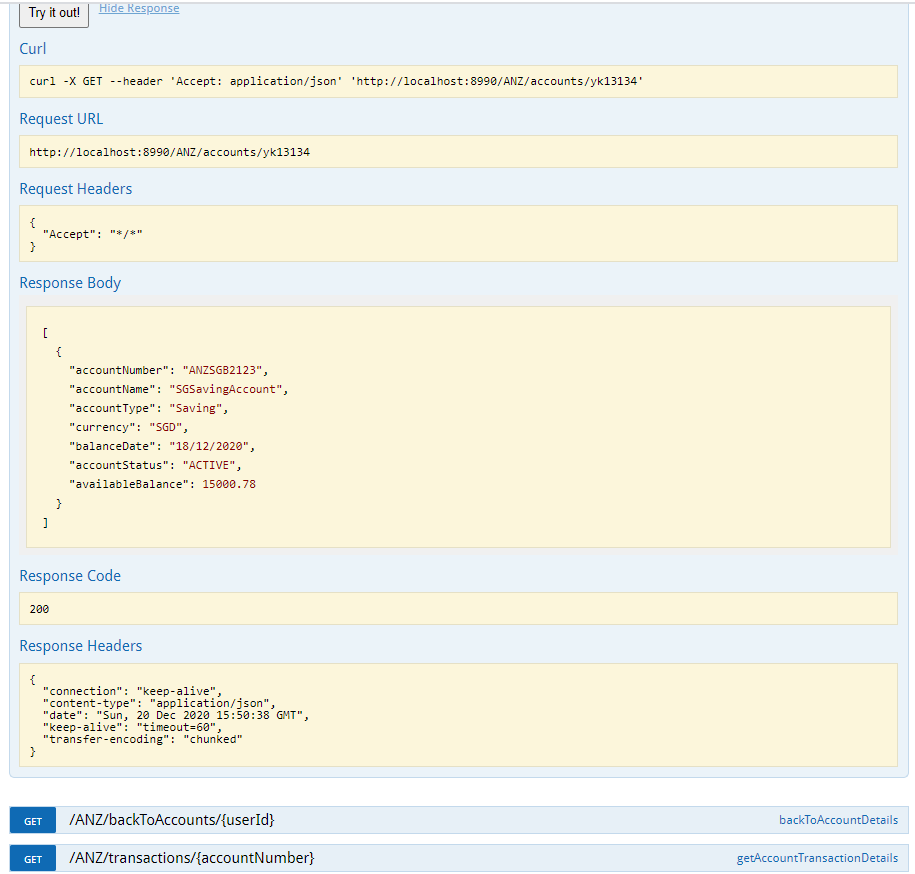
1. **Account Details Fetch service:**

### Get Endpoint URI: http://localhost:8990/ANZ/accounts/{userId}

### sample: ://localhost:8990/ANZ/accounts/yk13134

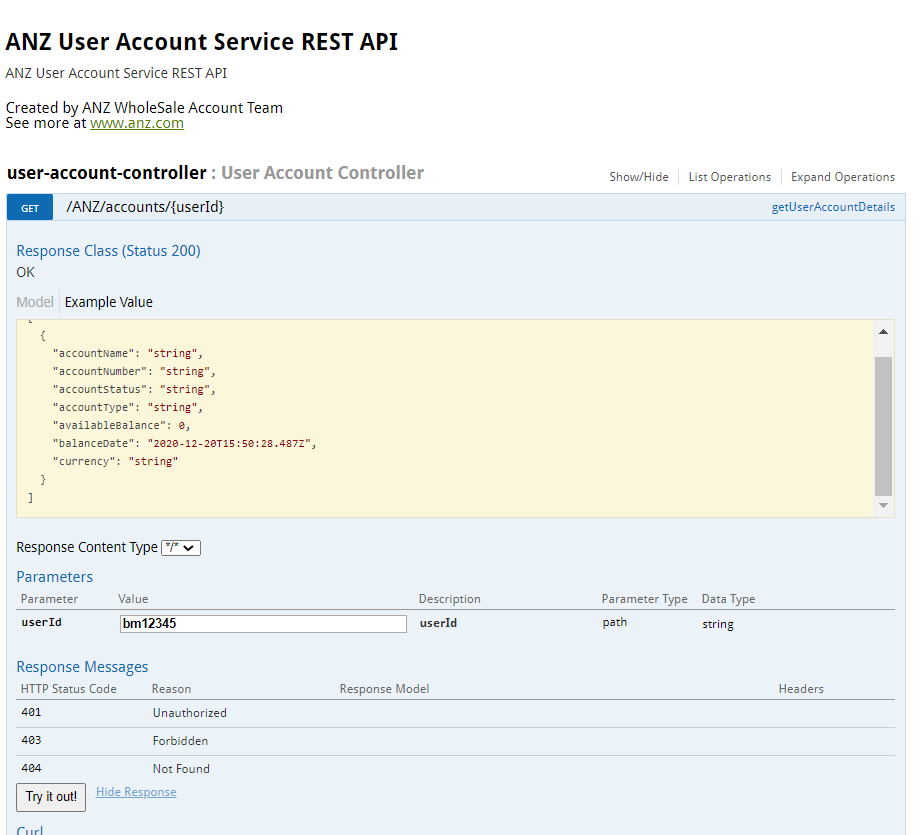
Sample request UserId is **yk13134 / am13543** as data setup





**Case#01🡪Account Details not available for given UserId** bm12345

sample::http://localhost:8990/ANZ/accounts/ bm12345

****

****

2) **Transaction Details Fetch service:**Get Endpoint URI: http://localhost:8990[/ANZ/transactions/{accountNumber}](http://localhost:8990/swagger-ui.html#!/user-account-controller/getAccountTransactionDetailsUsingGET)

sample: http://localhost:8990[/ANZ/transactions/ ANZSGB2123](http://localhost:8990/swagger-ui.html#!/user-account-controller/getAccountTransactionDetailsUsingGET)

Response body:

[

{

"accountNumber": "ANZSGB2123",

"accountName": "SGSavingAccount",

"currency": "SGD",

"valueDate": "Jun 02 2020",

"debitAmount": null,

"creditAmount": 2000,

"transactionType": "Credit",

"transactionNarrative": ""

},

{

"accountNumber": "ANZSGB2123",

"accountName": "SGSavingAccount",

"currency": "SGD",

"valueDate": "May 02 2020",

"debitAmount": null,

"creditAmount": 3000,

"transactionType": "Credit",

"transactionNarrative": ""

},

{

"accountNumber": "ANZSGB2123",

"accountName": "SGSavingAccount",

"currency": "SGD",

"valueDate": "Jun 30 2020",

"debitAmount": null,

"creditAmount": 4000,

"transactionType": "Credit",

"transactionNarrative": ""

},

{

"accountNumber": "ANZSGB2123",

"accountName": "SGSavingAccount",

"currency": "SGD",

"valueDate": "Aug 12 2020",

"debitAmount": null,

"creditAmount": 5000,

"transactionType": "Credit",

"transactionNarrative": ""

},

{

"accountNumber": "ANZSGB2123",

"accountName": "SGSavingAccount",

"currency": "SGD",

"valueDate": "Sep 22 2020",

"debitAmount": null,

"creditAmount": 1000,

"transactionType": "Credit",

"transactionNarrative": ""

},

{

"accountNumber": "ANZSGB2123",

"accountName": "SGSavingAccount",

"currency": "SGD",

"valueDate": "Oct 12 2020",

"debitAmount": 8000,

"creditAmount": null,

"transactionType": "Debit",

"transactionNarrative": "pay to family"

},

{

"accountNumber": "ANZSGB2123",

"accountName": "SGSavingAccount",

"currency": "SGD",

"valueDate": "Nov 27 2020",

"debitAmount": null,

"creditAmount": 6000,

"transactionType": "Credit",

"transactionNarrative": ""

},

{

"accountNumber": "ANZSGB2123",

"accountName": "SGSavingAccount",

"currency": "SGD",

"valueDate": "Nov 29 2020",

"debitAmount": null,

"creditAmount": 500.5,

"transactionType": "Credit",

"transactionNarrative": ""

},

{

"accountNumber": "ANZSGB2123",

"accountName": "SGSavingAccount",

"currency": "SGD",

"valueDate": "Dec 02 2020",

"debitAmount": null,

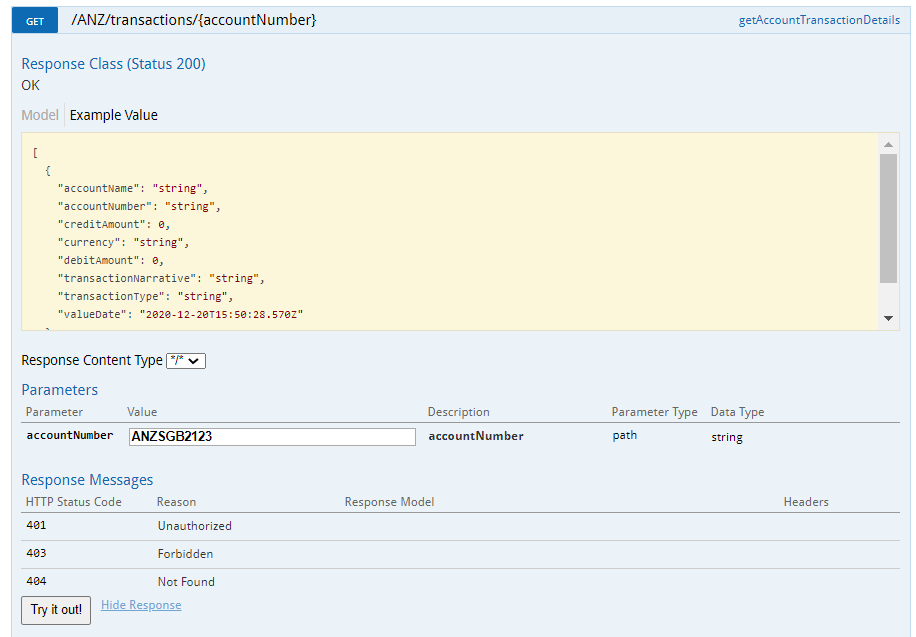
"creditAmount": 1500.28,

"transactionType": "Credit",

"transactionNarrative": ""

}

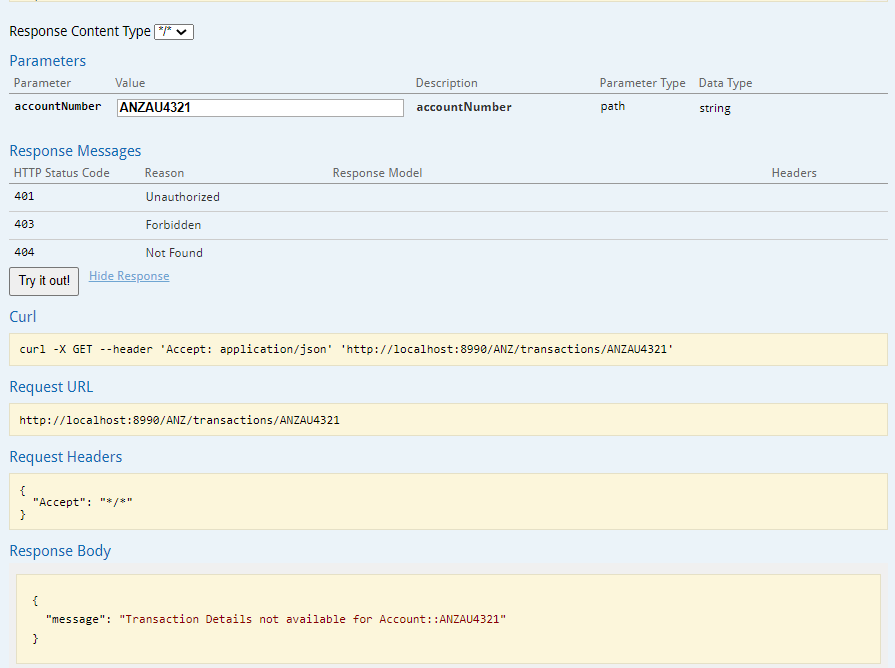
]





**Case#01🡪Transaction Details not available for given Account Number**  ANZAU4321

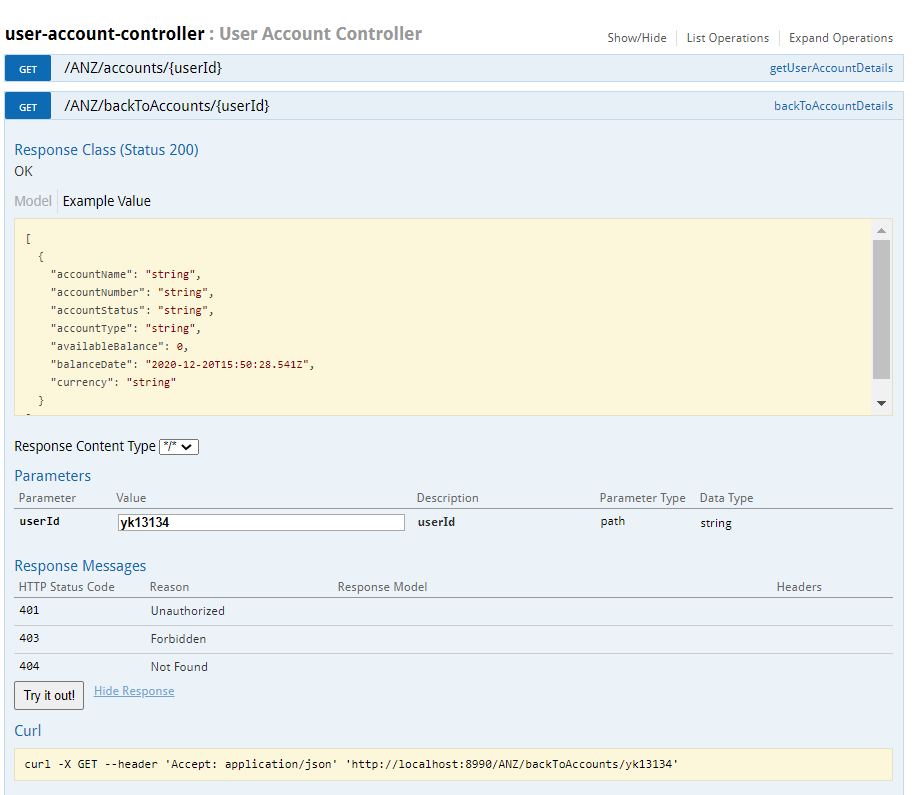
sample: http://localhost:8990[/ANZ/transactions/ ANZAU4321](http://localhost:8990/swagger-ui.html#!/user-account-controller/getAccountTransactionDetailsUsingGET)

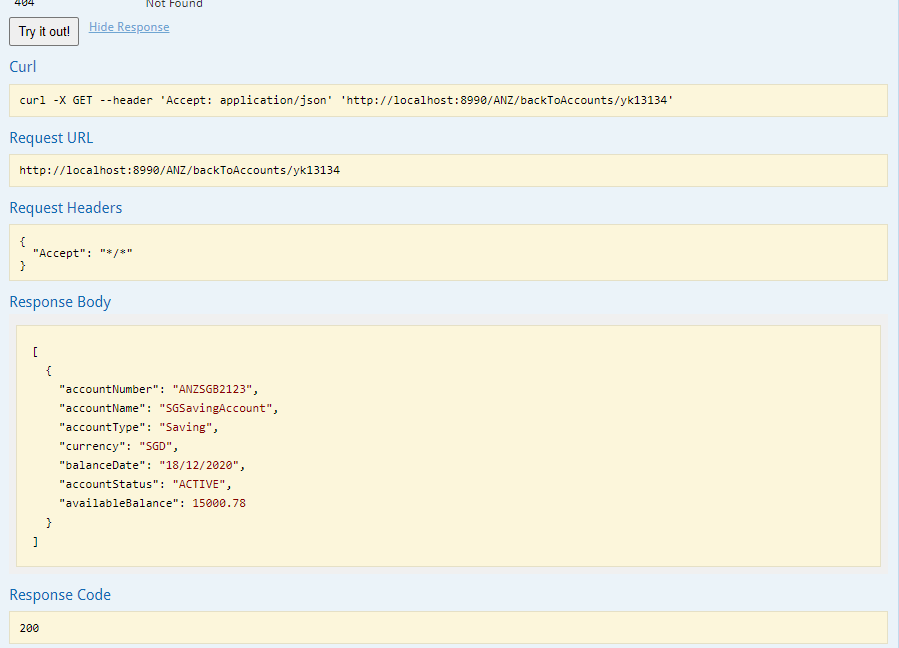


**3)Back To User Account Details service:**

### Get Endpoint URI: http://localhost:8990/ANZ/accounts/{userId}

### sample: ://localhost:8990/ANZ/accounts/yk13134





**Fetch account Details hit database for fetching account details**

2020-12-20 23:50:37.696 INFO 25688 --- [nio-8990-exec-8] c.a.a.controller.UserAccountController : getUserAccountDetails(..) method execution Start

2020-12-20 23:50:37.751 INFO 25688 --- [nio-8990-exec-8] c.a.a.service.AccountDetailsServiceImpl : findAccountByUserIdAndStatus(..) method execution

Hibernate: select accountdet0\_.account\_number as account\_1\_0\_, accountdet0\_.available\_balance as availabl2\_0\_, accountdet0\_.account\_name as account\_3\_0\_, accountdet0\_.account\_status as account\_4\_0\_, accountdet0\_.account\_type as account\_5\_0\_, accountdet0\_.balance\_date as balance\_6\_0\_, accountdet0\_.currency as currency7\_0\_, accountdet0\_.acct\_user\_id as acct\_use8\_0\_ from account\_details accountdet0\_ where accountdet0\_.acct\_user\_id=? and accountdet0\_.account\_status=?

2020-12-20 23:50:37.982 INFO 25688 --- [nio-8990-exec-8] c.a.a.service.AccountDetailsServiceImpl : convertAccountDetailsToVOObject(..) method execution Start

2020-12-20 23:50:38.094 INFO 25688 --- [nio-8990-exec-8] c.a.a.service.AccountDetailsServiceImpl : convertAccountDetailsToVOObject(..) method execution End

2020-12-20 23:50:38.095 INFO 25688 --- [nio-8990-exec-8] c.a.a.controller.UserAccountController : getUserAccountDetails(..) method execution End

2020-12-20 23:57:32.279 INFO 25688 --- [nio-8990-exec-2] c.a.a.controller.UserAccountController : getUserAccountDetails(..) method execution Start

2020-12-20 23:57:32.279 INFO 25688 --- [nio-8990-exec-2] c.a.a.service.AccountDetailsServiceImpl : findAccountByUserIdAndStatus(..) method

Execution

**Below Find below Logs with out hitting database again,Accouint result have fetched from cache.**

2020-12-21 00:10:00.528\_[0;39m \_[32m INFO\_[0;39m \_[35m25688\_[0;39m \_[2m---\_[0;39m \_[2m[nio-8990-exec-8]\_[0;39m \_[36mc.a.a.controller.UserAccountController \_[0;39m \_[2m:\_[0;39m backToAccountDetails(..) method execution End

\_[2m2020-12-21 00:16:11.767\_[0;39m \_[32m INFO\_[0;39m \_[35m25688\_[0;39m \_[2m---\_[0;39m \_[2m[io-8990-exec-10]\_[0;39m \_[36mc.a.a.controller.UserAccountController \_[0;39m \_[2m:\_[0;39m backToAccountDetails(..) method execution Start

\_[2m2020-12-21 00:16:11.767\_[0;39m \_[32m INFO\_[0;39m \_[35m25688\_[0;39m \_[2m---\_[0;39m \_[2m[io-8990-exec-10]\_[0;39m \_[36mc.a.a.service.AccountDetailsServiceImpl \_[0;39m \_[2m:\_[0;39m convertAccountDetailsToVOObject(..) method execution Start

\_[2m2020-12-21 00:16:11.768\_[0;39m \_[32m INFO\_[0;39m \_[35m25688\_[0;39m \_[2m---\_[0;39m \_[2m[io-8990-exec-10]\_[0;39m \_[36mc.a.a.service.AccountDetailsServiceImpl \_[0;39m \_[2m:\_[0;39m convertAccountDetailsToVOObject(..) method execution End

\_[2m2020-12-21 00:16:11.769\_[0;39m \_[32m INFO\_[0;39m \_[35m25688\_[0;39m \_[2m---\_[0;39m \_[2m[io-8990-exec-10]\_[0;39m \_[36mc.a.a.controller.UserAccountController \_[0;39m \_[2m:\_[0;39m backToAccountDetails(..) method execution End

**Note:**

* I assumed entitlement already done, so I haven't verified user existence in this API.
* I would like to implement security using JWT(Json Web Token) for this API but unfortunately I couldn't spend sufficient time on this, because of my deliverables.