



Capstone Project for GDP 2021

Done by:

Hajar Al.tuwaigri

30 Nov 2021

Capstone Project for GDP 2021	1
1) Database (PostgresSQL)	3
2) Project Architecture	4
3) Testing	5
4) Appendix	17

1) Database (PostgreSQL)

1) Create the Transaction table in the pgAdmin 4:

The screenshot shows the pgAdmin 4 interface for creating a new table named 'transaction'. The 'Columns' tab is active, displaying 12 columns with the following details:

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
transaction_id	bigint			Enabled	Enabled	nextval('transaction_transacti')
from_account	bigint			Disabled	Disabled	
to_account	bigint			Disabled	Enabled	
from_account_name	text			Disabled	Enabled	
to_account_name	text			Disabled	Enabled	
same_bank_transaction	boolean			Disabled	Enabled	
other_bank	text			Disabled	Enabled	
amount	double precision			Disabled	Enabled	
date	date			Disabled	Enabled	
time	timestamp with time zone			Disabled	Enabled	
transaction_type	text			Disabled	Enabled	
status	text			Disabled	Enabled	

Buttons at the bottom include Close, Reset, and Save.

2) Run the Data Base :

The screenshot shows the pgAdmin 4 interface with the 'transaction' table selected in the browser tree under 'Tables (2) > account > transaction'. The 'Dashboard' tab is selected, displaying several performance monitoring charts and a table of server activity.

Database sessions: Shows Total (blue), Active (green), and Idle (red) sessions. Total is around 10, Active is around 1.

Transactions per second: Shows Transactions (blue), Commits (green), and Rollbacks (red). Peaks are around 3 transactions per second.

Tuples in: Shows Inserts (blue), Updates (green), and Deletes (red). Total is around 1.

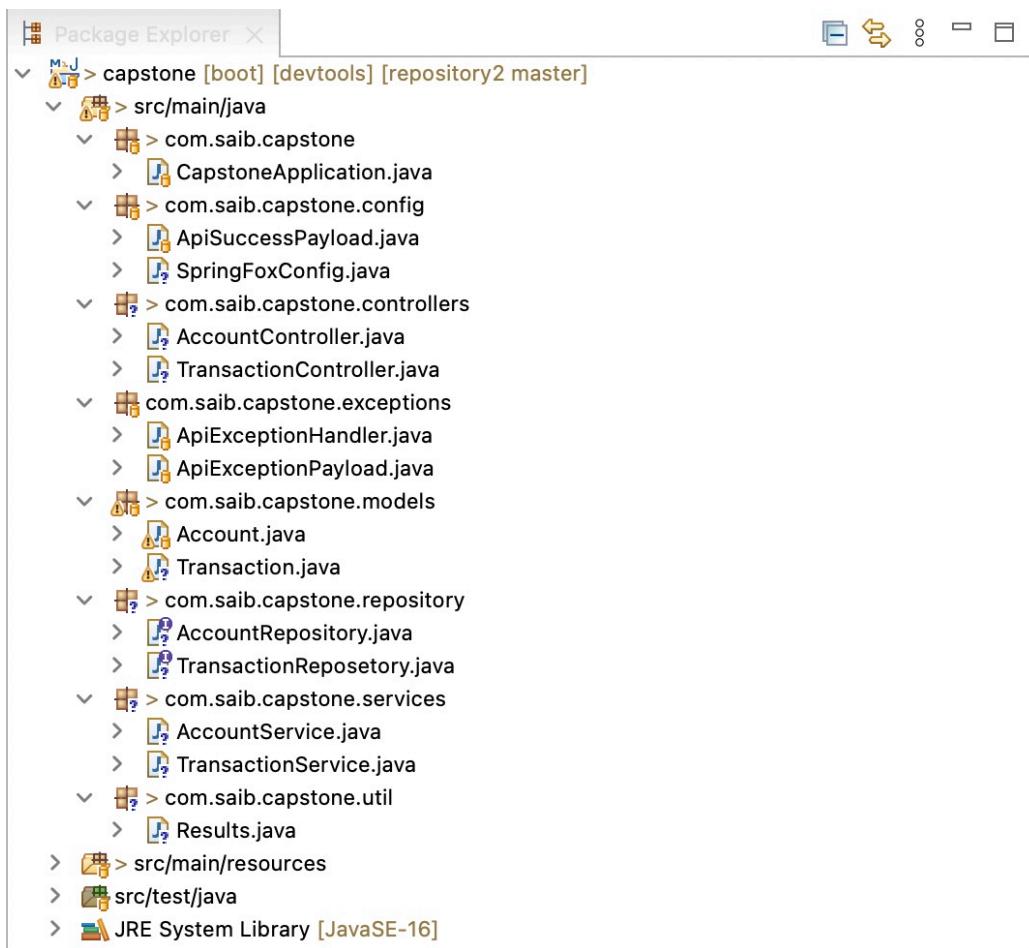
Tuples out: Shows Fetched (blue), Returned (green), and Deleted (red). Total is around 1400.

Block I/O: Shows Reads (blue) and Hits (green). Total is around 90.

Server activity: Shows Sessions, Locks, and Prepared Transactions. One session is listed: PID 910, User postgres, Application pgAdmin 4 - DB:saib, Client ::1, Backend start 2021-12-01 09:05:40 +03, State active, Wait event, Blocking PIDs.

2) Project Architecture

Our Capstone Project File Architecture :



3) Testing

Run the “jar file” to check the api’s by use “swagger tool” :

```
hajar_macbook_pro@Hajars-MBP Jar File % java -jar capstone-0.0.1-SNAPSHOT.jar

:: Spring Boot :: (v2.6.0)

2021-11-30 19:35:23,544 INFO 7579 --- [           main] com.saib.capstone.CapstoneApplication : Starting CapstoneApplication v0.0.1-SNAPSHOT using Java 17 on Hajars-MBP with
PID 7579 (/Users/hajar_macbook_pro/Desktop/Capstone Project/Jar File /capstone-0.0.1-SNAPSHOT.jar started by hajar_macbook_pro in /Users/hajar_macbook_pro/Desktop/Capstone Project/Jar File )
2021-11-30 19:35:23,547 INFO 7579 --- [           main] com.saib.capstone.CapstoneApplication : No active profile set, falling back to default profiles: default
2021-11-30 19:35:24,599 INFO 7579 --- [           main] s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2021-11-30 19:35:24,669 INFO 7579 --- [           main] s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 56 ms. Found 2 JPA repository interfaces.
2021-11-30 19:35:25,053 INFO 7579 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'io.sentry.spring.tracing.SentryAdviceConfiguration' of type [io.sentry.spring.tracing.SentryAdviceConfiguration] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
2021-11-30 19:35:25,059 INFO 7579 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'io.sentry.spring.tracing.SentrySpanPointcutConfiguration' of type [io.sentry.spring.tracing.SentrySpanPointcutConfiguration] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
2021-11-30 19:35:25,063 INFO 7579 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'sentrySpanPointcut' of type [org.springframework.aop.support.ComposablePointcut] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
2021-11-30 19:35:25,064 INFO 7579 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'io.sentry.spring.boot.SentryAutoConfiguration$HubConfiguration' of type [io.sentry.spring.boot.SentryAutoConfiguration$HubConfiguration] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
2021-11-30 19:35:25,074 INFO 7579 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'inAppPackagesResolver' of type [io.sentry.spring.boot.InAppIncludesResolver] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
2021-11-30 19:35:25,076 INFO 7579 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'sentryOptionsConfiguration' of type [io.sentry.spring.boot.SentryAutoConfiguration$HubConfiguration$Lambda$457$0x0000000000000053b0] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
2021-11-30 19:35:25,245 INFO 7579 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'sentryIo.sentry.spring.boot.SentryProperties' of type [io.sentry.spring.boot.SentryProperties] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
2021-11-30 19:35:25,255 INFO 7579 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'capstoneApplication' of type [com.saib.capstone.CapstoneApplication$$EnhancerBySpringGCLIBS$$c2a508cd] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
2021-11-30 19:35:25,287 INFO 7579 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'sentryHub' of type [io.sentry.HubAdapter] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
2021-11-30 19:35:25,293 INFO 7579 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'sentrySpanAdvice' of type [io.sentry.spring.tracing.SentrySpanAdvice] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
2021-11-30 19:35:25,298 INFO 7579 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'sentrySpanAdvisor' of type [org.springframework.aop.support.DefaultPointCutAdvisor] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
2021-11-30 19:35:25,298 INFO 7579 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'io.sentry.spring.tracing.SentryTransactionPointcutConfiguration' of type [io.sentry.spring.tracing.SentryTransactionPointcutConfiguration] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
2021-11-30 19:35:25,297 INFO 7579 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'sentryTransactionPointcut' of type [org.springframework.aop.support.ComposablePointcut] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
2021-11-30 19:35:25,299 INFO 7579 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'sentryTransactionAdvice' of type [io.sentry.spring.tracing.SentryTransactionAdvice] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
2021-11-30 19:35:25,300 INFO 7579 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'sentryTransactionAdvisor' of type [org.springframework.aop.support.DefaultPointCutAdvisor] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
2021-11-30 19:35:25,709 INFO 7579 --- [           main] o.s.w.e.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 9090 (https)
2021-11-30 19:35:25,727 INFO 7579 --- [           main] o.apache.catalina.core.StandardService : Starting service [tomcat]
2021-11-30 19:35:25,728 INFO 7579 --- [           main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.55]
2021-11-30 19:35:25,790 INFO 7579 --- [           main] o.a.c.c.[Tomcat].[localhost].[] : Initializing Spring Embedded WebApplicationContext
2021-11-30 19:35:25,790 INFO 7579 --- [           main] w.s.c.WebApplicationInitializer : Root WebApplicationContext: initialization completed in 2181 ms
```

Transaction Test : Swagger Tool:

The screenshot shows the Swagger UI interface for a RESTful API. At the top, there are tabs for "Swagger UI", "Console" (disabled), "Metrics (Fajar) - GitHub", "Stack Configuration code: IDV-DTS - Fajar...", and "Swagger UI". A search bar says "Select a spec" with "default" selected. Below the header, the title "Api Documentation 1.0" is displayed, along with the base URL "[Base URL: localhost:8080 /]" and the API documentation URL "[https://localhost:8080/v2/api-docs]". There are links for "Terms of service" and "Apache 2.0".

The main content area shows three controller groups: "account-controller", "basic-error-controller", and "transaction-controller". The "transaction-controller" group is expanded, showing its methods:

- GET** /transaction getAllTransaction
- POST** /transaction addTransaction
- GET** /transaction/{transactionId} getTransactionByTransactionId
- PUT** /transaction/{transactionId} updateTransaction
- DELETE** /transaction/{transactionId} deleteTransaction
- GET** /transaction/{transactionType}/{date} findTransactionByDateAndTransactionType
- GET** /transaction/all/ getAllTransaction
- GET** /transaction/all/sorted getAllTransaction
- GET** /transaction/date/{date} findTransactionByDate
- GET** /transaction/transactionType/{transactionType} findTransactionByTransactionType

At the bottom, there is a "Models" section.

1) addTransaction

POST /transaction addTransaction

Cancel

Parameters

Name	Description
transaction * required (body)	transaction

Example ValueModel

```
{
  "amount": 5000,
  "date": "2021-12-01T08:58:53.921Z",
  "fromAccount": 1101154,
  "fromAccountName": "Mohammad",
  "otherBank": "no",
  "sameBankTransaction": true,
  "status": "active",
  "time": "2021-12-01T08:58:53.921Z",
  "toAccount": 100104237,
  "toAccountName": "Maher",
  "transactionId": 1135231204,
  "transactionType": "credit"
}
```

Cancel

Parameter content type

application/json

Execute Clear

Responses

Response content type */*

Curl

```
curl -X POST "https://localhost:8080/transaction" -H "accept: /*" -H "Content-Type: application/json" -d "{ \"amount\": 5000, \"date\": \"2021-12-01T08:58:53.921Z\", \"fromAccount\": 1101154, \"fromAccountName\": \"Mohammad\", \"otherBank\": \"no\", \"sameBankTransaction\": true, \"status\": \"active\", \"time\": \"2021-12-01T08:58:53.921Z\", \"toAccount\": 100104237, \"toAccountName\": \"Maher\", \"transactionId\": 1135231204, \"transactionType\": \"credit\"}"
```

Request URL

https://localhost:8080/transaction

Server response

Code	Details
201	<p>Response body</p> <pre>{ "message": "Transaction created successfully", "status": 201, "httpStatus": "201 CREATED", "success": true, "exception": false, "body": "success" }</pre> <p>Download</p> <p>Response headers</p> <pre>connection: keep-alive content-type: application/json date: Wed, 01 Dec 2021 09:09:07 GMT keep-alive: timeout=60 transfer-encoding: Identity</pre>

please check “Post_Output.csv” file

2) getAllTransaction

GET /transaction getAllTransaction

Parameters Cancel

No parameters

Execute Clear

Responses Response content type */*

Curl

```
curl -X GET "https://localhost:8080/transaction" -H "accept: */*"
```

Request URL

```
https://localhost:8080/transaction
```

Server response

Code	Details
200	<p>Response body</p> <pre>{ "message": "Transaction Fetched", "status": 200, "httpStatus": "200 OK", "success": true, "exception": false, "body": [{ "transactionId": 113523100, "fromAccount": 1101103, "toAccount": 10010022, "fromAccountName": "ALEX", "toAccountName": "sam", "sameBankTransaction": true, "otherBank": "no", "amount": 5000, "date": "1970-01-01T08:58:53.921", "time": "1970-01-01T08:58:53.921".] }</pre> <p style="text-align: right;">Download</p> <p>Response headers</p> <pre>connection: keep-alive content-type: application/json date: Wed, 01 Dec 2021 09:16:03 GMT keep-alive: timeout=60 transfer-encoding: Identity</pre>

Responses

Code	Description
200	<p>OK</p> <p>Example ValueModel</p> <pre>{ "body": {}, "exception": true, "httpStatus": "string", "message": "string", "status": 0 }</pre>

3) getTransactionByTransactionId

GET /transactionss/{transactionId} getTransactionByTransactionId

Cancel

Parameters

Name	Description
transactionId <small>required</small> integer(\$int64) (path)	transactionId 113523100

Execute Clear

Responses Response content type */*

Curl

```
curl -X GET "https://localhost:8080/transactionss/113523100" -H "accept: */*"
```

Request URL

```
https://localhost:8080/transactionss/113523100
```

Server response

Code	Details				
200	<p>Response body</p> <pre>{ "message": "Success", "status": 200, "httpStatus": "200 OK", "success": true, "exception": false, "body": { "transactionId": 113523100, "fromAccount": 1101103, "toAccount": 10010022, "fromAccountName": "ALEX", "toAccountName": "sam", "sameBankTransaction": true, "otherBank": "no", "amount": 5000, "date": "1970-01-01T08:58:53.921", "time": "1970-01-01T08:58:53.921", "transactionType": "credit" } }</pre> <p>Download</p> <p>Response headers</p> <pre>connection: keep-alive content-type: application/json date: Wed, 01 Dec 2021 09:26:24 GMT keep-alive: timeout=60 transfer-encoding: Identity</pre> <p>Responses</p> <table border="1"><thead><tr><th>Code</th><th>Description</th></tr></thead><tbody><tr><td>200</td><td>OK</td></tr></tbody></table> <p>Example ValueModel</p>	Code	Description	200	OK
Code	Description				
200	OK				

4) findTransactionByTransactionType

GET /transaction/transactionType/{transactionType} getTransactionByTransactionType

Cancel

Parameters

Name	Description
transactionType * required string (path)	transactionType Credit

Execute Clear

Responses Response content type */*

Curl

```
curl -X GET "https://localhost:8080/transaction/transactionType/Credit%20" -H "accept: */*"
```

Request URL

```
https://localhost:8080/transaction/transactionType/Credit%20
```

Server response

Code	Details
200	<p>Response body</p> <pre>{ "message": "Transaction Found", "status": 200, "httpStatus": "200 OK", "success": true, "exception": false, "body": [] }</pre> <p>Download</p> <p>Response headers</p> <pre>connection: keep-alive content-type: application/json date: Wed, 01 Dec 2021 09:31:44 GMT keep-alive: timeout=60 transfer-encoding: Identity</pre>

Responses

Code	Description
200	OK

Example ValueModel

```
{  
    "body": {},  
    "exception": true,  
    "httpStatus": "string",  
    "message": "string",  
    "status": 0,  
    "success": true  
}
```

5) getAllTransaction (sorted data)

GET /transaction/all/sorted getAllTransaction

Parameters

Name	Description
pageNumber * required integer(\$int32) (query)	pageNumber 0
pageSize * required integer(\$int32) (query)	pageSize 7
sortBy * required string (query)	sortBy transactionType

Responses

Curl

```
curl -X GET "https://localhost:8080/transaction/all/sorted?pageNumber=0&pageSize=7&sortBy=transactionType" -H "accept: */*"
```

Request URL

Server response

Code	Details
200	Response body <pre>{ "message": "Transaction Found", "status": 200, "httpStatus": "200 OK", "success": true, "exception": false, "body": [{ "transactionId": 1135231111, "fromAccount": 11011144, "toAccount": 100104222, "fromAccountName": "Jasmin", "toAccountName": "Emaan", "sameBankTransaction": true, "otherBank": "no", "amount": 5000, "date": "1970-01-01T08:58:53.921", "time": "1970-01-01T08:58:53.921". }] }</pre> <p>Download</p> Response headers <pre>connection: keep-alive content-type: application/json date: Wed, 01 Dec 2021 10:18:58 GMT keep-alive: timeout=60 transfer-encoding: Identity</pre>

please check “Sorted_ByType.json” file

6) updateTransaction

PUT /transaction/{transactionId} updateTransaction

Parameters

Cancel

Name	Description
transaction * required (body)	transaction
transactionId * required integer(\$int64) (path)	transactionId

Example ValueModel

```
{  
    "transactionId": 113523100,  
    "fromAccount": 1101103,  
    "toAccount": 10010022,  
    "fromAccountName": "ALEX",  
    "toAccountName": "sam",  
    "sameBankTransaction": true,  
    "otherBank": "no",  
    "amount": 300,  
    "date": "1970-01-01T08:58:53.921",  
    "time": "1970-01-01T08:58:53.921",  
    "transactionType": "credit",  
    "status": "active"  
}
```

Cancel

Parameter content type
application/json

transactionId
113523100

Execute Clear

Responses

Response content type */*

Curl

```
curl -X PUT "https://localhost:8080/transaction/113523100" -H "accept: */*" -H "Content-Type: application/json" -d "{  
    \"transactionId\": 113523100, \"fromAccount\": 1101103, \"toAccount\": 10010022, \"fromAccountName\": \"ALEX\",  
    \"toAccountName\": \"sam\", \"sameBankTransaction\": true, \"otherBank\": \"no\", \"amount\": 300, \"date\": \"1970-01-01T08:58:53.921\", \"time\": \"1970-01-01T08:58:53.921\", \"transactionType\": \"credit\", \"status\": \"active\" }"
```

Request URL
<https://localhost:8080/transaction/113523100>

Server response

Code	Details
200	Response body <pre>{ "message": "success", "status": 200, "httpStatus": "200 OK", "success": true, "exception": false, "body": "success" }</pre> <p>Download</p> Response headers

7) deleteTransaction

The screenshot shows a REST API tool interface for the `deleteTransaction` endpoint. At the top, there is a red header bar with the method `DELETE` and the URL `/transaction/{transactionId}`. Below this is a table for parameters:

Name	Description
<code>transactionId</code> * required integer(\$int64) (path)	transactionId 113523100

Below the parameters are two buttons: `Execute` (blue) and `Clear` (grey). Under the `Responses` section, there is a `Curl` command and a `Request URL`. The `Server response` section shows a `Code` of `200` and a `Details` section containing a JSON response body and response headers. The `Example ValueModel` section shows a JSON schema for the response body.

Parameters

Name	Description
<code>transactionId</code> * required integer(\$int64) (path)	transactionId 113523100

Responses

Curl

```
curl -X DELETE "https://localhost:8080/transaction/113523100" -H "accept: */*"
```

Request URL

```
https://localhost:8080/transaction/113523100
```

Server response

Code	Details
200	Response body <pre>{ "message": "success", "status": 200, "httpStatus": "200 OK", "success": true, "exception": false, "body": "success" }</pre> Download Response headers <pre>connection: keep-alive content-type: application/json date: Wed, 01 Dec 2021 10:32:31 GMT keep-alive: timeout=60 transfer-encoding: Identity</pre>

Responses

Code	Description
200	OK

Example ValueModel

```
{  
    "body": {},  
    "exception": true,  
    "httpStatus": "string",  
    "message": "string",  
    "status": 0,  
    "success": true  
}
```

Please check “transactionAfterDelete.csv” file

8) getAllTransaction (paging)

GET /transaction/all getAllTransaction

Parameters Cancel

Name	Description
pageNumber * required integer(\$int32) (query)	pageNumber <input type="text" value="3"/>
pageSize * required integer(\$int32) (query)	pageSize <input type="text" value="6"/>

Execute Clear

Responses Response content type */*

Curl

```
curl -X GET "https://localhost:8080/transaction/all?pageNumber=3&pageSize=6" -H "accept: */*"
```

Request URL

```
https://localhost:8080/transaction/all?pageNumber=3&pageSize=6
```

Server response

Code	Details
200	<p>Response body</p> <pre>{ "message": "Transaction Found", "status": 200, "httpStatus": "200 OK", "success": true, "exception": false, "body": [] }</pre> Download <p>Response headers</p> <pre>connection: keep-alive content-type: application/json date: Wed, 01 Dec 2021 10:41:03 GMT keep-alive: timeout=60 transfer-encoding: Identity</pre>

Responses

Code	Description
200	<p>OK</p> <p>Example ValueModel</p> <pre>{ "body": {}, "exception": true, "httpStatus": "string", "message": "string" }</pre>

9) findTransactionByDate

GET /transaction/date/{date} findTransactionByDate

Cancel

Parameters

Name	Description
date * required string(\$date) (query)	date 2021-12-02

Execute Clear

Responses Response content type */*

Curl

```
curl -X GET "https://localhost:8080/transaction/date/{date}?date=2021-12-02" -H "accept: */*"
```

Request URL

```
https://localhost:8080/transaction/date/{date}?date=2021-12-02
```

Server response

Code	Details
200	Response body
	<pre>{ "transactionId": 0, "fromAccount": 0, "toAccount": 0, "fromAccountName": "A", "toAccountName": "B", "sameBankTransaction": true, "otherBank": "string", "amount": 0, "date": "2021-12-02", "time": "2021-12-02T06:36:06.407", "transactionType": "credit", "status": "string" }, { }</pre>

10) findTransactionByDateAndTransactionType

GET /transaction/{transactionType}/{date} findTransactionByDateAndTransactionType

Parameters

Name	Description
date * required string(\$date) (query)	date 2021-12-02
transactionType * required string (path)	transactionType credit

Execute **Clear**

Responses Response content type */*

Curl

```
curl -X GET "https://localhost:8080/transaction/credit/{date}?date=2021-12-02" -H "accept: */*"
```

Request URL

```
https://localhost:8080/transaction/credit/{date}?date=2021-12-02
```

Server response

Code	Details
200	<p>Response body</p> <pre>"httpStatus": "200 OK", "success": true, "exception": false, "body": [{ "transactionId": 0, "fromAccount": 0, "toAccount": 0, "fromAccountName": "A", "toAccountName": "B", "sameBankTransaction": true, "otherBank": "string", "amount": 0, "date": "2021-12-02", "time": "2021-12-02T06:06:40Z", "transactionType": "credit", "status": "string" },</pre> <p>Download</p> <p>Response headers</p> <pre>connection: keep-alive content-type: application/json date: Thu, 02 Dec 2021 07:32:19 GMT keep-alive: timeout=60 transfer-encoding: Identity</pre>

Responses

Code	Description
------	-------------

4) Appendix

Thank to everyone gave me this chance to try Spring Boot platform and increase my knowledge.

I faced some problems with the pgAdmin 4 to I create the table more than once so you may notice the deference between data!