

This project is using java 8 and maven 3.8.4

What are included in the project?

Local service/api to mockup the account validation data sources.

URI1: <http://localhost:8080/v1/api/account/validate>

Postman interface showing a POST request to `http://localhost:8080/v1/api/account/validate`. The request body is a JSON object: `{\"accountNumber\": \"12345678\"}`. The response status is `200 OK`, and the response body is a JSON object: `{\"isValid\": false}`.

URI2: <http://localhost:8080/v2/api/account/validate>

Postman interface showing a POST request to `http://localhost:8080/v2/api/account/validate`. The request body is a JSON object: `{\"accountNumber\": \"12345678\"}`. The response status is `200 OK`, and the response body is a JSON object: `{\"isValid\": true}`.

REST API for validating account

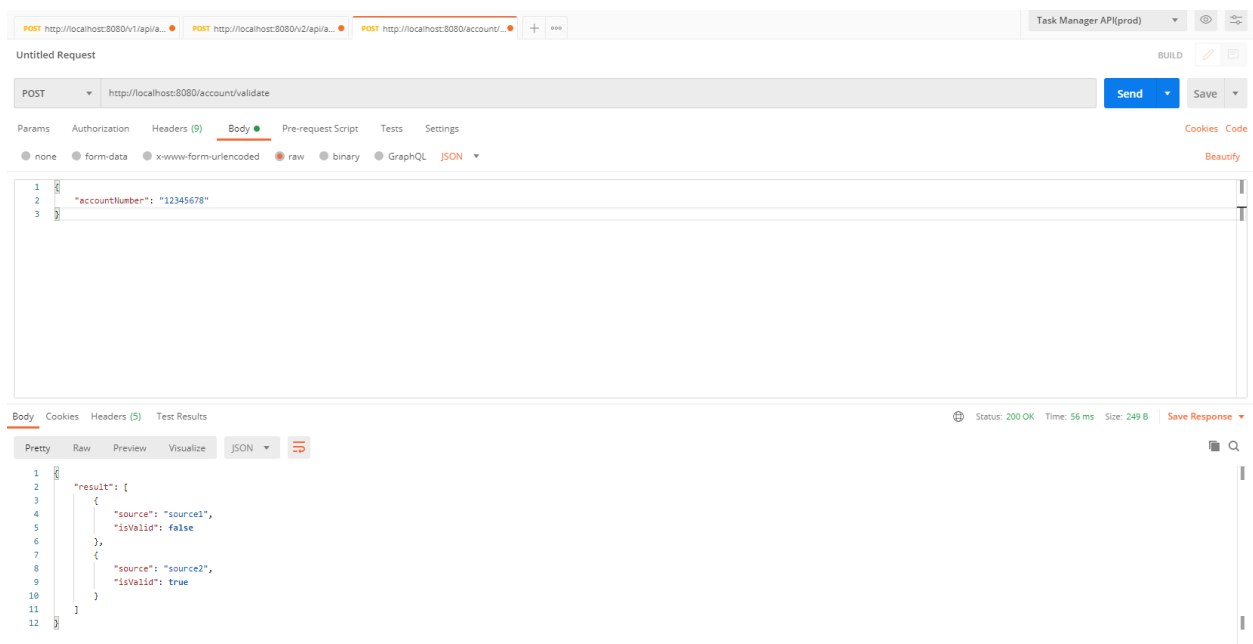
Data sources uri have been set in file **application.properties**. In this demo, it's set to the local data source services. We can changes it to a really uri accordingly.

```
AccountValidateService.java AccountValidateResponse.java AccountController.java application.properties x
1 #server.servlet.context-path=/api/homework
2
3 # data source uri config
4 source1=http://localhost:8080/v1/api/account/validate
5 source2=http://localhost:8080/v2/api/account/validate
6
```

Account validation URI: <http://localhost:8080/account/validate>

Tested in Postman.

When there is no source in request, the response includes the validations from all sources.



When there are sources in request, the response includes the validations from the sources in request.

The screenshot shows the Postman interface for a POST request to `http://localhost:8080/account/validate`. The request body is a JSON object with an `accountNumber` of `"12345678"` and a single `source` of `"source1"`. The response status is 200 OK, and the response body is a JSON object with a `result` array containing one object: `{ "source": "source1", "isValid": false }`.

```
1 {
2   "accountNumber": "12345678",
3   "sources": [
4     "source1"
5   ]
6 }
```

```
1 {
2   "result": [
3     {
4       "source": "source1",
5       "isValid": false
6     }
7   ]
8 }
```

When there are invalid sources in request, they would be ignored.

The screenshot shows the Postman interface for a POST request to `http://localhost:8080/account/validate`. The request body is a JSON object with an `accountNumber` of `"12345678"` and three `sources`: `"source1"`, `"source2"`, and `"source3"`. The response status is 200 OK, and the response body is a JSON object with a `result` array containing two objects: `{ "source": "source1", "isValid": false }` and `{ "source": "source2", "isValid": true }`. The `source3` is not present in the response, indicating it was ignored.

```
1 {
2   "accountNumber": "12345678",
3   "sources": [
4     "source1",
5     "source2",
6     "source3"
7   ]
8 }
```

```
1 {
2   "result": [
3     {
4       "source": "source1",
5       "isValid": false
6     },
7     {
8       "source": "source2",
9       "isValid": true
10    }
11  ]
12 }
```

REST API url could be set and changed in **application.properties** file. As the mockup services, I commented the configuration out.

```
AccountValidateService.java AccountValidateResponse.java AccountController.java application.properties x
1 #server.servlet.context-path=/api/homework
2 |
```

Request timeout has been set to 2s in **application.properties** file. Add `Thread.sleep` (it's commented out in source codes) to generate the default timeout exception.

```
AccountValidateService.java AccountValidateResponse.java AccountController.java application.properties x
1 #server.servlet.context-path=/api/homework
2
3 # data source uri config
4 source1=http://localhost:8080/v1/api/account/validate
5 source2=http://localhost:8080/v2/api/account/validate
6
7 # rest api response time limit 2s
8 spring.mvc.async.request-timeout=2000
```

POST http://localhost:8080/v1/api/a... POST http://localhost:8080/v2/api/a... POST http://localhost:8080/account/... Task Manager API(prod)

Untitled Request BUILD

POST http://localhost:8080/account/validate Send Save

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2   "accountNumber": "12345678",
3   "sources": [
4     "source1",
5     "source2"
6   ]
7 }
```

Body Cookies Headers (4) Test Results Status: 500 Internal Server Error Time: 2.75 s Size: 1.03 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "timestamp": "2022-02-06T20:32:15.351+00:00",
3   "status": 500,
4   "error": "Internal Server Error",
5   "trace": "java.lang.InterruptedException: sleep interrupted\\n\\tat java.base/java.lang.Thread.sleep(Native Method)\\n\\tat com.example.demo.account.AccountController.lambda$0(AccountController.java:26)\\n\\tat org.springframework.web.context.request.async.WebAsyncManager.lambda$startCallableProcessing$4(WebAsyncManager.java:337)\\n\\tat java.base/java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:539)\\n\\tat java.base/java.util.concurrent.FutureTask.run(FutureTask.java:264)\\n\\tat java.base/java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1136)\\n\\tat java.base/java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:635)\\n\\tat java.base/java.lang.Thread.run(Thread.java:833)\\n\\n",
6   "message": "sleep interrupted",
7   "path": "/account/validate"
8 }
```