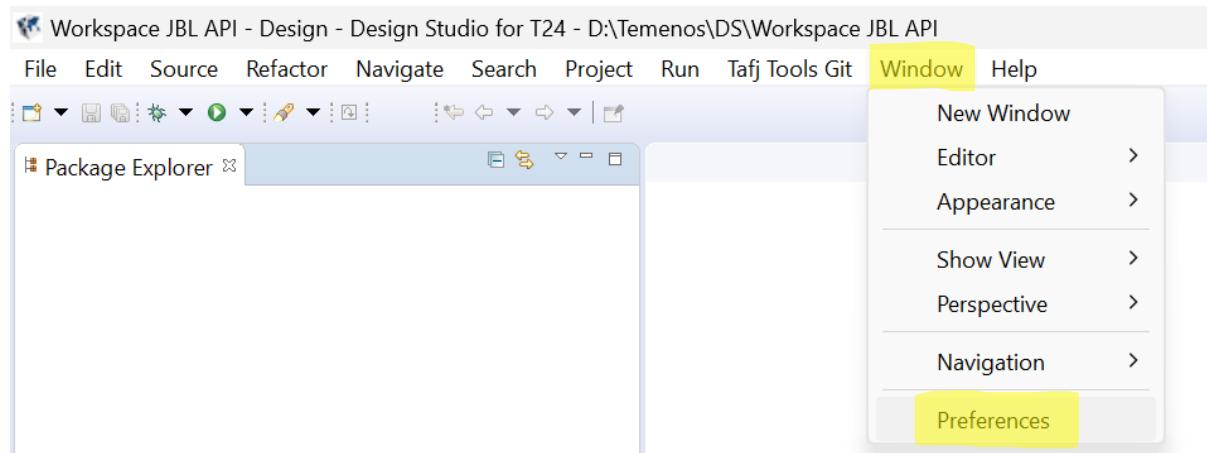
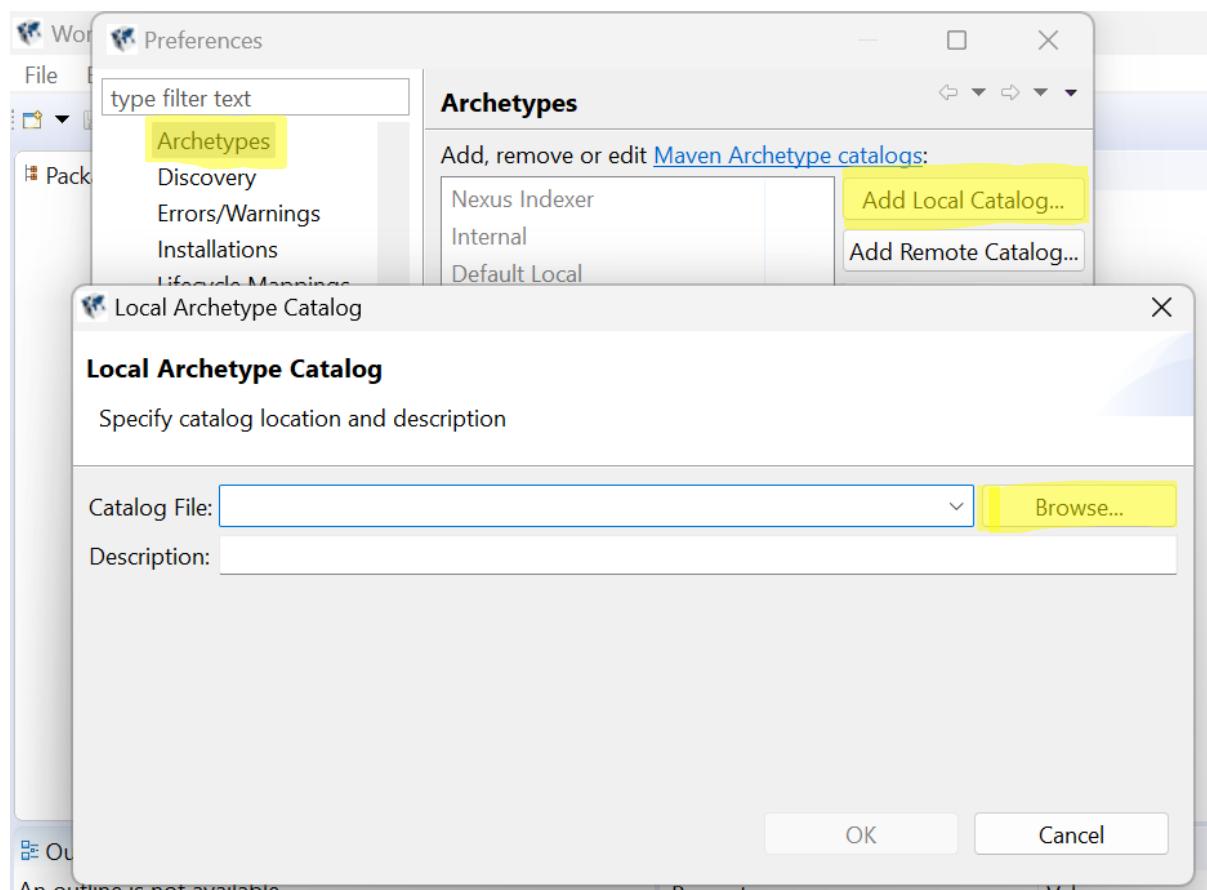


IRIS API

Adding Archetype in Design Studio:

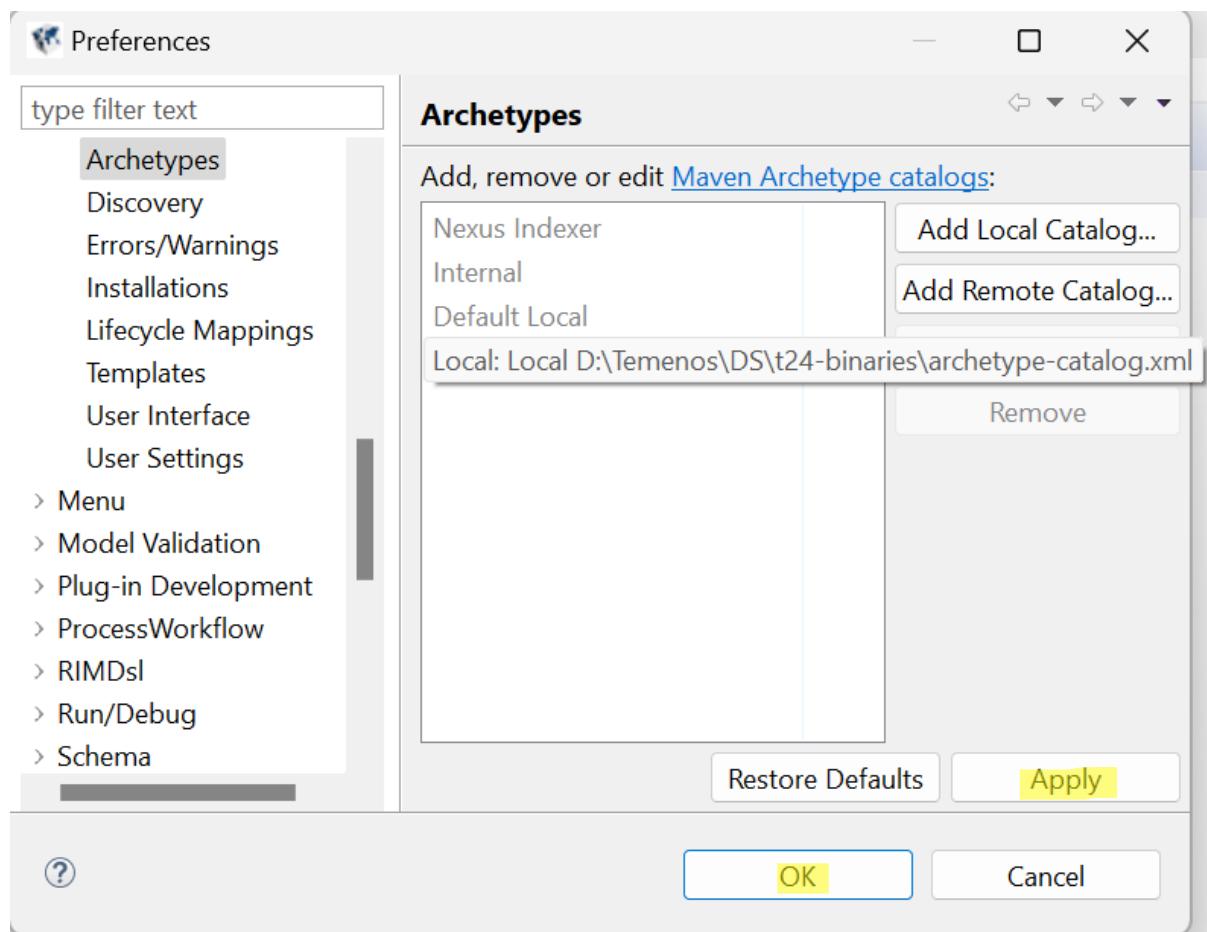


Window > Preferences > Archetypes > Add Local Catalog

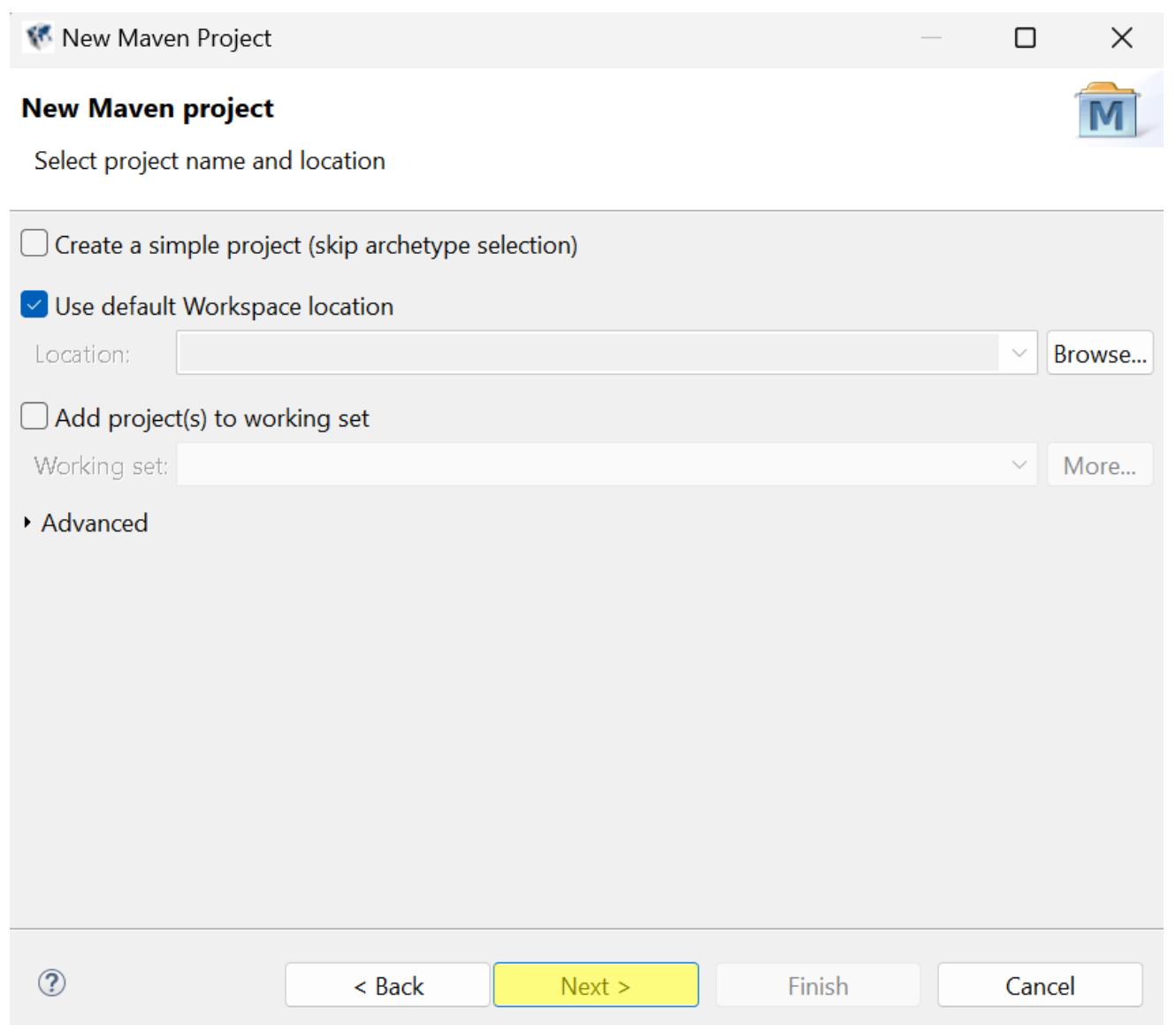
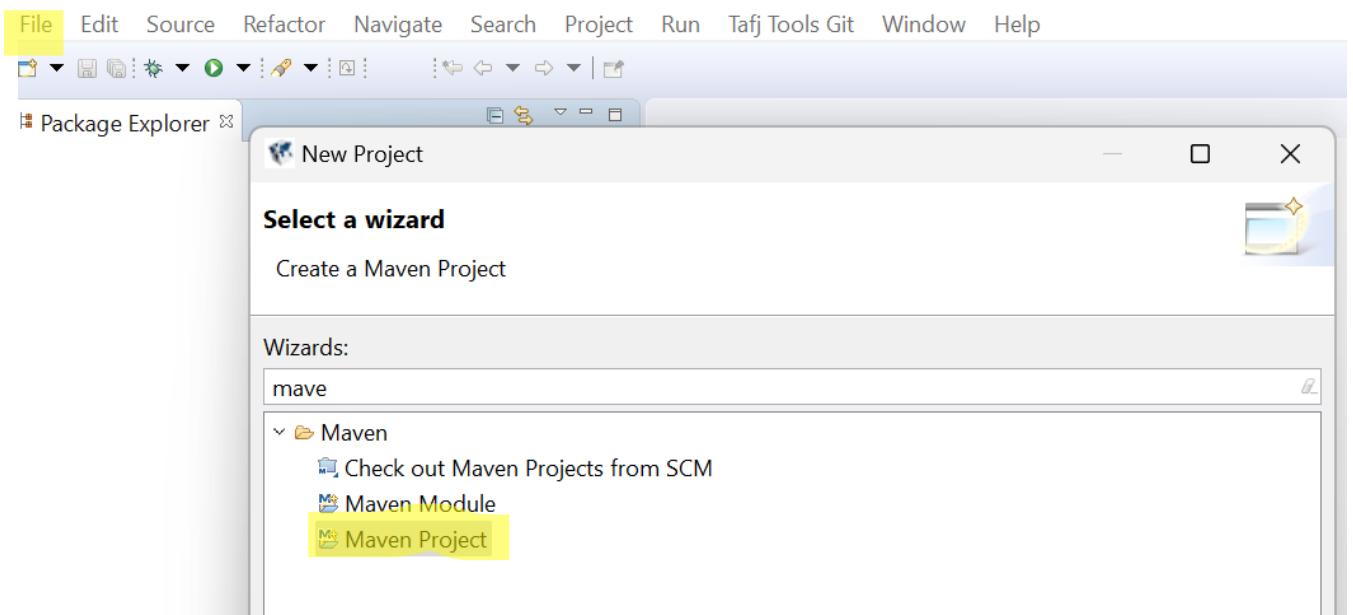


T24-binaries > D:\Temenos\DS\t24-binaries -- archetype-catalog.xml

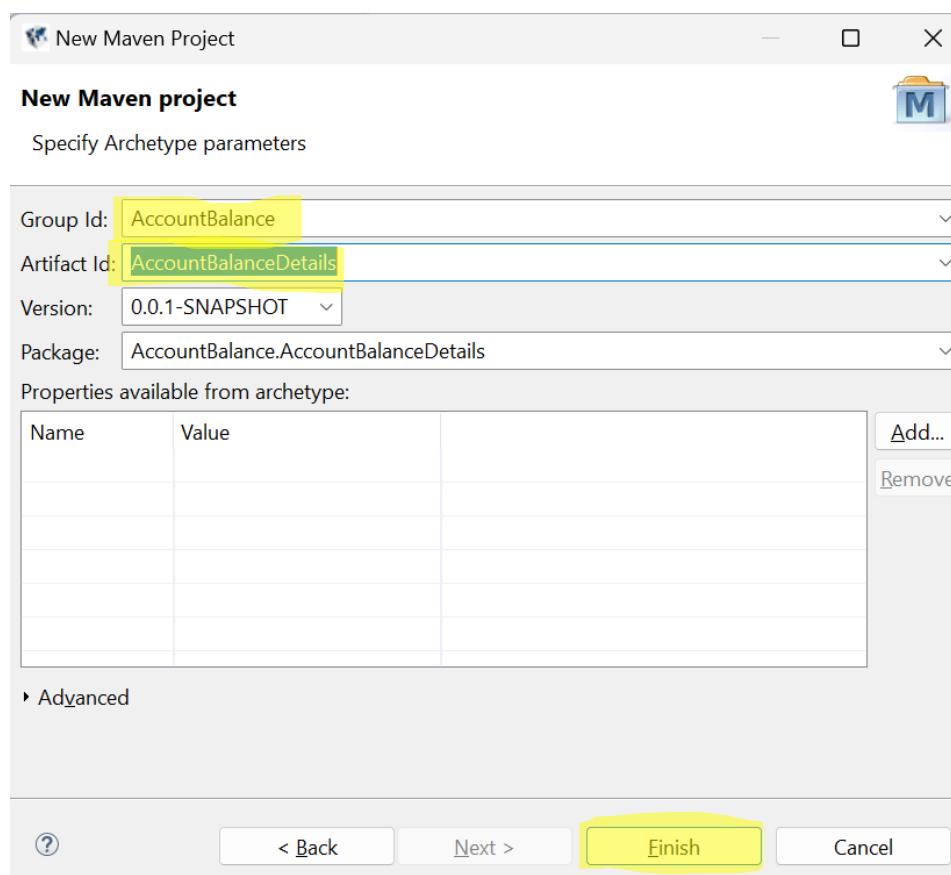
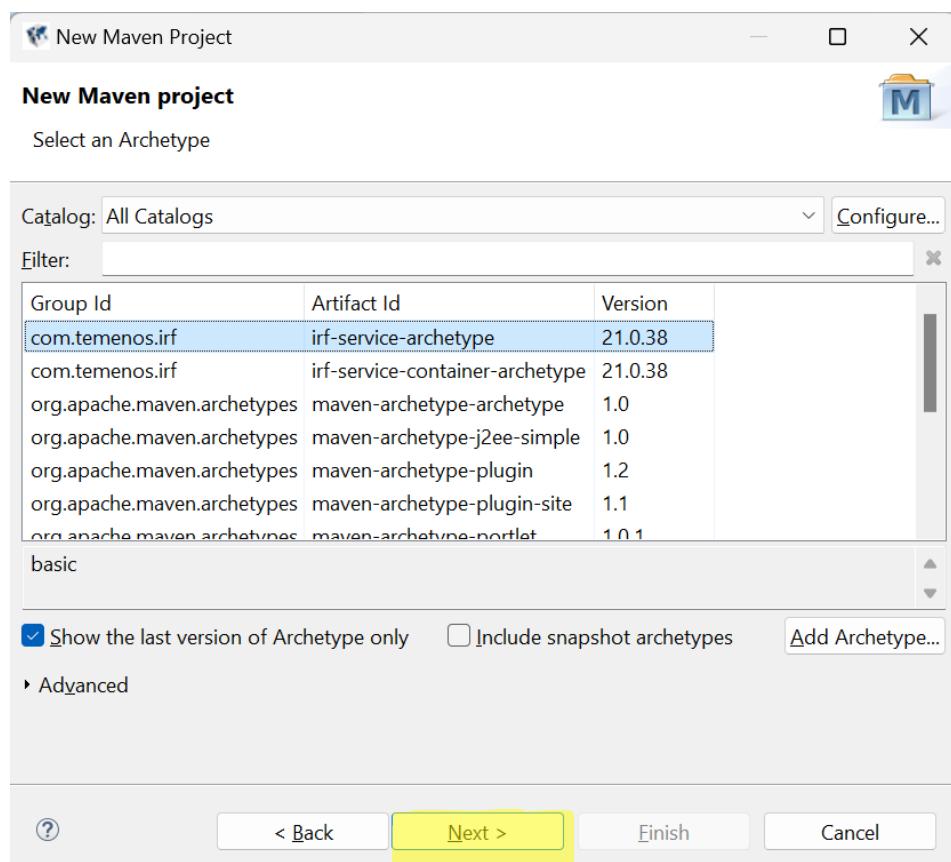
This PC > Local Disk (D:) > Temenos > DS > t24-binaries >				
Folder				
	Name	Date modified	Type	Size
	xalan	27/12/2023 06:57	File folder	
	xerces	27/12/2023 06:57	File folder	
	xml-apis	27/12/2023 06:57	File folder	
	xmlpull	27/12/2023 06:57	File folder	
T	xml-resolver	27/12/2023 06:57	File folder	
X	xmlunit	27/12/2023 06:57	File folder	
K	xom	27/12/2023 06:57	File folder	
4	xpp3	27/12/2023 06:57	File folder	
	archetype-catalog.xml	27/12/2023 06:57	xmlfile	1 KB



New Mavel Project from DS:



There must be two **Maven** Projects (irf-service-archetype & irf-service-container -archetype)



The screenshot shows the 'New Maven Project' dialog box with the title 'New Maven project'. The main area is titled 'Select an Archetype' and contains a table of available archetypes. The table has columns for Group Id, Artifact Id, and Version. A filter input field is above the table. At the bottom, there are checkboxes for 'Show the last version of Archetype only' and 'Include snapshot archetypes', along with a 'Add Archetype...' button. Navigation buttons at the bottom include '?', '< Back', 'Next >', 'Finish', and 'Cancel'.

Group Id	Artifact Id	Version
com.temenos.irf	irf-service-archetype	21.0.38
com.temenos.irf	irf-service-container-archetype	21.0.38
org.apache.maven.archetypes	maven-archetype-archetype	1.0
org.apache.maven.archetypes	maven-archetype-j2ee-simple	1.0
org.apache.maven.archetypes	maven-archetype-plugin	1.2
org.apache.maven.archetypes	maven-archetype-plugin-site	1.1
org.apache.maven.archetypes	maven-archetype-web-portlet	1.0.1

Show the last version of Archetype only Include snapshot archetypes Add Archetype...

< Back **Next >** Finish Cancel

New Maven project

New Maven project

Specify Archetype parameters

Group Id: AccountBalance

Artifact Id: AccountBalanceContainer

Version: 0.0.1-SNAPSHOT

Package: AccountBalance.AccountBalanceContainer

Properties available from archetype:

Name	Value	
		Add...
		Remove

Advanced

?

< Back

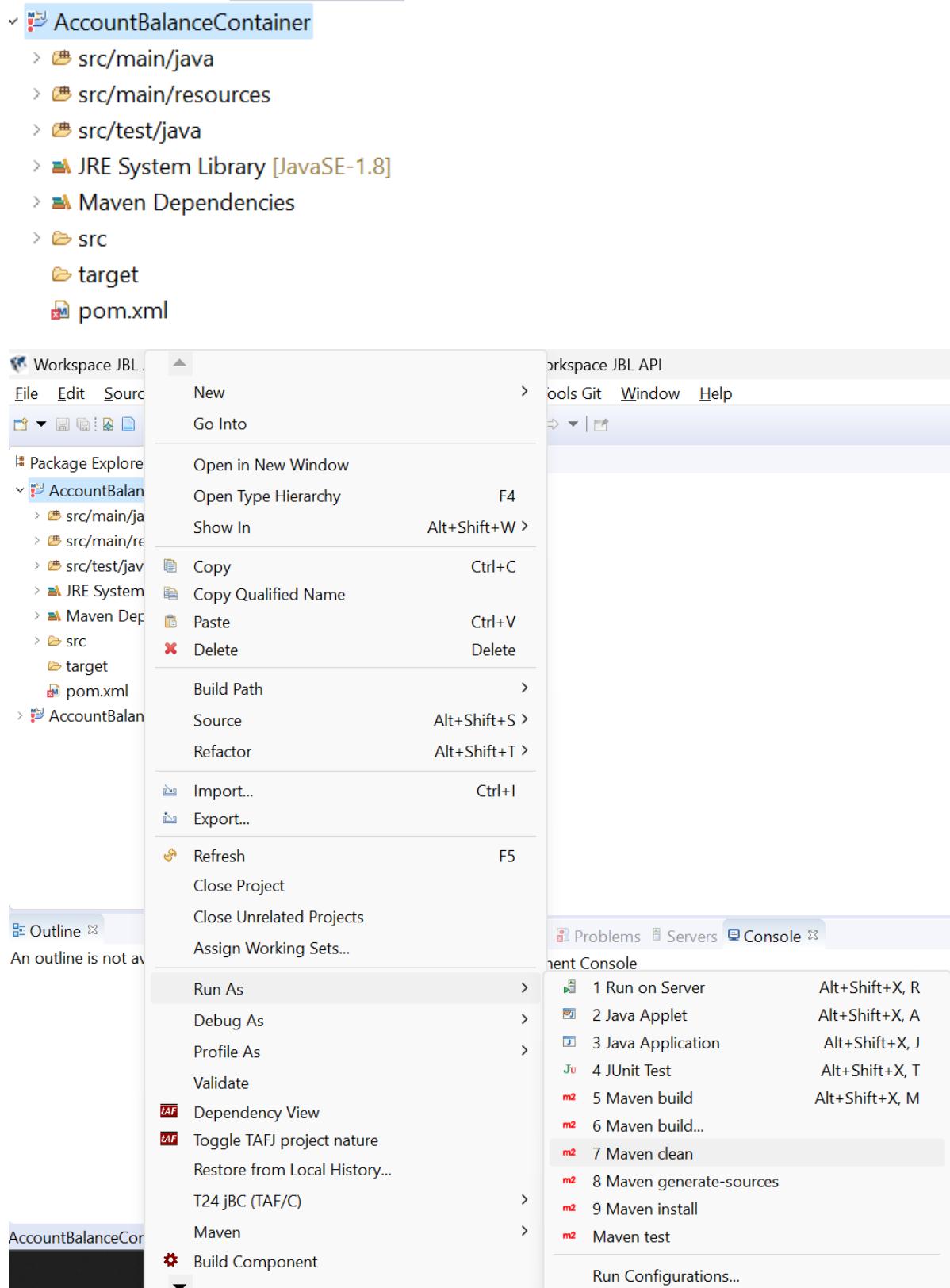
Next >

Finish

Cancel

Clean the both container and Service project by Selecting,

Right click on Project → Run As → Maven Clean.



```

Properties Problems Servers Console
<terminated> D:\Temenos\DS\jdk\bin\javaw.exe (Jan 29, 2024, 10:53:08 AM)
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building AccountBalanceContainer 21.0.38
[INFO]
[INFO] -----
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ AccountBalanceContainer ---
[INFO] Deleting D:\Temenos\DS\Workspace JBL API\AccountBalanceContainer\target
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO]
[INFO] -----
[INFO] Total time: 0.278 s
[INFO] Finished at: 2024-01-29T10:53:09+06:00
[INFO] Final Memory: 15M/487M
[TNF01] -----

```

```

Properties Problems Servers Console
<terminated> D:\Temenos\DS\jdk\bin\javaw.exe (Jan 29, 2024, 10:54:06 AM)
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building AccountBalanceDetails 0.0.1-SNAPSHOT
[INFO]
[INFO] -----
[INFO] --- maven-clean-plugin:3.0.0:clean (default-clean) @ AccountBalanceDetails ---
[INFO] Deleting D:\Temenos\DS\Workspace JBL API\AccountBalanceDetails\target
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO]
[INFO] -----
[INFO] Total time: 0.353 s
[INFO] Finished at: 2024-01-29T10:54:07+06:00
[INFO] Final Memory: 16M/487M
[INFO] -----

```

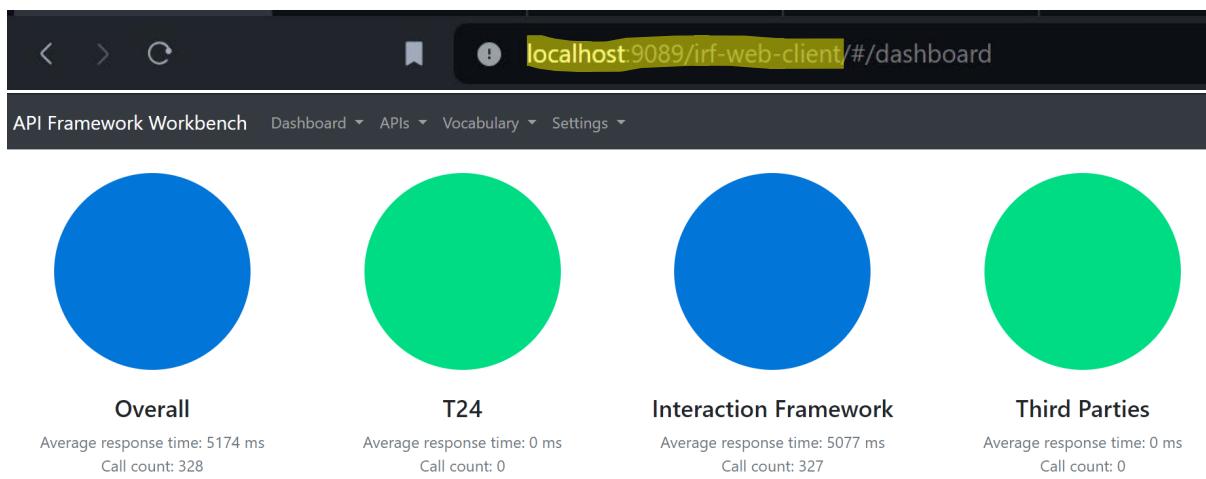
IRIS 2.0 war File Generation

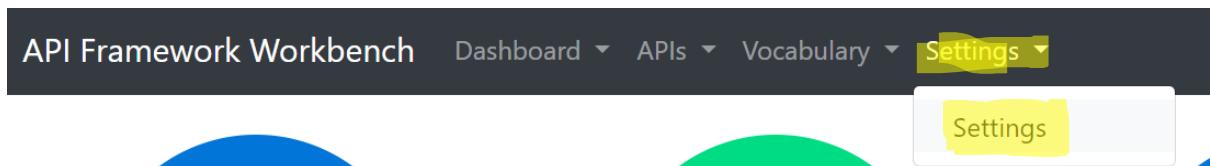
Deploy the 2 war files which is given by temenos in your JBoss

irf-web-client - Workbench

irf-test-web - To test the IRIS Services

Give the base URL in the browser-> Workbench gets opened...





Select settings tab → click settings → Then the servers page gets opened.

Servers

Servers		
Active	Name	URL
<input checked="" type="radio"/>	TestWeb Server	http://localhost:9089/irf-provider-container
<input type="radio"/>	Default Server	http://127.0.0.1:8080

[Add Server](#)

Amend Server

 Name:

 URL:

API Creation from Workbench

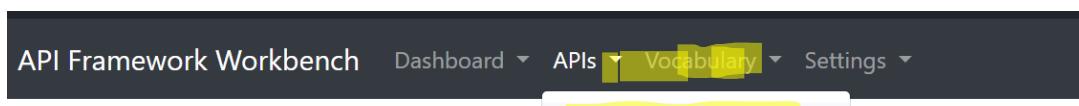
API Creation:

First We should create a Version or Enquiry in below format.

Product.API.verb.version -----EX: ST.API.CUSTOMERS.1.0.0

go to localhost:9089/irf-web-client

Click API>create provider API.



Create Provider API

[Create Provider API](#)

The Available Artefacts can be displayed. The Artefacts can be Versions(Screen), Enquiries(Query),AA Product(Product) and so on.

Select the data's which you want to import.

The API definition page will be displayed.

Select the required artefacts from available artefacts.

Available Artefacts

Product 140 Query 917

Screen 470 Evidence 9

Account

message

getAccountBalance AC.API.WS.ACCOUNT.BALANCES.1.0.0 (AC)

getAccountBalance PZ.API.BERLIN.1.3.ACCOUNTS.BALANCE.1.1.0 (PZ)

getAccountBalance PZ.API.LISTET.1.4.ACCOUNTS.BALANCE.1.0.0 (PZ)

getAccountBalanceMovement

API Definition

Add artefacts to the API definition by choosing from the available artefacts.
OR
Select the json file of an existing API inventory to modify an existing definition

No file chosen

Title	GetCusAcBal	Key	jbl-api
Description	GetCusAcBalance		
Version	v1.0.0	Schemes	<input checked="" type="checkbox"/> https <input type="checkbox"/> ftp
Base Path	/api	Host	localhost:9089

Provide key , url, summary, description. (**give request and response payload versions If it is AA API**)

AC.API.WS.ACCOUNT.BALANCES.1.0.0

Main Parameters		URL Parameters
Operation	getAccountBalance	* Name Data Type
Description		<input type="checkbox"/> CustomerId string
Summary		<input type="checkbox"/> customerId string
Tags	<input type="checkbox"/> CORPORATE <input type="checkbox"/> INCLUSIVE BANKING <input type="checkbox"/> ISLAMIC BANKING <input type="checkbox"/> RETAIL <input type="checkbox"/> TREASURY <input type="checkbox"/> WEALTH	
Domain	party	
HTTP Method	GET	
Operation security	Public	
Client Type	INTERNAL	
<input type="checkbox"/> Bulk	<input type="checkbox"/> Requires consent	<input type="checkbox"/> Deprecated
ViewResponseFields		

Selection Mapping

Selection Field	Operand	Map Type	Map To
customerId	equals	<input checked="" type="radio"/> Parameter <input type="radio"/> Constant <input type="radio"/> None	CustomerId

click **NEXT** on top of the screen>**FINISH**

API Definition

Title	GetCusAcBal	Key	jbl-api
Description	GetCusAcBalance		
Version	v1.0.0	Schemes	https
Base Path	/api	Host	localhost:9089

API Endpoints

Path	/ws/{CustomerId}
HTTP Method	GET
Operation	getAccountBalance
Operation Security	Public
Target	AC.API.WS.ACCOUNT.BALANCES.1.0.0
Target Type	Query

2. Review

3. Download

Generating API artefacts

Process complete
Click the link to download the generated artefacts

Download

IRIS Workbench will generate a zip based on selected transact artefacts. Download it. That zip contains 3 folders. i.e. **api-docs**, **inventory**, **services**

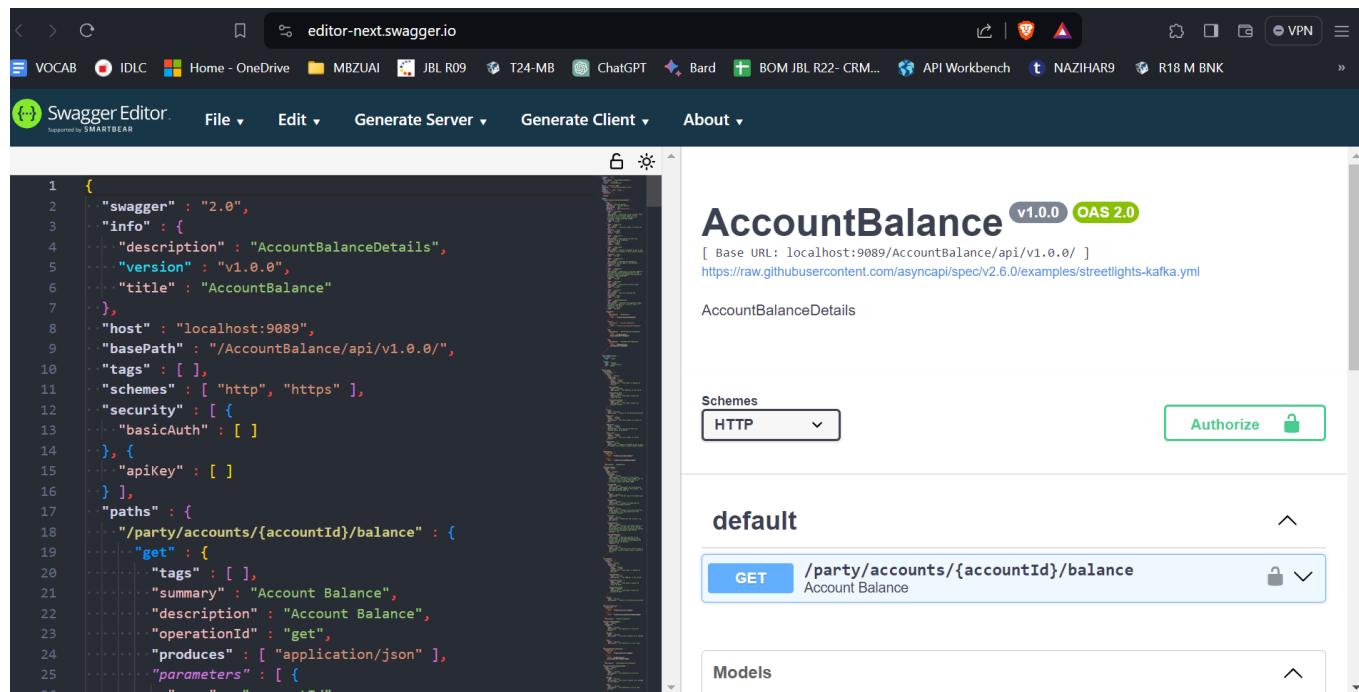
After creating a **Maven** project (**AccountBalanceDetails**),

Delete all files in `src/main/resources`

 api-docs	File folder
 inventory	File folder
 services	File folder
 src	File folder

api-docs – This folder contains swagger which will show the specification of our API

Ex:



The screenshot shows the Swagger Editor interface with the following details:

- Title:** AccountBalance **v1.0.0 OAS 2.0**
- Base URL:** localhost:9089/AccountBalance/api/v1.0.0/
- Spec URL:** https://raw.githubusercontent.com/asyncapi/spec/v2.6.0/examples/streetlights-kafka.yml
- Schemes:** HTTP
- Authorize:** A green button with a lock icon.
- default:** A section containing a single API endpoint:
 - Method:** GET
 - Path:** /party/accounts/{accountId}/balance
 - Description:** Account Balance
- Models:** A section showing various data models used in the API.

On the left side, there is a code editor displaying the generated `swagger.json` file for the `AccountBalance` API.

Inventory – This folder contains details of the generated artefacts/APIs. Which is useful to load existing APIs to the workbench rather than selecting artefacts from scratch.

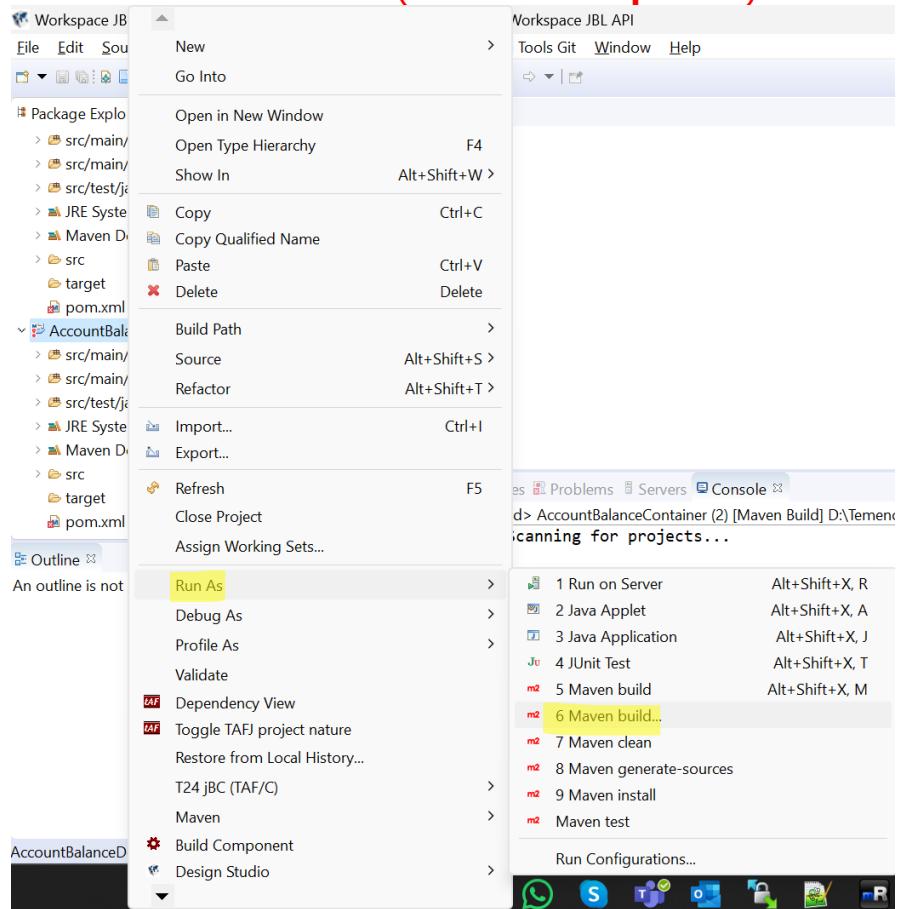
Services- This folder contains the xml routes for routing the API to appropriate transact artefact.

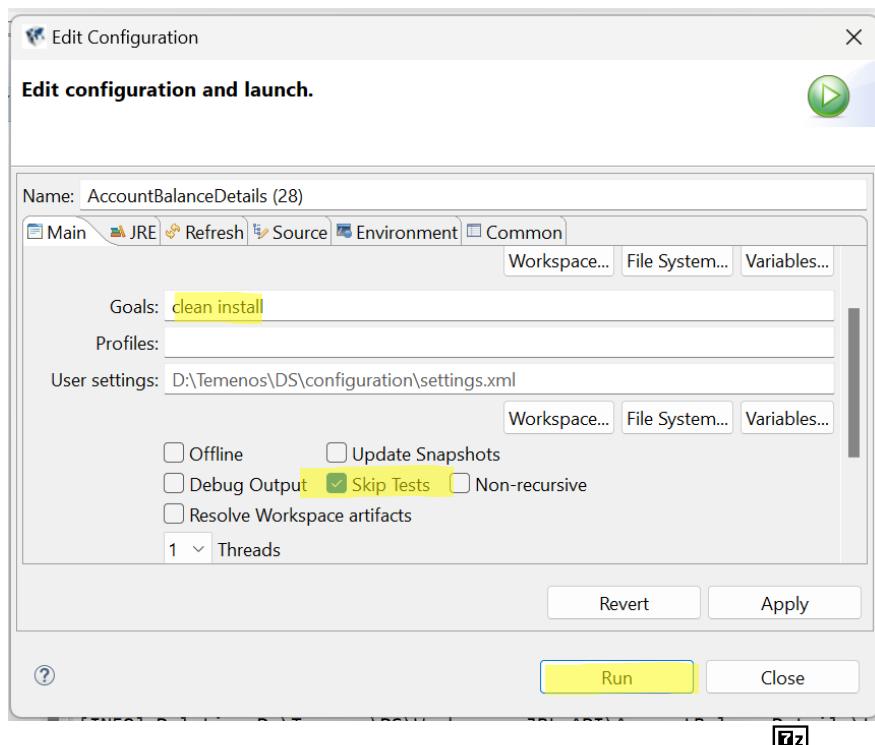
This file alone is fine to deploy in a container to get our API work.

Paste the **Downloaded** files from the **Workbench** in the Below path,
D:\Temenos\DS\Workspace JBL API\AccountBalanceDetails\src\main\resources

D:\Temenos\DS\Workspace JBL API\AccountBalanceDetails\src\main\resources				
	Name	Date modified	Type	Size
▶	api-docs	28/01/2024 20:18	File folder	
▶	inventory	28/01/2024 20:17	File folder	
▶	services	28/01/2024 20:17	File folder	
▶	src	28/01/2024 20:17	File folder	

Right click on the main project(**AccountBalanceDetails**)>Run as> **Maven build**...>clean install> run (**Musk check Skip Tests**)





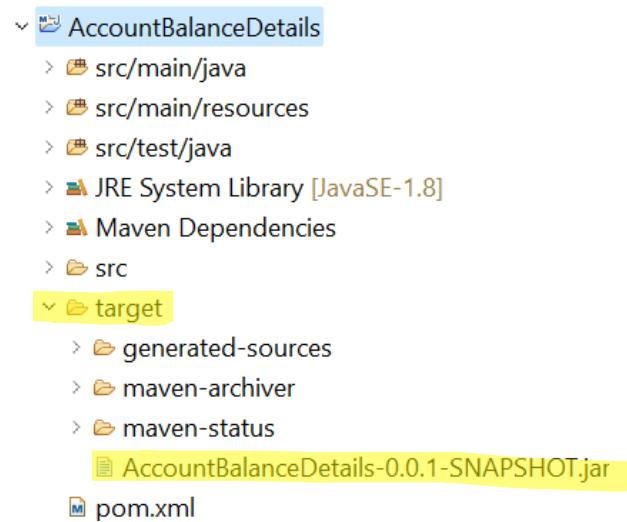
If the BUILD is FAILED then need to download the .jar file and paste it inside the **t24 binaries** directory like this,
D:\Temenos\DS\t24-binaries\org\json\json\20170516



archetype-catalog.xml

We also need to replace each **http** to **https** in **archetype-catalog.xml** file.

If the **BUILD** is **SUCCESSFUL** then the **jar** file will be created here.



Go to the **container** project >pom.xml

Go to **dependencies** tab and add your jars (created in main project)

Go to **pom.xml** tab

buildforJMS > true ; **Not Required/Optional** for Model Bank.

```

<profile>
    <id>buildForJMS</id>
    <activation>
        <activeByDefault>true</activeByDefault>
    </activation>
    <build>
        <finalName>irisR18</finalName>
        <plugins>
            <plugin>
                <artifactId>maven-resources-plugin</artifactId>
                <version>${maven.resources.plugin}</version>
                <executions>
                    <execution>
                        <id>copywebresource</id>
                    ...

```

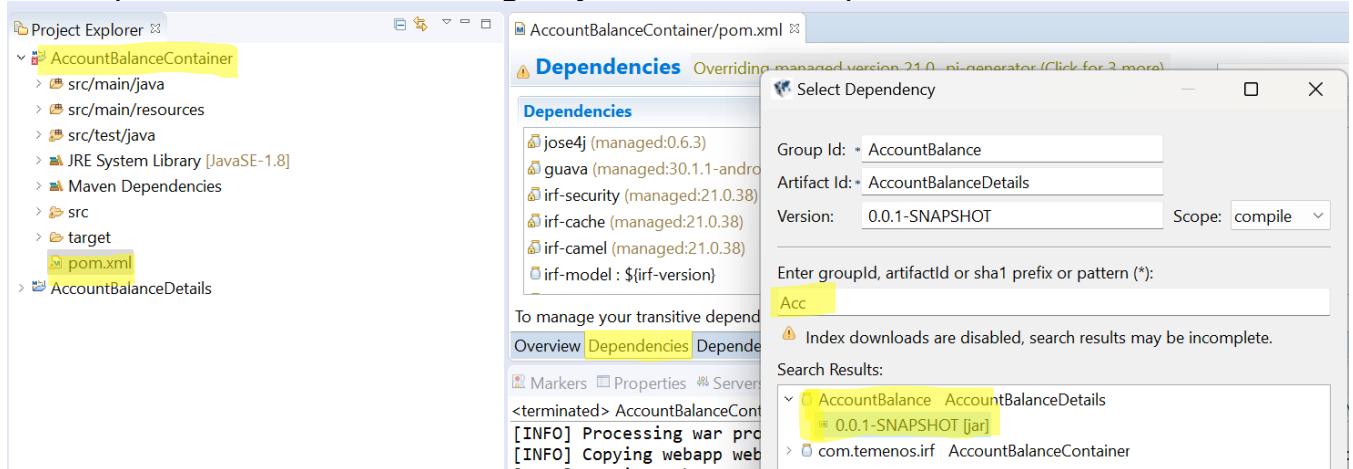
Adding Dependencies:

Special Note: We must build the main project first then the container project.

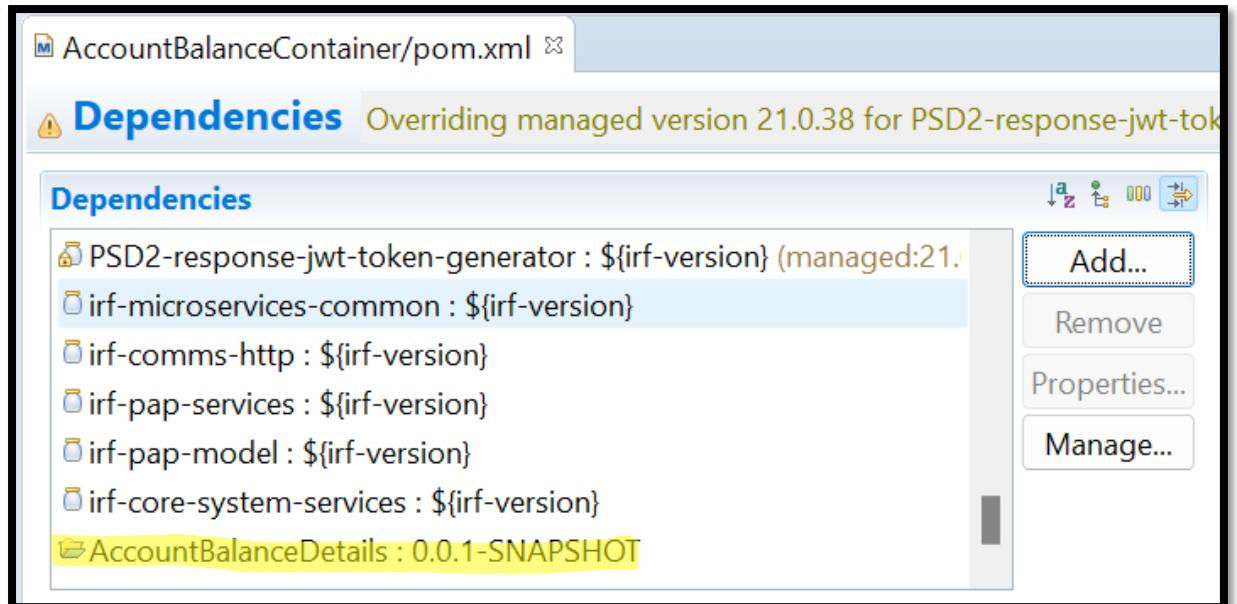
Double click on pom.xml → select Dependencies tab → click Add.

The **container** will have the main project **jar file** as a Dependency.

Search your service name in Enter **groupId** field → select your service → Click **OK**.



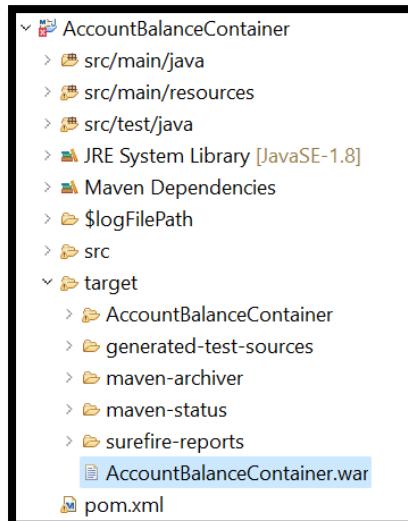
Then **Save** it. → Ctrl +S



If the jboss is a remote server then,
 Provide the jboss environment: username and password.
`Java.naming.security.principal = username`
`Java.naming.security.credentials = Password`

```
<terminated> AccountBalanceContainer (9) [Maven Build] D:\Temenos\DS\jdk\bin\javaw.exe (Jan 29, 2024, 4:23:27 PM)
[INFO] Processing war project
[INFO] Copying webapp webResources [D:\Temenos\DS\Workspace JBL API\AccountBalanceContainer\src\main\webapp\jbossenabled]
[INFO] Copying webapp resources [D:\Temenos\DS\Workspace JBL API\AccountBalanceContainer\src\main\webapp]
[INFO] Webapp assembled in [786 msecs]
[INFO] Building war: D:\Temenos\DS\Workspace JBL API\AccountBalanceContainer\target\AccountBalanceContainer.war
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ AccountBalanceContainer ---
[INFO] Installing D:\Temenos\DS\Workspace JBL API\AccountBalanceContainer\target\AccountBalanceContainer.war to D:\Temenos\DS\t24-binaries\com\temenos\AccountBalanceContainer\AccountBalanceContainer.war
[INFO] Installing D:\Temenos\DS\Workspace JBL API\AccountBalanceContainer\pom.xml to D:\Temenos\DS\t24-binaries\com\temenos\AccountBalanceContainer\pom.xml
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 25.663 s
[INFO] Finished at: 2024-01-29T16:23:54+06:00
[INFO] Final Memory: 79M/661M
[INFO] -----
```

D:\Temenos\DS\Workspace JBL API\AccountBalanceContainer\target



To Enable/disable xacml:

Comment out below line to disable xacml. (OR) **Uncomment** to enable xacml.

```

<!-- enable below two beans for microservice based metric usage strategy -->
<!-- <bean id="metricflushstrategy" class="com.temenos.irf.core.metrics.MicroserviceSimpleCountFlushStrategy" />
<!-- Bean for XACML -->
<bean id="irisProcessorAspect" class="com.temenos.irf.core.xacml.authz.IrisProcessorXacmlAspect" />
<import resource="classpath*:services/**/*-service-v*.xml" />

```

To Enable/disable JWT:

Comment out below set of lines to **disable** JWT (OR)

UnComment below set of lines to **enable** JWT.

```

<!-- Uncomment below filters to enable JWT based token validation
<filter>
    <filter-name>springSecurityFilterChain</filter-name>
    <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>
</filter>

<filter-mapping>
    <filter-name>springSecurityFilterChain</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping> -->

```

If **jms.properties** is required to change from the war file,

Name	Size	Packed Si...	Modified	Create
xacml	3 653	1 121	2024-01-29 16:35	
irf-config	27 041	7 729	2024-01-29 16:35	
spring-saml-iris-authenticator.xml	2 050	681	2024-01-29 16:35	
spring-jwt-iris-authenticator.xml	8 503	1 784	2024-01-29 16:35	
spring-jwt-context.properties	484	241	2024-01-29 16:24	
log4j2.properties	2 437	683	2024-01-29 16:35	
jms.properties	1 895	816	2024-01-29 16:24	
applicationContext.xml	4 562	1 152	2024-01-29 16:24	

1 / 8 object(s) selected 1 895 1 895 2024-01-29 16:24:32

Deployment :

Deploy the .war file in your jboss. **D:\Temenos\jboss\standalone\deployments**

Name Date modified Type Size			
AccountBalanceContainer.war	29/01/2024 16:35	WAR File	141,671 KB
AccountBalanceContainer.war.deployed	29/01/2024 16:35	DEPLOYED File	1 KB

Adding server to view the API :

Open workbench → click on Settings tab → click Add Server button.

The screenshot shows the 'API Framework Workbench' interface. On the left, there is a 'Servers' list with three entries: 'TestWeb Server', 'Default Server', and 'AccBalDetails'. The 'AccBalDetails' entry is selected. On the right, a modal dialog titled 'Amend Server' is open, showing the 'Name' field set to 'AccBalDetails' and the 'URL' field set to 'http://localhost:9089/AccountBalanceContainer'. There are 'Cancel' and 'Amend' buttons at the bottom of the dialog.

Checking the Deployed Services

The screenshot shows the 'Dashboard' section of the API Framework Workbench. It displays various service counts: 1 Domains, 3 Endpoints, 4 Services, and 3 APIs. Below this, there is a section labeled 'other' with a cloud icon.

CLICK - API docs

The screenshot shows the API documentation for 'Acb-v1.0.0-swagger v1.0.0'. It features a 'party-accounts-Acb' endpoint with a single GET method: 'GET /v1.0.0/party/accounts/{accountId}/balance'. A 'API docs' button is located in the top right corner of the documentation panel.

Swagger
Supported by SMARTBEAR

http://localhost:9089/AccountBalanceContainer/api/v1.0.0/meta/apidocs/Acb-v1.0.0

Explore

AccountBalance v1.0.0

[Base URL: localhost:9089/AccountBalance/api/v1.0.0 /]
http://localhost:9089/AccountBalanceContainer/api/v1.0.0/meta/apidocs/Acb-v1.0.0

AccountBalanceDetails

Schemes: HTTP Authorize

default

GET /party/accounts/{accountId}/balance Account Balance

default

GET /party/accounts/{accountId}/balance Account Balance

Account Balance

Parameters

Try it out

GET /party/accounts/{accountId}/balance Account Balance

Account Balance

Parameters

Name Description

accountId * required string (path) Identifier of the account. Often referred to as the account number, yet for consistency this is always referred to as accountId. Accepts both IBAN & BBAN

120456

Execute Cancel Clear

Responses

Response content type: application/json

Curl

```
curl -X GET "http://localhost:9089/AccountBalance/api/v1.0.0/party/accounts/120456/balance" -H "accept: application/json"
```

Request URL

http://localhost:9089/AccountBalance/api/v1.0.0/party/accounts/120456/balance

Checking API in Postman:

Run postman → click '+' button → paste your API link and add War filename in the link.

Click **Send** → Response will be shown.

HTTP http://localhost:9089/CusAcBalContainer/api/v1.0.0/party/ws/800155 Save

GET http://localhost:9089/CusAcBalContainer/api/v1.0.0/party/ws/800155 Send

Params Authorization (7) Headers (7) Body Pre-request Script Tests Settings Cookies

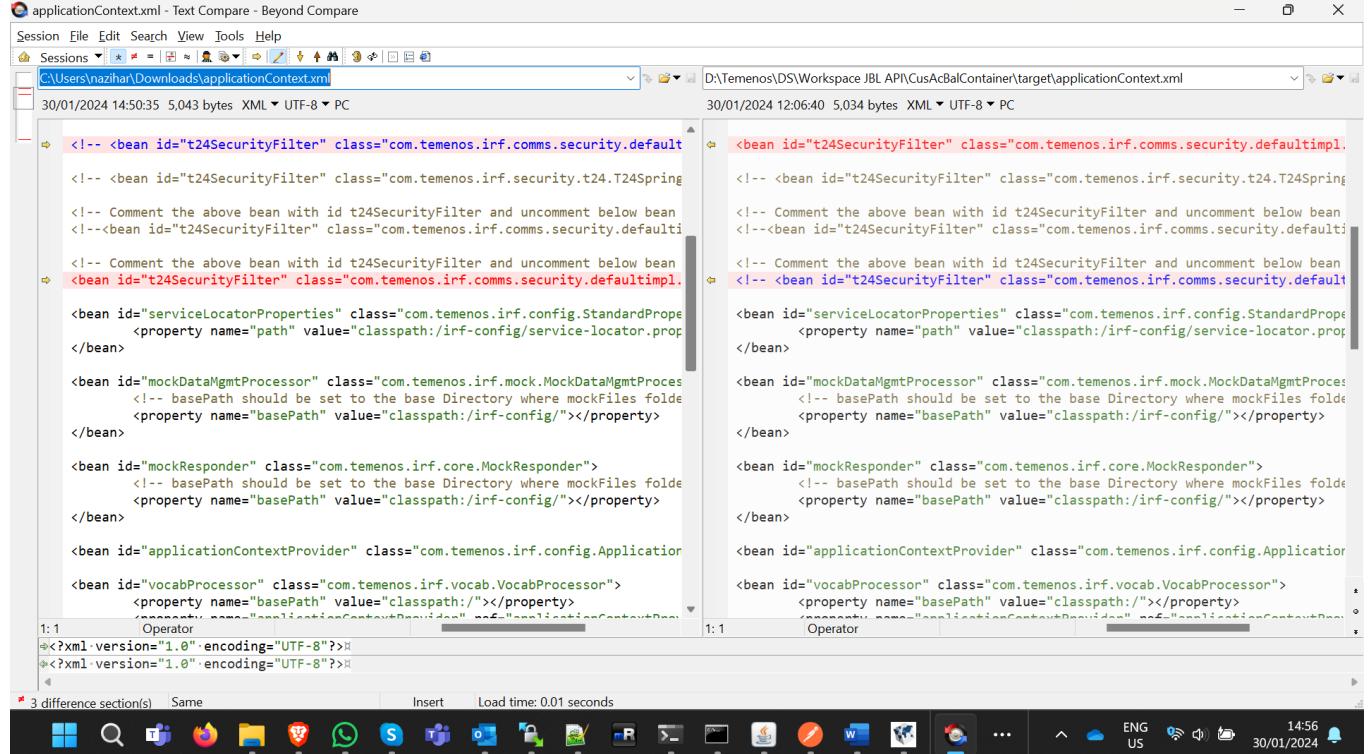
If authentication(Basic Auth) error is found then **applicationContext.xml** file need to be changed.

Comment out these lines:

```
<!-- <bean id="t24SecurityFilter" class="com.temenos.irf.comms.security.defaultimpl.NullBean" /> -->
<!-- <bean id="irisProcessorAspect" class="com.temenos.irf.core.xacml.authz.IrisProcessorXacmlAspect" /> -->
```

Uncomment:

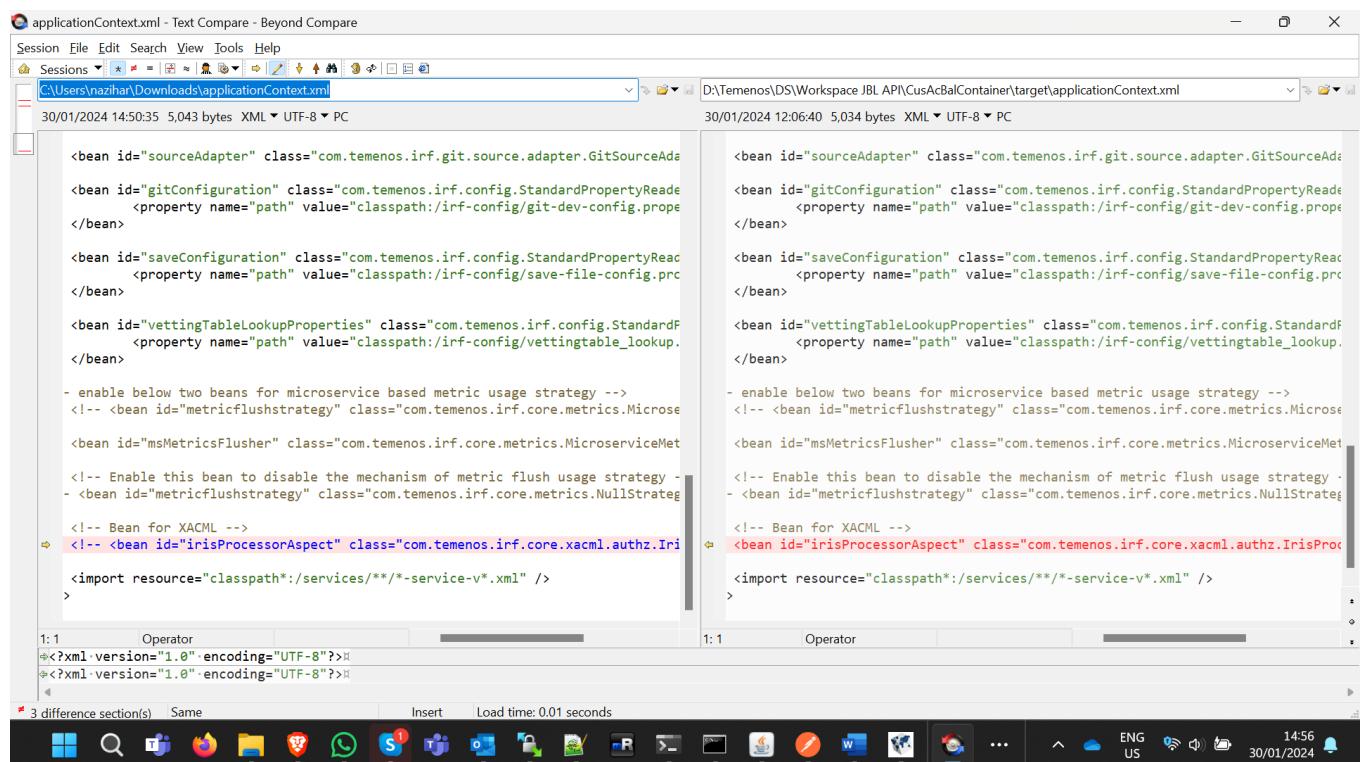
```
<bean id="t24SecurityFilter" class="com.temenos.irf.comms.security.defaultimpl.T24BasicAuthenticationCheck" />
```



The screenshot shows a comparison between two versions of the `applicationContext.xml` file using the Beyond Compare tool. The left pane displays the original file from `C:\Users\nazihar\Downloads\applicationContext.xml`, and the right pane shows the modified file from `D:\Temenos\DS\Workspace JBL API\CusAcBalContainer\target\applicationContext.xml`. The modified file contains uncommented bean definitions for `t24SecurityFilter` and `irisProcessorAspect`.

```
<!-- <bean id="t24SecurityFilter" class="com.temenos.irf.comms.security.defaultimpl.NullBean" /> -->
<!-- <bean id="irisProcessorAspect" class="com.temenos.irf.core.xacml.authz.IrisProcessorXacmlAspect" /> -->

<bean id="t24SecurityFilter" class="com.temenos.irf.comms.security.defaultimpl.T24BasicAuthenticationCheck" />
```



The screenshot shows a comparison between two versions of the `applicationContext.xml` file using the Beyond Compare tool. The left pane displays the original file from `C:\Users\nazihar\Downloads\applicationContext.xml`, and the right pane shows the modified file from `D:\Temenos\DS\Workspace JBL API\CusAcBalContainer\target\applicationContext.xml`. The modified file contains uncommented bean definitions for `sourceAdapter`, `gitConfiguration`, `saveConfiguration`, `vettingTableLookupProperties`, `msMetricsFlusher`, `metricflushstrategy`, and `irisProcessorAspect`.

```
<!-- <bean id="sourceAdapter" class="com.temenos.irf.git.source.adapter.GitSourceAda
<!-- <bean id="gitConfiguration" class="com.temenos.irf.config.StandardPropertyReade
<!-- <bean id="saveConfiguration" class="com.temenos.irf.config.StandardPropertyReac
<!-- <bean id="vettingTableLookupProperties" class="com.temenos.irf.config.StandardF
<!-- <bean id="msMetricsFlusher" class="com.temenos.irf.core.metrics.MicroserviceMet
<!-- Enable this bean to disable the mechanism of metric flush usage strategy -
<!-- <bean id="metricflushstrategy" class="com.temenos.irf.core.metrics.NullStrate
<!-- Bean for XACML --
<!-- <bean id="irisProcessorAspect" class="com.temenos.irf.core.xacml.authz.Iri
<import resource="classpath*:services/**/*-service-v*.xml" />

<!-- <bean id="sourceAdapter" class="com.temenos.irf.git.source.adapter.GitSourceAda
<!-- <bean id="gitConfiguration" class="com.temenos.irf.config.StandardPropertyReade
<!-- <bean id="saveConfiguration" class="com.temenos.irf.config.StandardPropertyReac
<!-- <bean id="vettingTableLookupProperties" class="com.temenos.irf.config.StandardF
<!-- <bean id="msMetricsFlusher" class="com.temenos.irf.core.metrics.MicroserviceMet
<!-- Enable this bean to disable the mechanism of metric flush usage strategy -
<!-- <bean id="metricflushstrategy" class="com.temenos.irf.core.metrics.NullStrate
<!-- Bean for XACML --
<!-- <bean id="irisProcessorAspect" class="com.temenos.irf.core.xacml.authz.Iri
<import resource="classpath*:services/**/*-service-v*.xml" />
```

After changing and replacing the file in the war file it works!

The screenshot shows a Postman interface with the following details:

- Request URL:** `http://localhost:9089/CusAcBalContainer/api/v1.0.0/party/ws/800155`
- Method:** GET
- Headers:** Postman-Token (selected), Host
- Body:** JSON (Pretty)
- Response Status:** 200 OK, 12.33 s, 1.49 KB
- Response Body (Pretty JSON):**

```
1  {
2      "header": {
3          "audit": {
4              "T24_time": 12045,
5              "responseParse_time": 1,
6              "requestParse_time": 271
7          },
8          "page_start": 1,
9          "page_token": "202104152599054588.01,99",
10         "total_size": 4,
11         "page_size": 99,
12         "status": "success"
13     },
14     "body": [
15         {
16             "accountId": "117258",
17             "clearedBalance": 9986222.5,
18             "displayName": "MD SHIBLI MOLLAH",
19             "workingBalance": 9186222.5,
20             "onlineActualBalance": 9986222.5,
21             "currencyId": "USD",
22             "categoryId": "Current Account",
23             "availableBalance": 9186222.5
24         },
25         {
26             "accountId": "120448",
27                 "clearedBalance": 44993682.15,
28                 "displayName": "MD SHIBLI MOLLAH",
29                 "workingBalance": 44993682.15,
30                 "onlineActualBalance": 44993682.15,
31                 "currencyId": "USD",
32                 "categoryId": "Current Account",
33                 "availableBalance": 44993682.15
34         },
35         {
36             "accountId": "120456",
37             "clearedBalance": 999212.5,
38             "displayName": "SHIBLI SAVINGS",
39             "workingBalance": 999212.5,
40                 "onlineActualBalance": 999212.5,
41                 "currencyId": "USD",
42                 "categoryId": "Staff Account",
43                 "availableBalance": 999212.5
44         },
45         {
46             "accountId": "120464",
47             "clearedBalance": 998962.5,
48             "displayName": "MD SHIBLI",
49             "workingBalance": 998962.5,
50             "onlineActualBalance": 998962.5,
51             "currencyId": "USD",
52         }
53     ]
54 }
```

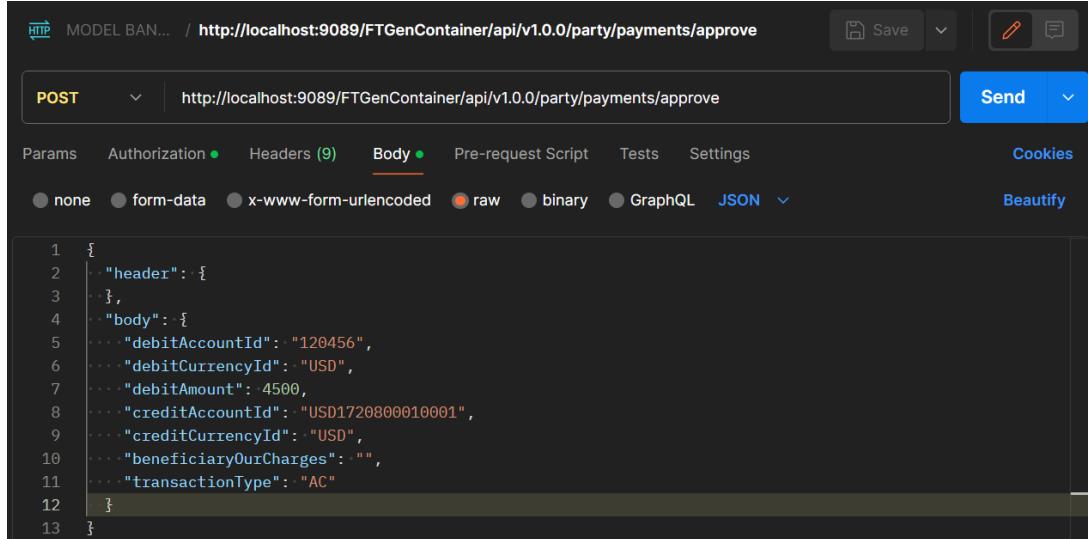
Response:

```
{  
    "header": {  
        "audit": {  
            "T24_time": 12045,  
            "responseParse_time": 1,  
            "requestParse_time": 271  
        },  
        "page_start": 1,  
        "page_token": "202104152599054588.01,99",  
        "total_size": 4,  
        "page_size": 99,  
        "status": "success"  
    },  
    "body": [  
        {  
            "accountId": "117258",  
            "clearedBalance": 9986222.5,  
            "displayName": "MD SHIBLI MOLLAH",  
            "workingBalance": 9186222.5,  
            "onlineActualBalance": 9986222.5,  
            "currencyId": "USD",  
            "categoryId": "Current Account",  
            "availableBalance": 9186222.5  
        },  
        {  
            "accountId": "120448",  
            "clearedBalance": 44993682.15,  
            "displayName": "MD SHIBLI MOLLAH",  
            "workingBalance": 44993682.15,  
            "onlineActualBalance": 44993682.15,  
            "currencyId": "USD",  
            "categoryId": "Current Account",  
            "availableBalance": 44993682.15  
        },  
        {  
            "accountId": "120456",  
            "clearedBalance": 999212.5,  
            "displayName": "SHIBLI SAVINGS",  
            "workingBalance": 999212.5,  
            "onlineActualBalance": 999212.5,  
            "currencyId": "USD",  
            "categoryId": "Staff Account",  
            "availableBalance": 999212.5  
        },  
        {  
            "accountId": "120464",  
            "clearedBalance": 998962.5,  
            "displayName": "MD SHIBLI",  
            "workingBalance": 998962.5,  
            "onlineActualBalance": 998962.5,  
            "currencyId": "USD",  
            "categoryId": "Savings Acct",  
            "availableBalance": 998962.5  
        }  
    ]  
}
```

VERSION - POST

Api creation and war file creation is almost same as the process for ENQUIRY.

POST request format:



```
POST http://localhost:9089/FTGenContainer/api/v1.0.0/party/payments/approve
```

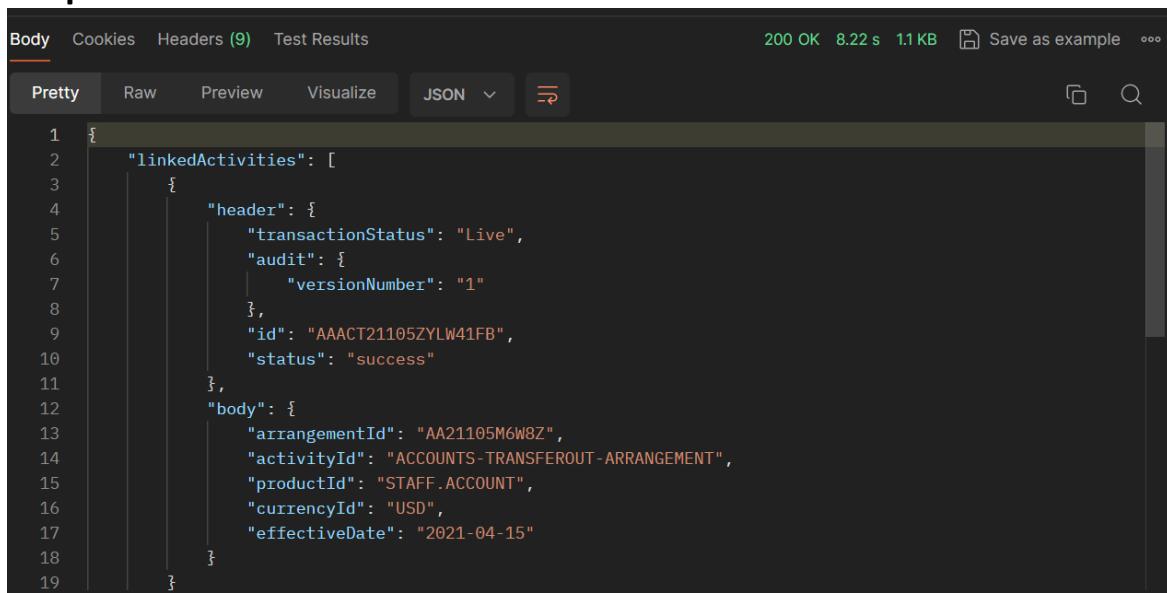
Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Beautify

```
1 {
2   "header": {
3     ...
4   },
5   "body": {
6     "debitAccountId": "120456",
7     "debitCurrencyId": "USD",
8     "debitAmount": 4500,
9     "creditAccountId": "USD1720800010001",
10    "creditCurrencyId": "USD",
11    "beneficiaryOurCharges": "",
12    "transactionType": "AC"
13 }
```

Body:

```
{  
  "header": {  
  },  
  "body": {  
    "debitAccountId": "120456",  
    "debitCurrencyId": "USD",  
    "debitAmount": 4500,  
    "creditAccountId": "USD1720800010001",  
    "creditCurrencyId": "USD",  
    "beneficiaryOurCharges": "",  
    "transactionType": "AC"  
  }  
}
```

Response:



Body Cookies Headers (9) Test Results 200 OK 8.22 s 1.1 KB Save as example ⚙

Pretty Raw Preview Visualize JSON ⌂ ⌂

```
1 {  
  "linkedActivities": [  
    {  
      "header": {  
        "transactionStatus": "Live",  
        "audit": {  
          "versionNumber": "1"  
        },  
        "id": "AACT21105ZYLW41FB",  
        "status": "success"  
      },  
      "body": {  
        "arrangementId": "AA21105M6W8Z",  
        "activityId": "ACCOUNTS-TRANSFEROUT-ARRANGEMENT",  
        "productId": "STAFF-ACCOUNT",  
        "currencyId": "USD",  
        "effectiveDate": "2021-04-15"  
      }  
    }  
  ]  
}
```

```

        ],
        "header": {
            "transactionStatus": "Live",
            "audit": {
                "T24_time": 8209,
                "responseParse_time": 4,
                "requestParse_time": 2,
                "versionNumber": "1"
            },
            "id": "FT21105RT0MY",
            "status": "success"
        },
        "body": {
            "transactionType": "AC",
            "debitCurrencyId": "USD",
            "processingDate": "2021-04-15",
            "creditCurrencyId": "USD",
            "debitAccountId": "120456",
            "debitAmount": 4500,
            "creditAccountId": "USD1720800010001"
        }
    }
}

```

Orchestration

Question: Creating orchestration for FT and Account

Requirement : After Authorization of FT(version), The enquiry which shows the debit Account Id's Available balance

Step 1:

Choose the artefacts in the workbench and generate the zip file.

Enquiry : FUNDS.TRANSFER,FT.API.GENERIC.1.0.0

Version : PZ.API.ACCOUNTS.BALANCE.1.0.0

API Definition

Title	FTOrch	Key	mb-api
Description	FTOrchestration	Only Alphabetic characters are allowed	
Version	v1.0.0	Schemes	<input checked="" type="checkbox"/> https <input type="checkbox"/> http
Base Path	/api	Host	localhost:9089

API Endpoints

Path	/payments/approve
HTTP Method	POST
Operation	createFundsTransfer
Operation Security	Public
Target	FUNDS.TRANSFER,FT.API.GENERIC.1.0.0
Target Type	Screen

Path	/accounts/{accountId}/balance
HTTP Method	GET
Operation	getAccountBalance
Operation Security	Public
Target	PZ.API.ACCOUNTS.BALANCE.1.0.0
Target Type	Query

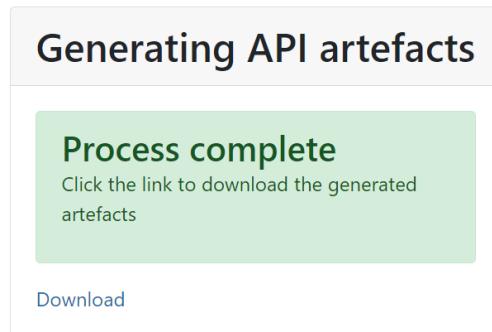
Inventory

```
{  
  "paths": [  
    {  
      "method": "POST",  
      "url": "/party/payments/approve",  
      "function": "input",  
      "tags": [],  
      "operationId": "createFundsTransfer",  
      "operationSecurity": "Public",  
      "resources": [  
        {  
          "key": "FUNDS.TRANSFER,FT.API.GENERIC.1.0.0",  
          "resourceType": "Screen",  
          "consentManaged": false  
        }  
      ],  
      "properties": {  
        "clientType": "INTERNAL",  
        "isBulk": false,  
        "deprecated": false  
      }  
    },  
    {  
      "method": "GET",  
      "url": "/party/accounts/{accountId}/balance",  
      "tags": [],  
      "operationId": "getAccountBalance",  
      "operationSecurity": "Public",  
      "resources": [  
        {  
          "key": "PZ.API.ACCOUNTS.BALANCE.1.0.0",  
          "resourceType": "Query",  
          "consentManaged": false,  
          "selections": [  
            {  
              "field": "ACCOUNTREFERENCE",  
              "param": "accountId",  
              "operand": "EQ",  
              "required": "",  
              "type": "string"  
            }  
          ]  
        }  
      ],  
      "properties": {  
        "clientType": "INTERNAL",  
        "isBulk": false,  
        "deprecated": false  
      }  
    },  
    {"version": "v1.0.0",  
     "title": "FTOrch",  
     "description": "The FTOrch API provides a set of endpoints for managing funds transfers between parties."}  
  ]  
}
```

```

"description": "FTOrchestration",
"key": "mb-api",
"schemes": [
    "http",
    "https"
],
"basepath": "/api",
"host": "localhost:9089"
}

```



Step 2:

Extract the zip file and place it in the new service project.
src -> main -> resources

Step 3:

Open the service xml file in the project.

```

└─ FtEnqGen
    ├─ src/main/java
    └─ src/main/resources
        ├─ api-docs
        ├─ inventory
        └─ services
            ├─ party-accounts
            │   ├─ party-accounts_mock_v1.0.0
            │   │   └─ party-accounts-mb-api-service-v1.0.0.xml
            │   └─ party-accounts-mb-api-service-v1.0.xml
            └─ party-payments
                ├─ party-payments_mock_v1.0.0
                │   └─ party-payments-mb-api-service-v1.0.0.xml
                └─ party-payments-mb-api-service-v1.0.xml
                    └─ party-payments-mb-api-service-v1.xml

```

Step 4:

Open the **xml file** and write the below code under the **t24 VERSION** processor line in the xml file.

```

<choice>
    <when>
        <simple>${headers.CamelHttpResponseCode} == 200</simple>
        <setHeader headerName = "debitAccountId" >
            <camel:jsonpath>$.body.debitAccountId</camel:jsonpath>
        </setHeader>
        <to uri="direct-vm:party-accounts.v1.0.0.getAccountBalance" />
    </when>
</choice>

```

Above xml file is for **VERSION**. Here I am passing debitAccountId field to the **ENQUIRY** as selection field.

Open the ENQUIRY xml file:

Here we are Mapping the **debitAccountId** to the selection field. (Here we passing **headerName** which is written in the version xml file)

Edit this line: `<constant>ACCOUNTREFERENCE EQ {debitAccountId}</constant>`

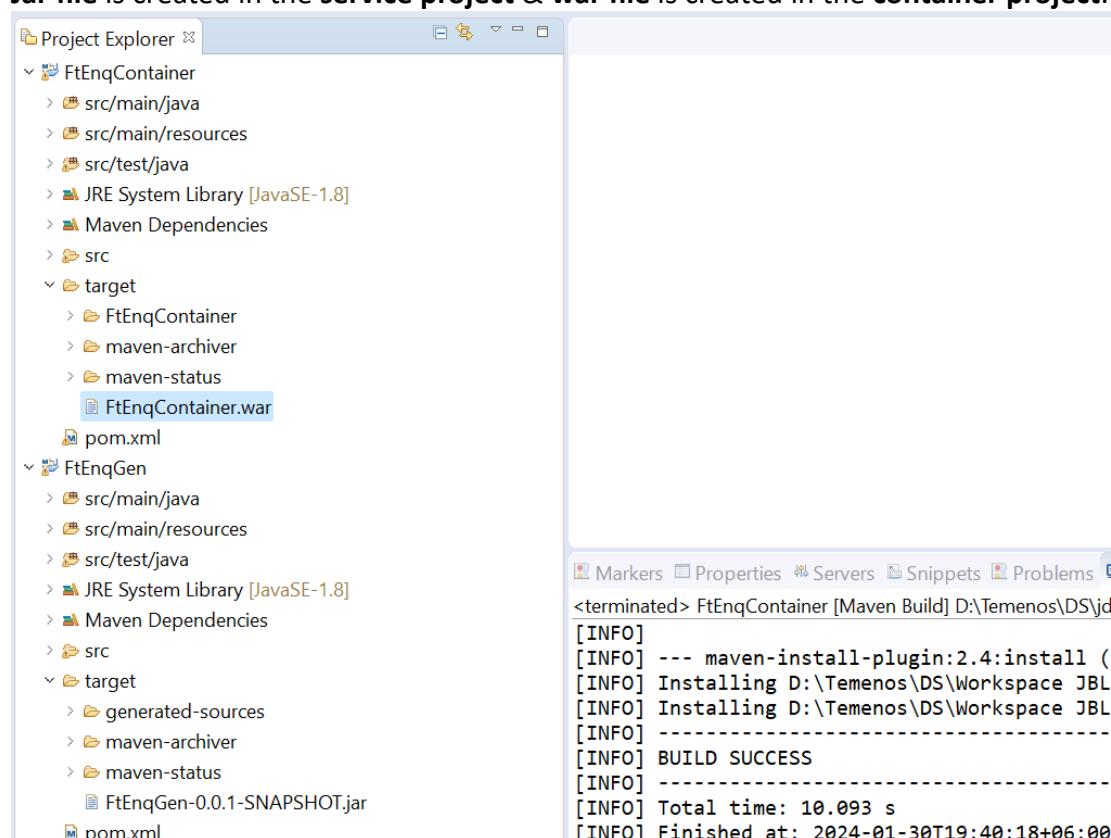


```
party-accounts-mb-api-service-v1.0.0.xml [x] party-payments-mb-api-service-v1.0.0.xml
<from uri="direct-vm:party-accounts.v1.0.0.getAccountBalance"/>
<setProperty propertyName="operationSecurity">
    <constant>Public</constant>
</setProperty>
<setProperty propertyName="isBulk">
    <constant>false</constant>
</setProperty>
<setProperty propertyName="target">
    <constant>PZ.API.ACCOUNTS.BALANCE.1.0.0</constant>
</setProperty>
<setProperty propertyName="selection1">
    <constant>ACCOUNTREFERENCE EQ {debitAccountId}</constant>
</setProperty>
<process ref="t24EnquiryProcessor"/>
</route>
<route id="direct.mockResponder">
    <from uri="direct:mockResponder"/>
    <process ref="mockResponder"/>
</route>
</camelContext>
</beans>
```

Step 5:

Right click on both project and Container **RunAs-> Maven clean & Install**.

Jar file is created in the **service project** & **war file** is created in the **container project**.



Step 6:

Deploy the war file in JBoss.

Add the server in the workbench.

- FtEnqContainer http://localhost:9089/FtEnqContainer



API Framework Workbench Dashboard ▾ APIs ▾ Vocabulary ▾ Settings ▾

1 Domains 6 Endpoints 7 Services 3 APIs

other

mb-api-v1.0.0-swagger v1.0.0 API docs

party-payments-mb-api party-accounts-mb-api

POST /v1.0.0/party/payments/approve GET /v1.0.0/party/accounts/{accountId}/balance

<http://localhost:9089/FtEnqContainer/api/v1.0.0/party/payments/approve>

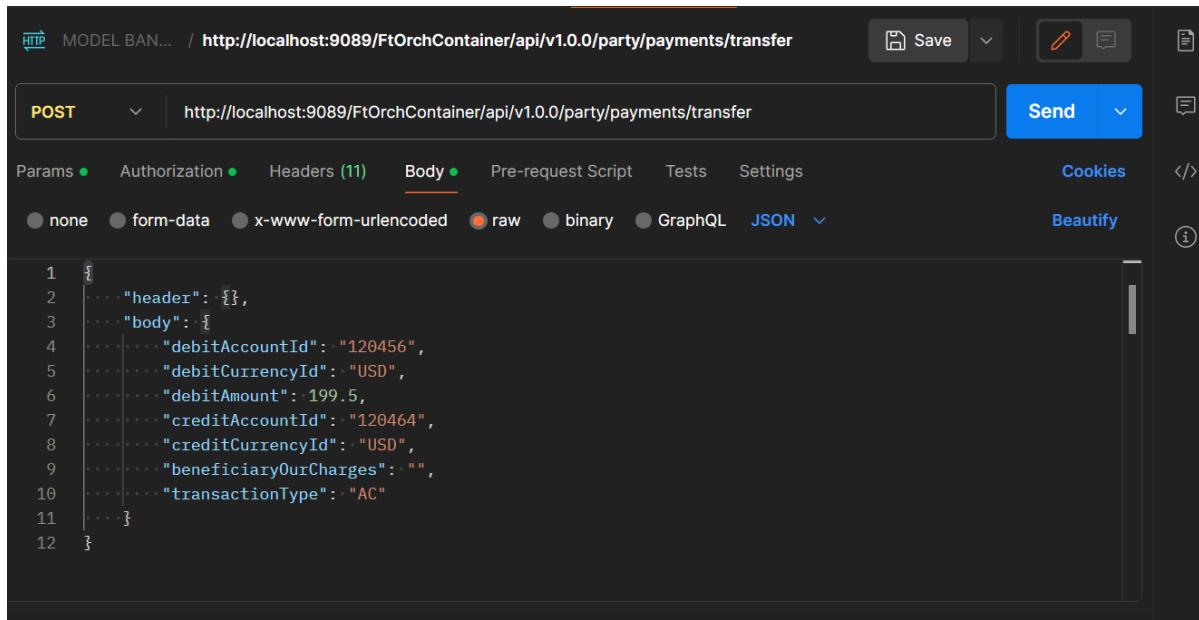
Special Note: If Authorization issue occurs then we need to update the [service xml](#)



LATESTparty-payments-orc-api-service-v1.0.0.xml

```
<setProperty propertyName="validate_only">
    <header>true</header>
</setProperty>
    <setProperty propertyName="userDetails">
        <header>Authorization</header>
    </setProperty>
    <setProperty propertyName="function">
        <constant>validate</constant>
    </setProperty>
<process ref="t24VersionProcessor"/>
<choice>
    <when>
        <simple>${headers.CamelHttpResponseCode} == 200</simple>
        <setHeader headerName="accountId">
            <camel:jsonpath suppressExceptions="true">$.body.debitAccountId</camel:jsonpath>
        </setHeader>
        <setHeader headerName="Authorization">
            <camel:exchangeProperty>userDetails</camel:exchangeProperty>
        </setHeader>
        <to uri="direct-vm:party-accounts.v1.0.0.getAcBalance" />
    </when>
</choice>
</route>
```

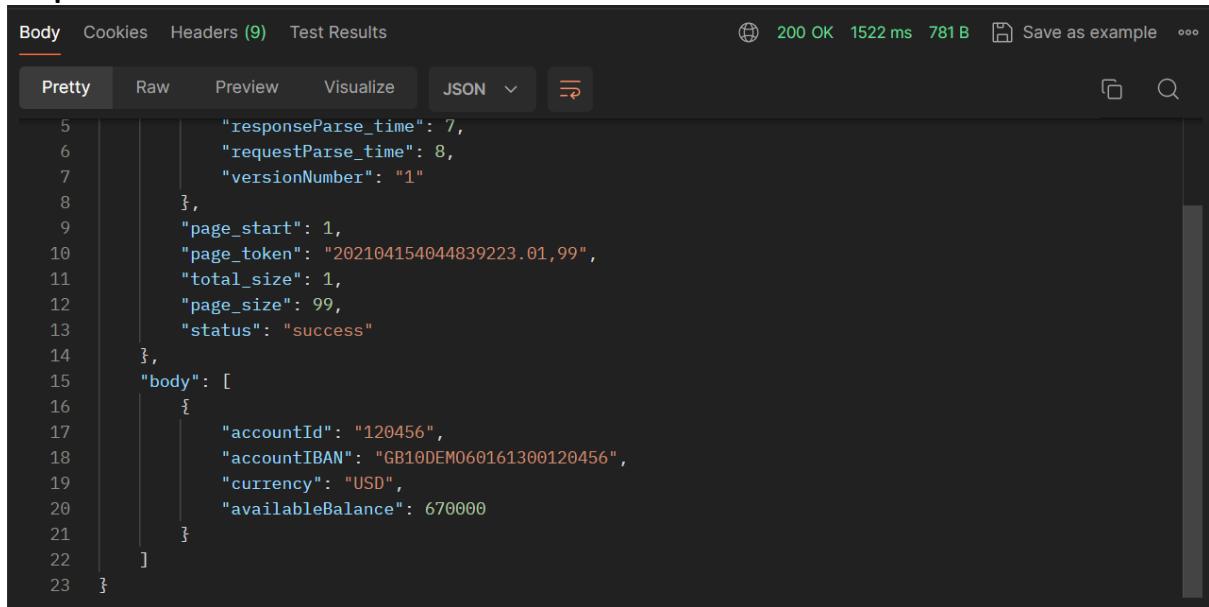
Step 7: Check it in POSTMAN



The screenshot shows the POSTMAN interface with a POST request to `http://localhost:9089/FtOrchContainer/api/v1.0/party/payments/transfer`. The request body is defined as follows:

```
1 {
2   ...
3   "header": {},
4   ...
5   "body": {
6     ...
7     "debitAccountId": "120456",
8     ...
9     "debitCurrencyId": "USD",
10    ...
11    "debitAmount": 199.5,
12    ...
13    "creditAccountId": "120464",
14    ...
15    "creditCurrencyId": "USD",
16    ...
17    "beneficiaryOurCharges": "",
18    ...
19    "transactionType": "AC"
20  }
21 }
```

Response:



The screenshot shows the POSTMAN interface displaying the response from the API call. The response is a 200 OK status with 1522 ms latency and 781 B size. The response body is a JSON object containing the following fields:

```
5   ...
6   ...
7   ...
8   ...
9   ...
10  ...
11  ...
12  ...
13  ...
14  ...
15  ...
16  ...
17  ...
18  ...
19  ...
20  ...
21  ...
22  ...
23  {
```

XACML/ JWT Token

What is Xacml?

Extensible Access Control Markup Language – Which is used for attribute based authorization.

For example, An API request must contain “**roleId**” as “**ADMIN**”. This **roleId** we can pass through headers or **JWT Token**.

Xacml policy file will be available inside the container.

`C:\Users\user\Downloads\jrf-provider-container.war\WEB-INF\classes\xacml\`

To Enable xacml: NO NEED TO ENABLE XACML if it's JWT Only. Keep the *applicationContext.xml* as it is.
Uncomment below line to enable xacml.

```

<!-- enable below two beans for microservice based metric usage strategy -->
<!-- <bean id="metricflushstrategy" class="com.temenos.irf.core.metrics.MicroserviceSimpleCountFlushStrategy" />
<bean id="msMetricsFlusher" class="com.temenos.irf.core.metrics.MicroserviceMetricsFlusher"/> -->
<!-- Enable this bean to disable the mechanism of metric flush usage strategy -->
<!-- <bean id="metricflushstrategy" class="com.temenos.irf.core.metrics.NullStrategy" /> -->
<!-- Bean for XACML -->
<bean id="irisProcessorAspect" class="com.temenos.irf.core.xacml.authz.IrisProcessorXacmlAspect" />
<import resource="classpath:/services/**/*-service-v*.xml" />

```

IRIS Auth token Generation(JWT Only)

The functionality required in IRIS is to provide capability in IRIS to validate the credentials passed in **header(Basic Authentication)** against T24 and send back the JWT token for successful authentication.

Deploy the war file **irf-auth-token-generation-container.war** in JBoss.

Check it in the postman by the below URL

<http://<host>:<port>/IrisAuthTokenGenerator/api/v1.0.0/generateauthToken>

[irf-auth-token-generation-container-21.0.59.war](http://<host>:<port>/IrisAuthTokenGenerator/api/v1.0.0/generateauthToken)

<http://<host>:<port>/IrisAuthTokenGenerator/api/v1.0.0/generateauthToken>

<http://localhost:9089/irf-auth-token-generation-container-21.0.59/api/v1.0.0/generateauthToken>

pass with credentials,

The Auth token has been generated.

HTTP ... / http://localhost:9089/irf-auth-token-generation-container-21.0.59/api/v1.0.0/gene...

GET http://localhost:9089/irf-auth-token-generation-container-21.0.59/api/v1.0.0/generateauthToken Send

Params • Authorization • Headers (8) Body Pre-request Script Tests Settings Cookies

Type Basic A... Username SHIBLI.02
The authorization header will be automatically generated when you send the request. Learn more about [Basic Auth](#) authorization.

Authorization: Basic SHIBLI.02:12345678

Body Cookies Headers (9) Test Results 200 OK 175 ms 985 B Save as example

Pretty Raw Preview Visualize JSON

```

1  {"\\"id_token\\":\"eyJhbGciOiJSUzI1NiJ9.
eyJpc3Mi0iJURU1FTk9TlwiYXVkJjoiVDI0VXNlcIIsImhdCI6MTcwNzEwOTg10SwiZXhwIjoxNzA3MTExNjU5LCJzdWIi0JTSE
1CTEkuMDIifQ.
DLkn0CP8B3IqnflXVK-40LEbRmKx4g88fG36aoYzTsZXpk8vBHJrL3XqwlPDEMGZA-dYdrJKKtxZITF04JqB11vWCdcpUrcKjwE-P9
hbaHNgAU0jk_tv1DlldVbmDu0ZeyK4p0pY48T_KWoGz4PpPbbjT_JsY_Fpd0mQu1sb5MOSHfRo27k-PEHO7UytM9I7QVRxASTmFiT
nWgdw0q51-Be33x6siZLef2Ag0gxo8RAUDizUFGdy-dScUkLA5mo9II1D11fNaYgCeW-wjNS9Z06MFaG4HCTD410pH0Y_g93kTygZ
zGWN07q47eG_n_YP8mLgsbCN9_Mv6PxFSN2Q\", \"token_type\": \"Bearer\"}"}

```

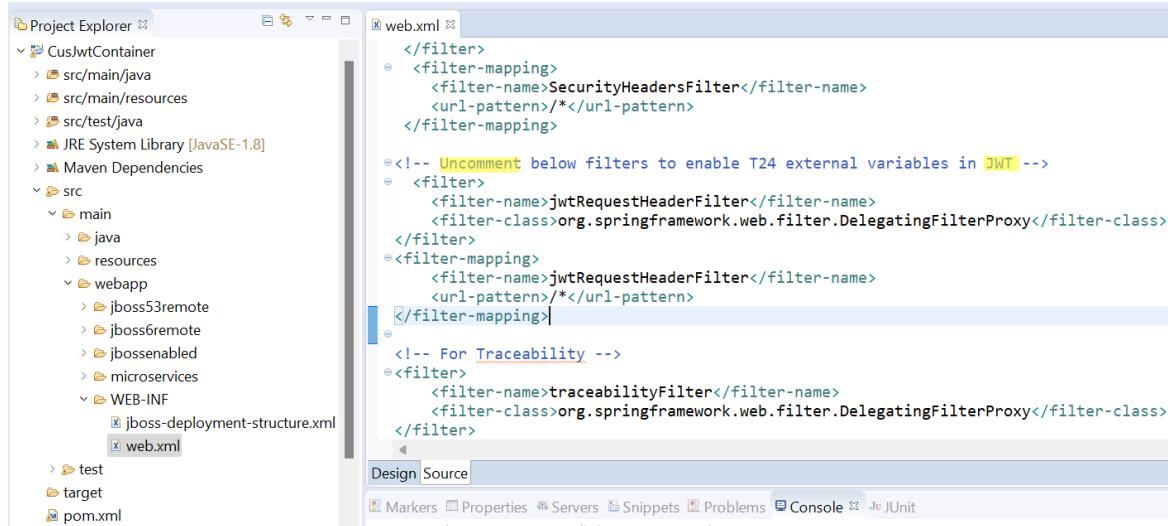
<https://jwt.io/>

The screenshot shows the jwt.io interface. On the left, under 'Encoded', is a long string of characters representing a JWT token. On the right, under 'Decoded', are the individual components:

- HEADER: ALGORITHM & TOKEN TYPE**: Contains the algorithm "RS256".
- PAYOUT: DATA**: Contains a JSON object with fields: iss (TEMENOS), aud (T24User), iat (1707109859, timestamp: Mon Feb 05 2024 11:10:59 GMT+0600 (Bangladesh Standard Time)), exp (1707111659), and sub (SHIBLI.02).
- VERIFY SIGNATURE**: A placeholder for a public key.

To Enable JWT:

UnComment below set of lines to enable JWT.

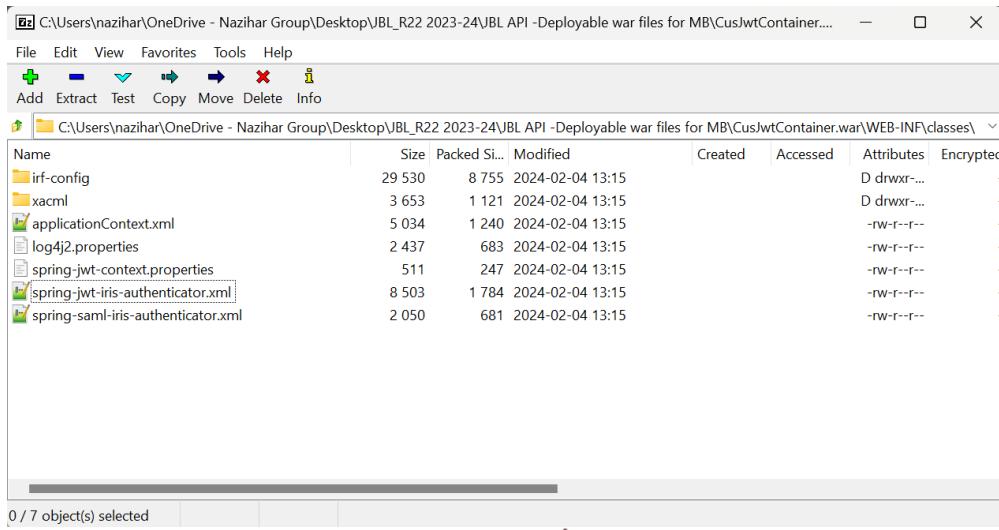


Change spring-jwt-iris-authenticator.xml:

Changed the OIDC provider issuer to TEMENOS if there is an issue with the provider like the SS below.

The screenshot shows a Postman test result for a 401 Unauthorized status. The response body is a JSON object with a message indicating that token validation failed because the expected issuer did not match the actual issuer, and a status of 401.

```
1 {
2   "message": "Token validation failed [Expected issue doesn't match with actual issuer!]",
3   "status": 401
4 }
```



After changing the xml file

spring-jwt-iris-authenticator.xml

it will look like this:

```
<!-- OIDC provider Server -->
<beans:bean id="oidcProviderServer" class="com.temenos.security.oidc.common.OidcProviderServer">
    <beans:property name="issuer" value="TEMENOS" />
    <beans:property name="pkEncoded" value="" />
    <beans:property name="pkCertEncoded" value="" />
    <beans:property name="pkCertFilePath" value="" />
    <beans:property name="pkJwksUri" value="" />
    <beans:property name="decryptingJwkEncoded" value="" />
    <beans:property name="principalClaim" value="sub" />
</beans:bean>
<!-- End OIDC provider Server -->
```

Now using API Key, the response is **OK** now.

Body	Cookies (1)	Headers (13)	Test Results
<pre> 1 { 2 "header": { 3 "audit": { 4 "T24_time": 13966, 5 "responseParse_time": 2, 6 "requestParse_time": 2152 7 }, 8 "page_start": 1, 9 "page_token": "202104154995439532.00,99", 10 "total_size": 4, 11 "page_size": 99, 12 "status": "success" </pre>			Save as example

Create JWT using Web tool(<https://www.javainuse.com/jwtgenerator>)

JWT Payload

Claim Type	Claim Value	
Iss	TEMENOS	X
Iat	2024-02-05T06:41:29.437Z	X
Exp	2024-02-05T06:41:29.437Z	X
Sub	JavaInUse	X
roleId	ADMIN	X

Payload to be generated

```
{  
    "Iss": "TEMENOS",  
    "Iat": "2024-02-05T06:41:29.437Z",  
    "Exp": "2024-02-05T06:41:29.437Z",  
    "Sub": "JavaInUse",  
    "roleId": "ADMIN"  
}
```

[remove all claims](#) [add claim](#)

Create JSON Web Token Using Secret Key

Algorithm: HS256 ▾ Key: javainuse-secret-key

[Create JWT](#)

```
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJKYXZhSw5Vc2UiLCJyb2x1SWQiOiJBRE1JTiIsIk1zcycI6IlRFTUVOT1MiLCJFeHAiOiIyMDI0LTAyLTA1VDA2OjQxOjI5Lj0
```

To be Continued...