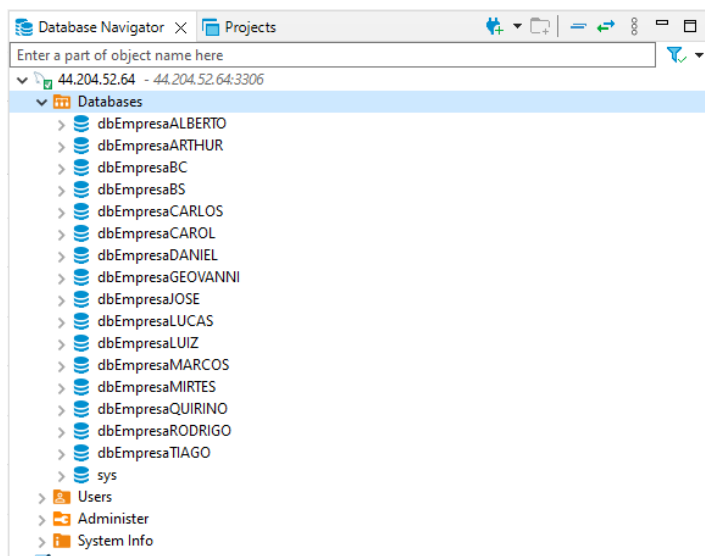


# Aula 03

## Exercícios:

### Projeto Empresa

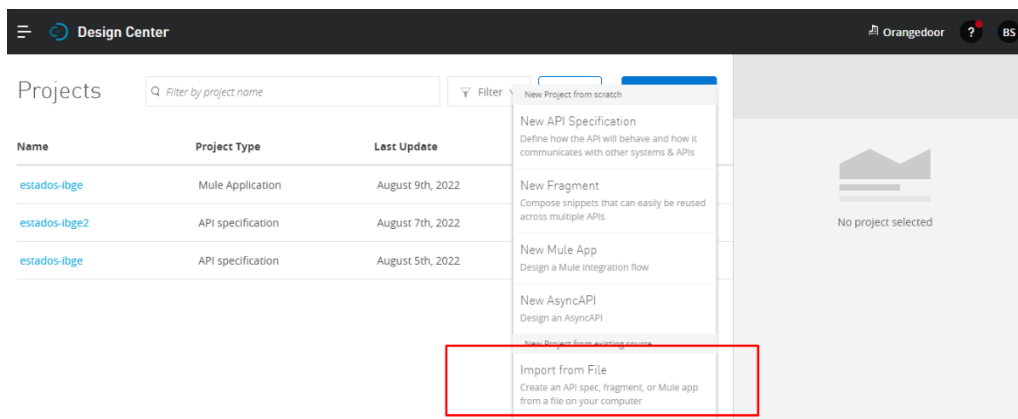
Neste momento trabalhar com o Projeto Empresa, onde disponibilizamos a todos os alunos um banco de dados na AWS, onde criaremos APIs para realizar consultas, inclusões alterações e exclusões de dados neste banco externo.



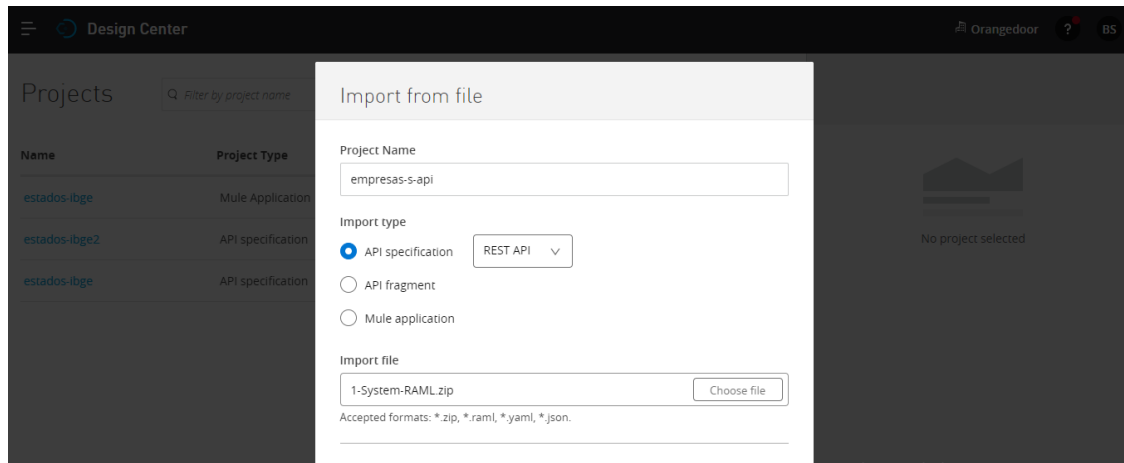
Agora iremos importar o RAML disponibilizado abaixo para que possamos utilizá-lo em nosso projeto.

Download: [System\\_Raml.zip](#)

Na Anypoint Platform > Design Center > + Create New > Import from File



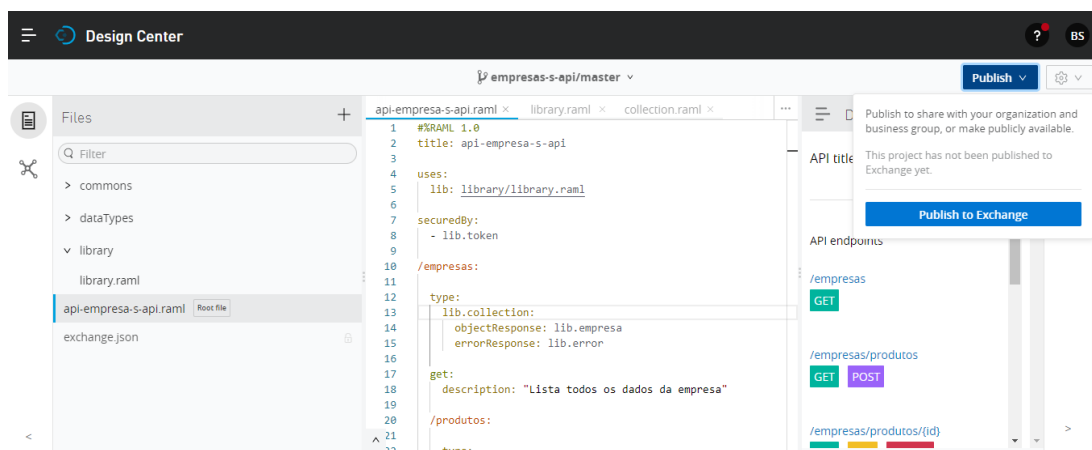
- Digitar o nome do projeto: **empresas-s-api-bs** e selecione o arquivo **System\_Raml.zip**



Clique em “Import”



- Será exibida toda a estrutura do RAML importado no Design Center



Agora iremos publicar nosso RAML no Exchange clicando no botão “Publish” e “Publish to Exchange”

Preencha os campos Asset Version = “1.0.0”, API = “v1” e abrir a aba “more options” e alterar o Asset Name para “Empresas”

**empresas-s-api**

Asset version (required)  
1.0.0

API version (required)  
v1

Asset versioning  
To publish to Exchange, you must version assets using [SemVer](#). Examples of good versions are 1.0.0 or 4.3.1.

Additional help  
• [Changing a project's main/root file](#)  
• [What is an API version?](#)

LifeCycle State ⓘ  
☒ Stable  
☐ Development

Stable  
State of release, ready to consume

Development  
In Process of Design and Development

▼ More options

Business Group ID  
d5c62e50-16a3-49db-bdf1-327f7edb2cb1 ▼

Asset ID  
empresas-s-api

Asset name  
Empresas

Cancel Publish to Exchange

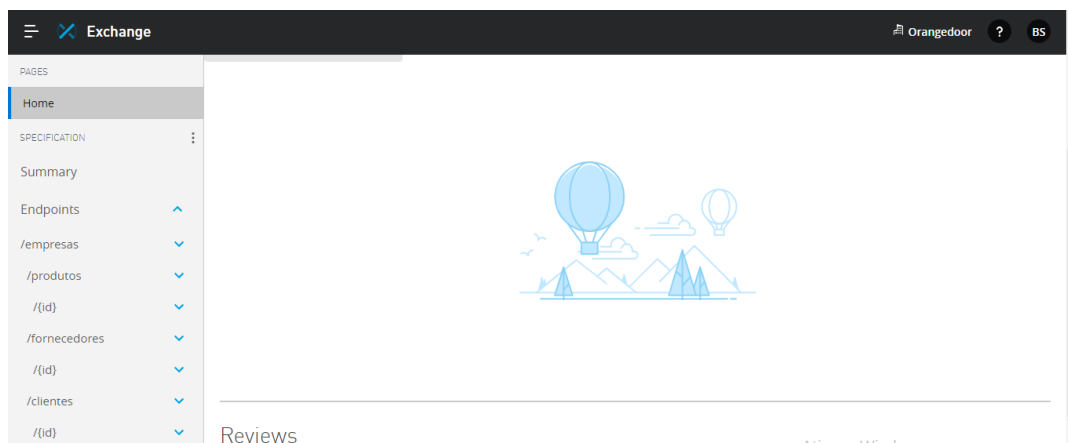
Clique no botão “Publish to Exchange”

**empresas-s-api**

Successfully published to [Exchange](#)

Done

Clicando no link “Exchange” você poderá ver o contrato de sua API.

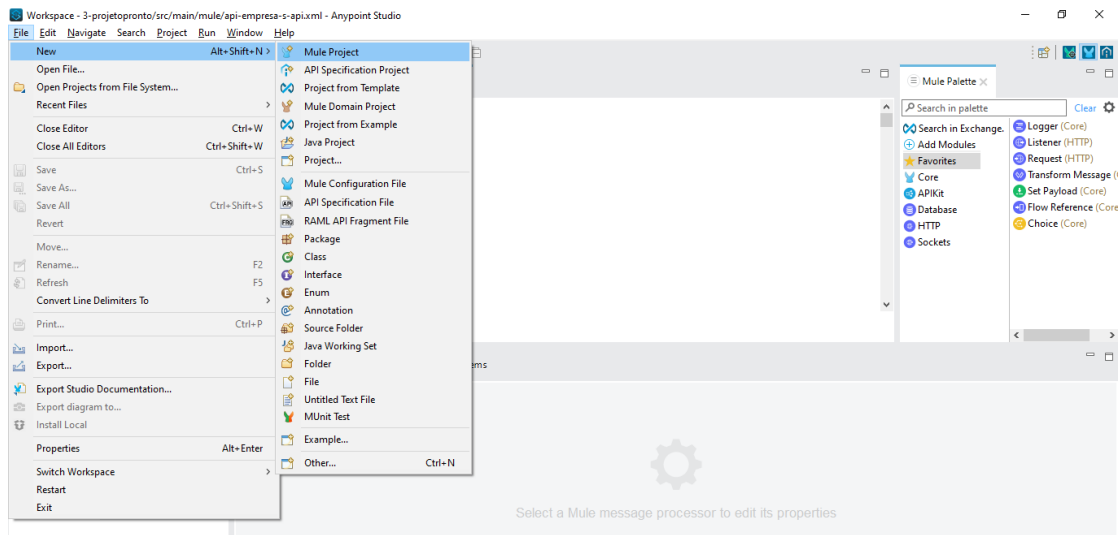


Após a publicação do contrato de sua API no Exchange, vamos iniciar a implementação de nossa API. (design first).

### Criando API no Anypoint Studio

Após o RAML pronto, iremos criar nossa API no Anypoint Studio, importando os dados do contrato do Exchange

Clique no menu File / New / Mule Project



Digite o nome do projeto “empresas-s-api”, na aba “Import a Published API” clicar no ícone “+” e “from Exchange”

Sharepoint.com/personal/bruno\_sales\_orangedoor.com.br/\_layouts/15/doc.aspx?sourceid=18d56d495\_e3df\_4eb\_a8a3-daab6d85d6a61&action=...

### New Mule Project

**Project Settings**  
Create a Mule project in the workspace or in an external location.

Project Name:

**Runtime**  
Mule Server 4.4.0 EE  
Mule Server 4.3.0 EE

[Install Runtimes](#)

**API Implementation**  
Add an API implementation to your project to automatically set up an APIkit router and create placeholder flows for each resource method

Import a published API | Import RAML from local file | Download RAML from Design Center

Start building API implementations by importing the specification here. [Learn more](#)

from Exchange | from Maven

Version

Please select or add a dependency to see more information.

☒ Scaffold flows from these API specifications

**Project location**  
☒ Use default location  
Location:  [Browse...](#)

[?](#) [Finish](#) [Cancel](#)

Caso seja solicitado, adicione sua conta da Plataforma Anypoint e clique em OK

User login

Standard authentication mode [Configure](#)

**MuleSoft**

We use ☒ Authentication Method

☒ Anypoint Platform

☐ External identity

Organization domain:

[Consent](#)

[Accept](#) [OK](#)

[Forgot your credentials?](#)  
[Use custom domain](#)

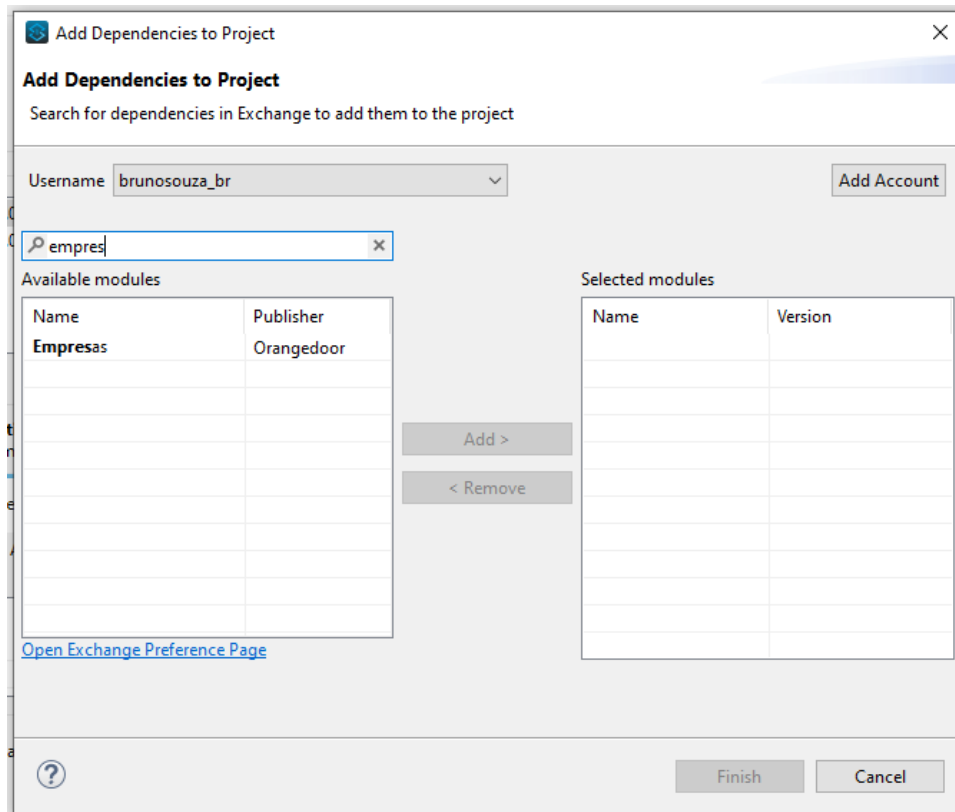
Don't have an account? [Sign up](#)

Login expires after:  days

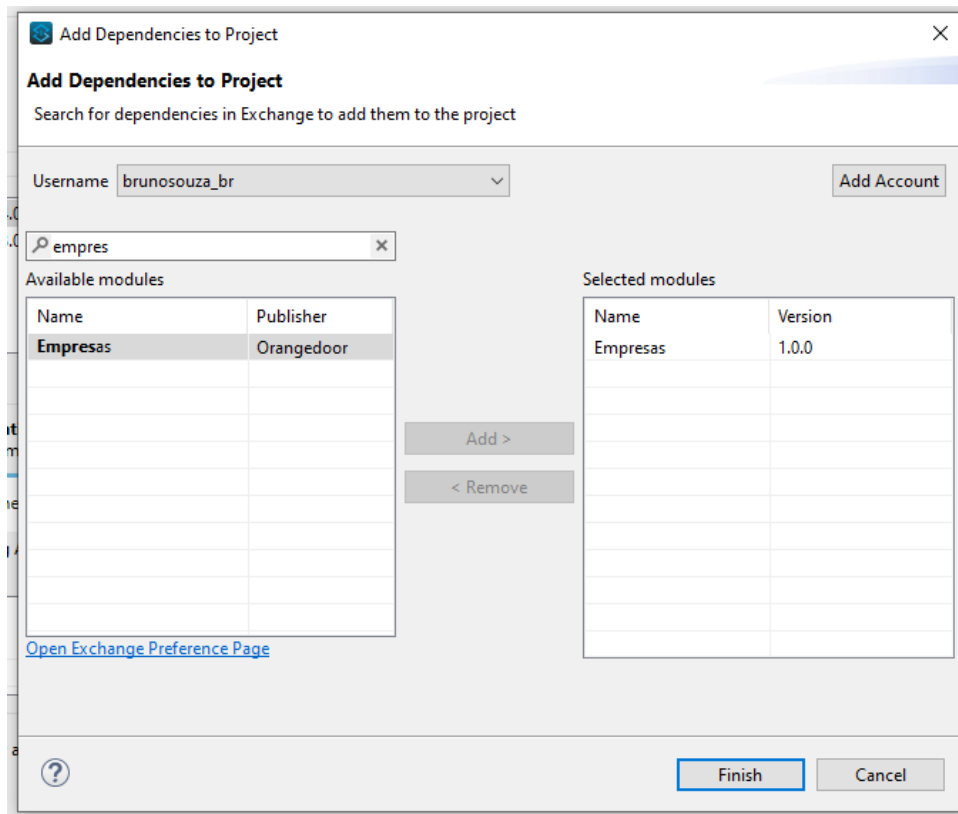
[Cancel](#)

[Account](#)

Realize a pesquisa pelo nome do RAML publicado no Exchange

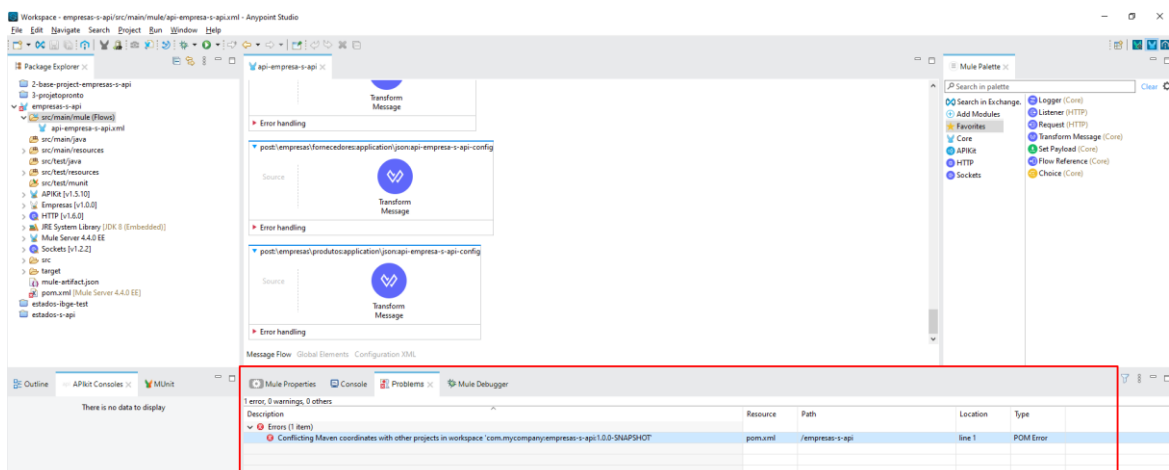


Clique em Add / Finish



Neste momento são importadas todas as informações referentes ao contrato da API publicado no Exchange e montada uma estrutura de flows com os componentes que se fazem necessários.

Notem que ocorreu um erro em nossa API, onde podemos ver mais detalhes na aba “Problems”



Neste caso identificou um erro no arquivo POM.xml

Maven



O Maven é uma ferramenta de código aberto mantida pela Apache. Trata-se de uma ferramenta de gestão de dependências, o Maven automatiza os processos de obtenção de dependências e de compilação de projetos Java.

Ele também é um utilitário de gerenciamento de projetos que o Mule implementa para aprimorar o desenvolvimento.

Quando criamos um projeto Maven, este projeto fica atrelado a um arquivo principal: o pom.xml. Neste arquivo POM (Project Object Model), nós descrevemos as dependências de nosso projeto e a maneira como este deve ser compilado. Com o Maven, é possível, por exemplo, automatizar a execução de testes unitários durante a fase de build, entre outras automatizações.

Para mais referências do Maven consulte: <https://maven.apache.org/guides/getting-started/>

Como agora que sabemos que nosso arquivo POM.xml é responsável por criar todas as dependências que estamos utilizando em nossa aplicação, vamos verificar a estrutura dele e também realizar a correção

```
api-empresa-s-api pom.xml
1 <?xml version="1.0" encoding="UTF-8"?>
20 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4
5   <groupId>com.mycompany</groupId>
6   <artifactId>empresas-s-api</artifactId>
7   <version>1.0.0-SNAPSHOT</version>
8   <packaging>mule-application</packaging>
9
10  <name>empresas-s-api</name>
11
12  <properties>
13    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
14    <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
15
16    <app.runtime>4.4.0-20220221</app.runtime>
17    <mule.maven.plugin.version>3.3.2</mule.maven.plugin.version>
18  </properties>
19
20  <build>
21    <plugins>
22      <plugin>
23        <groupId>org.apache.maven.plugins</groupId>
24        <artifactId>maven-clean-plugin</artifactId>
25        <version>3.0.0</version>
26      </plugin>
27      <plugin>
28        <groupId>org.mule.tools.maven</groupId>
29        <artifactId>mule-maven-plugin</artifactId>
30        <version>${mule.maven.plugin.version}</version>
31        <extensions>true</extensions>
32      </plugin>
33    </plugins>
34  </build>
35
36  <dependencies>
37    <dependency>
38      <groupId>org.mule.connectors</groupId>
39      <artifactId>mule-http-connector</artifactId>
40      <version>1.6.0</version>
41      <classifier>mule-plugin</classifier>
42    </dependency>
43    <dependency>
44      <groupId>org.mule.connectors</groupId>
45      <artifactId>mule-sockets-connector</artifactId>
46      <version>1.2.2</version>
47      <classifier>mule-plugin</classifier>
48    </dependency>
49    <dependency>
50      <groupId>org.mule.modules</groupId>
51      <artifactId>empresas-s-api</artifactId>
52      <version>1.0.0</version>
53      <classifier>api</classifier>
54      <type>zip</type>
55    </dependency>
56  </dependencies>
```

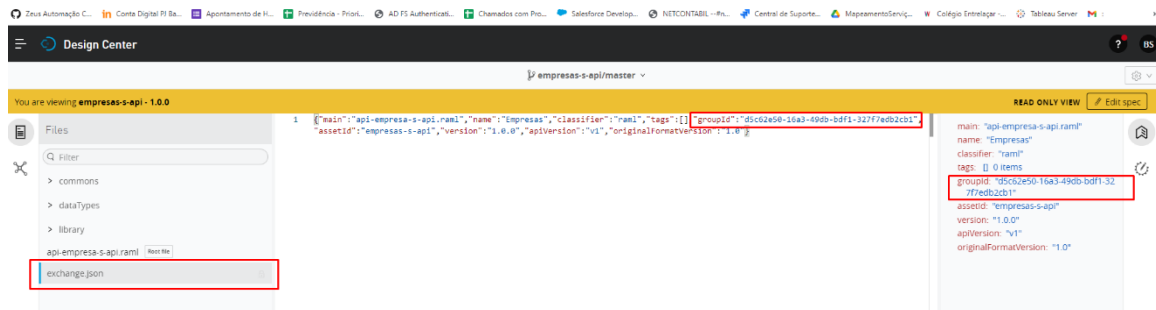
Ponto em que está gerando o problema:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/
  <modelVersion>4.0.0</modelVersion>

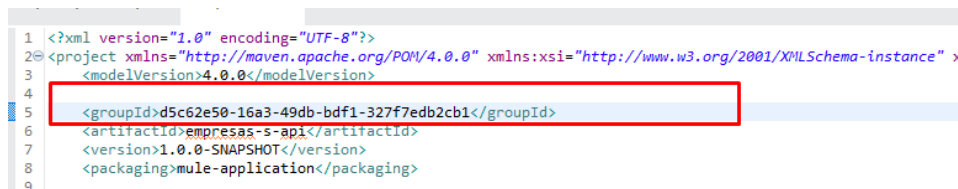
  <groupId>com.mycompany</groupId>
  <artifactId>empresas-s-api</artifactId>
  <version>1.0.0-SNAPSHOT</version>
  <packaging>mule-application</packaging>
```

Este groupId está incorreto, precisamos voltar ao RAML para verificar qual é o correto

Na Plataforma Anypoint vá até o Design Center e verifique o groupId no arquivo “exchange.json”

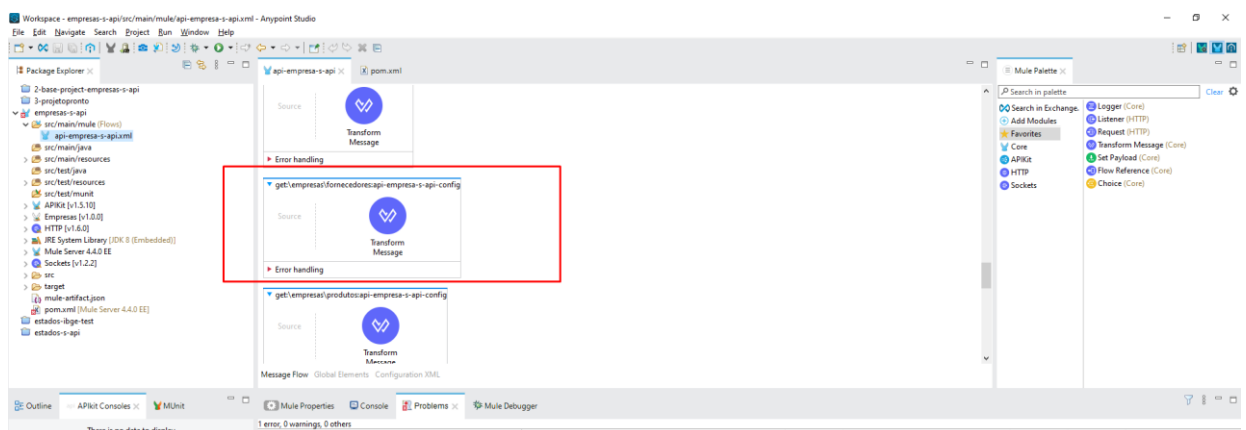


Copie o valor do groupId e altere no arquivo POM.xml

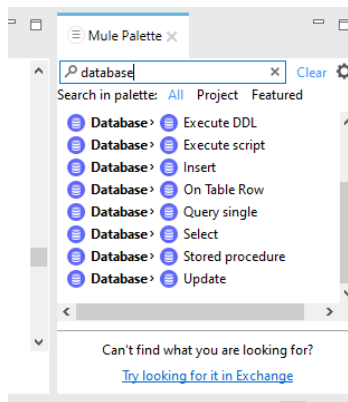
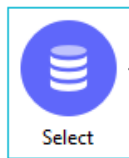


Iremos trabalhar agora no fluxo de consulta fornecedores

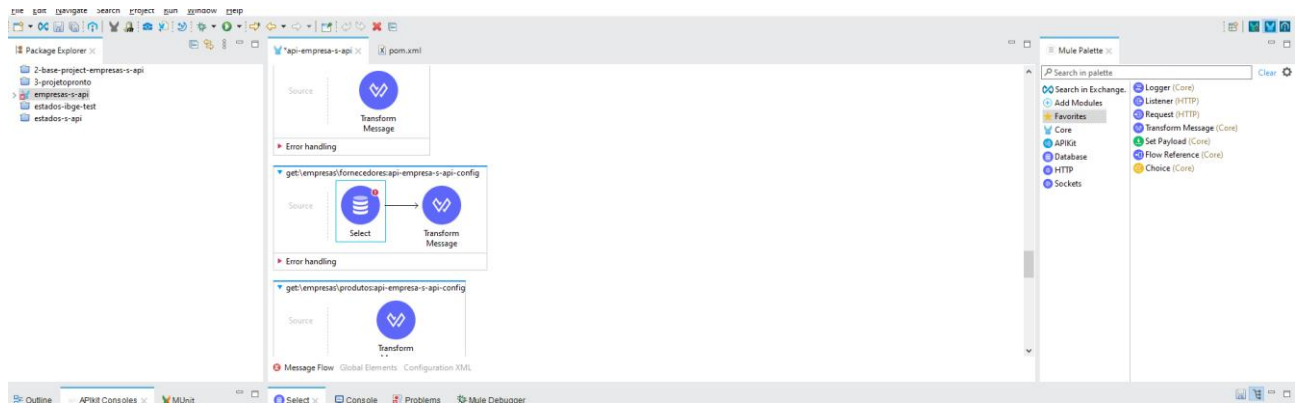
- Resource: get:\empresas\fornecedores:api-empresa-s-api-config



**Componente Database Select:** Componente de conexão com o BD, responsável por realizar consultas.



Incluir o componente Database Select no fluxo de consulta fornecedores



No arquivo POM.xml, precisamos incluir a dependência para podermos conectar no banco MySQL, onde estão os dados que iremos trabalhar.

```
*api-empresa-s-api pom.xml x
27<plugin>
28  <groupId>org.mule.tools.maven</groupId>
29  <artifactId>mule-maven-plugin</artifactId>
30  <version>${mule.maven.plugin.version}</version>
31  <extensions>true</extensions>
32</plugin>
33</plugins>
34</build>
35
36<dependencies>
37  <dependency>
38    <groupId>mysql</groupId>
39    <artifactId>mysql-connector-java</artifactId>
40    <version>8.0.30</version>
41  </dependency>
42  <dependency>
43    <groupId>org.mule.connectors</groupId>
44    <artifactId>mule-http-connector</artifactId>
45    <version>1.6.0</version>
46    <classifier>mule-plugin</classifier>
47  </dependency>
48  <dependency>
49    <groupId>org.mule.connectors</groupId>
50    <artifactId>mule-sockets-connector</artifactId>
51    <version>1.2.2</version>
52    <classifier>mule-plugin</classifier>
53  </dependency>
</dependencies>
```

```
<dependency>
<groupId>mysql</groupId>
<artifactId>mysql-connector-java</artifactId>
<version>8.0.30</version>
</dependency>
```

Acrescentar nas propriedades do componente Select os seguintes dados

Global Element Properties

Database Config

Default configuration

General Advanced Notes Help

Name: Database\_Config

Connection: Generic Connection

General Transactions Advanced

Required Libraries

JDBC Driver Configure...

Connection

URL: jdbc:mysql://localhost:3306/dbEmpresa

Driver class name: com.mysql.cj.jdbc.Driver

User: root

Password: Show password

Test Connection... OK Cancel

Connection: Generic Conection

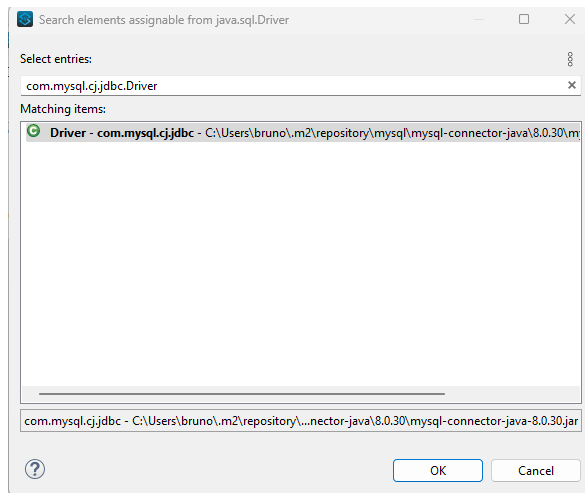
URL: *jdbc:mysql://localhost/dbEmpresa*

Driver class name: com.mysql.cj.jdbc.Driver (clique na lupa e pesquisar)

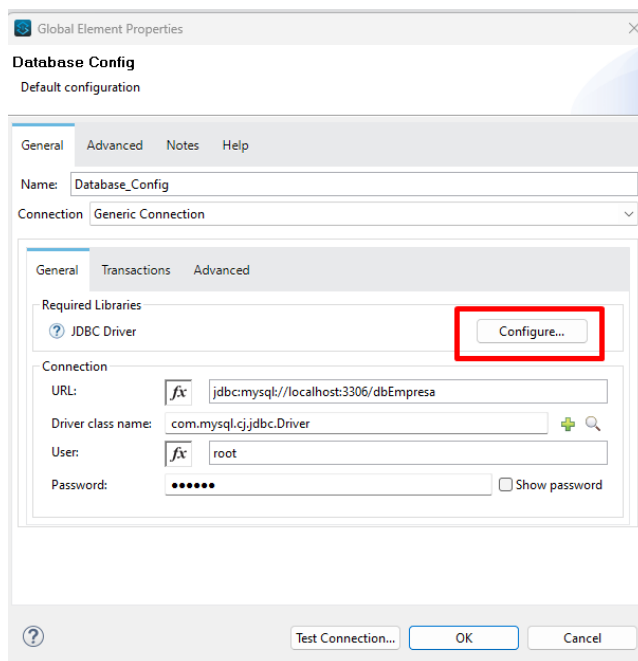
User: orange

Password: Orange@2022

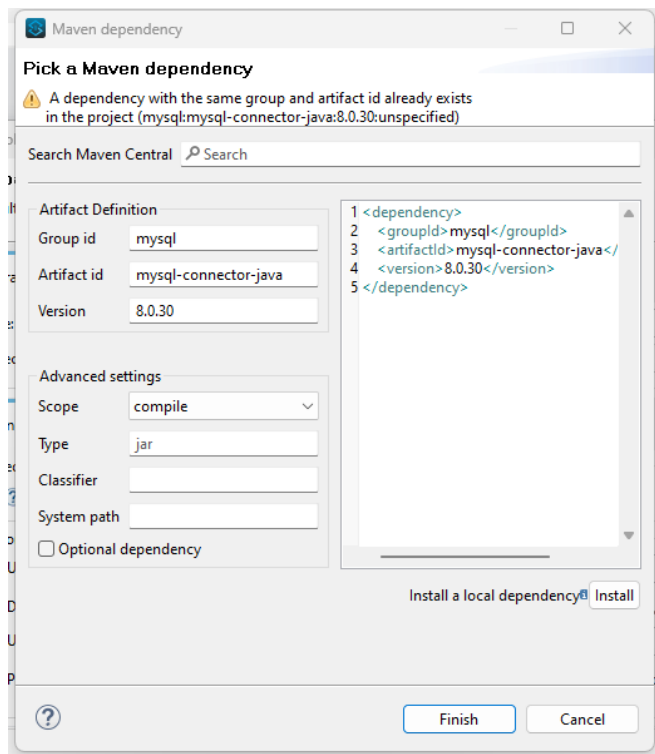
Clique em test connection.



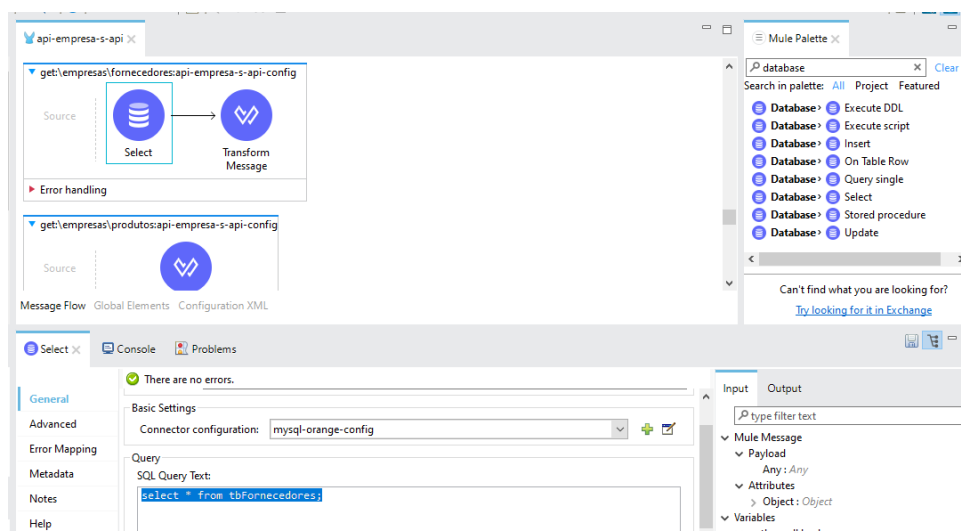
Clique em “Configure” para add as dependência do Maven



Digite as infos a seguir e clique em finish:



No campo SQL Query Text, digitar o select abaixo:  
`select * from tbFornecedores;`



Agora vamos testar nossa aplicação depois aplicar o mesmo para os fluxos de GET Produtos e GET Clientes.