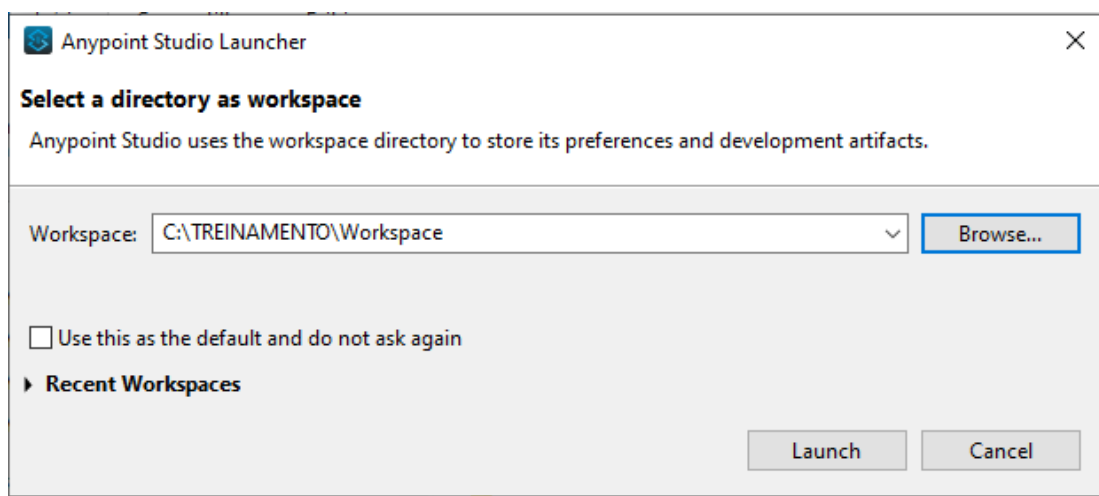


Aula 02

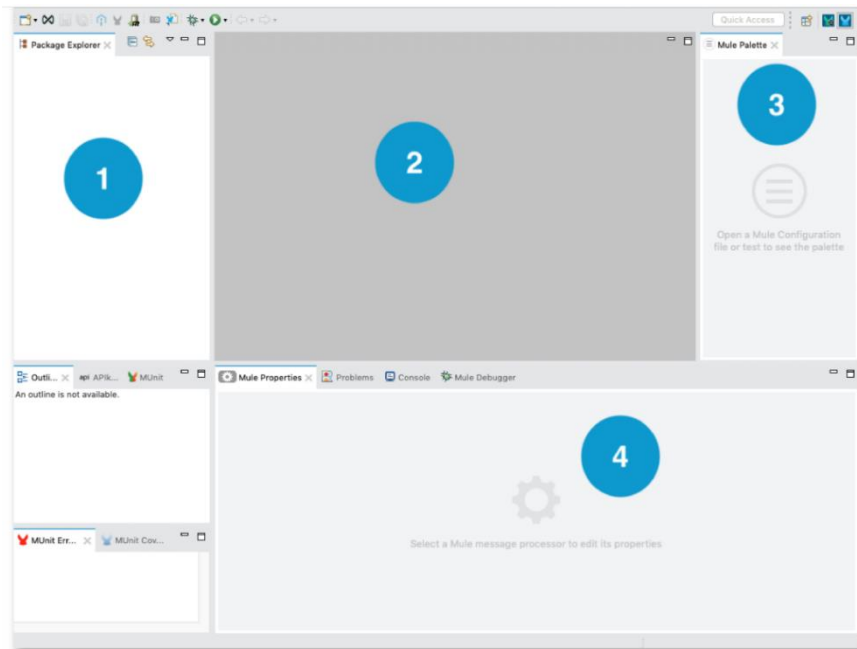
Iniciando o Anypoint Studio

Ao iniciar o Anypoint Studio, é solicitado que você escolha uma Workspace (espaço de trabalho), para deixar padronizado, crie no diretório raiz “c:\TREINAMENTO\Workspace”.

Será dentro deste diretório que todos os seus projetos Mule serão gravados.



Será aberta a seguinte janela da nossa IDE de desenvolvimento:



1 - O Package Explorer ajuda você a navegar pelos arquivos criados para cada projeto.

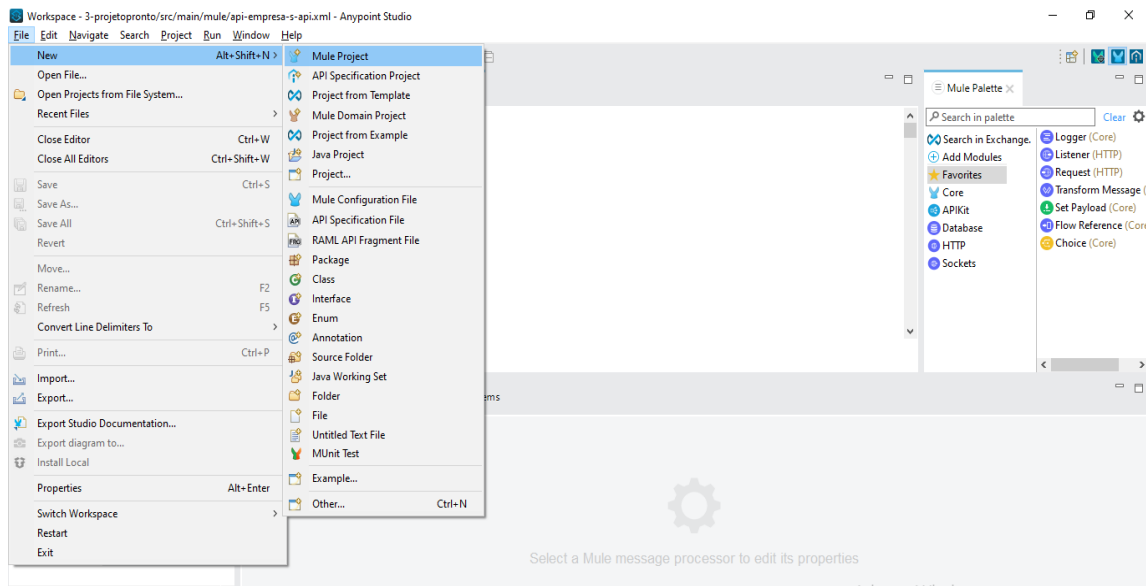
2 - Canvas tela é onde você arrasta e solta elementos do seu projeto e os conecta, criando um fluxo.

3 - A Mule Palette fornece acesso rápido aos componentes padrão ou personalizados.

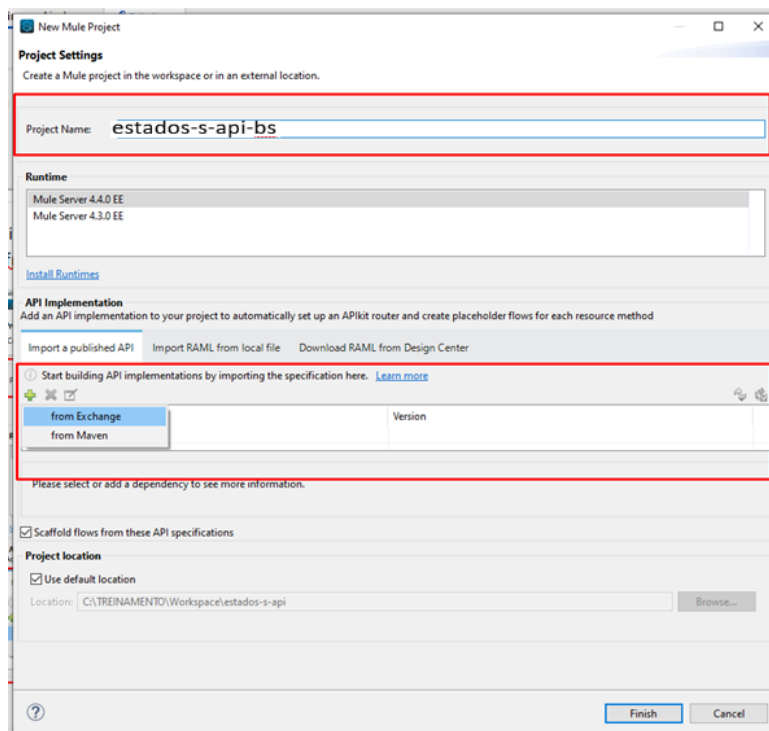
4 - As guias na seção inferior fornecem acesso às caixas de diálogo de configuração, ao depurador Mule e a outras caixas de diálogo do desenvolvedor.

Agora vamos criar nosso primeiro projeto usando o Anypoint Studio:

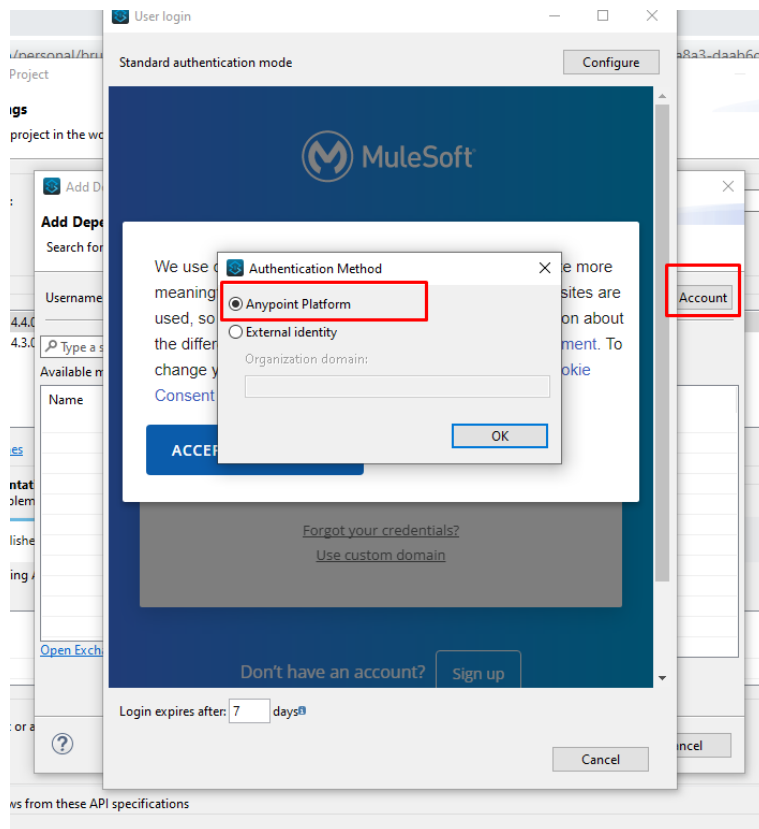
Clique no menu File / New / Mule Project



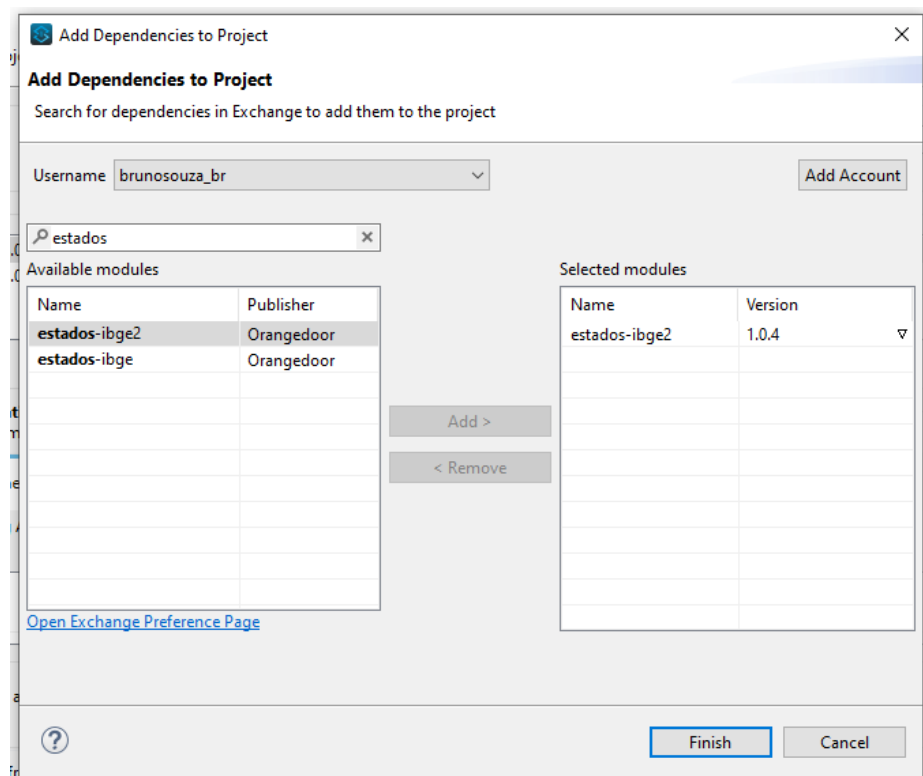
Digite o nome do projeto “estados-s-api-bs”, na aba “Import a Published API” clicar no ícone “+” e “from Exchange”



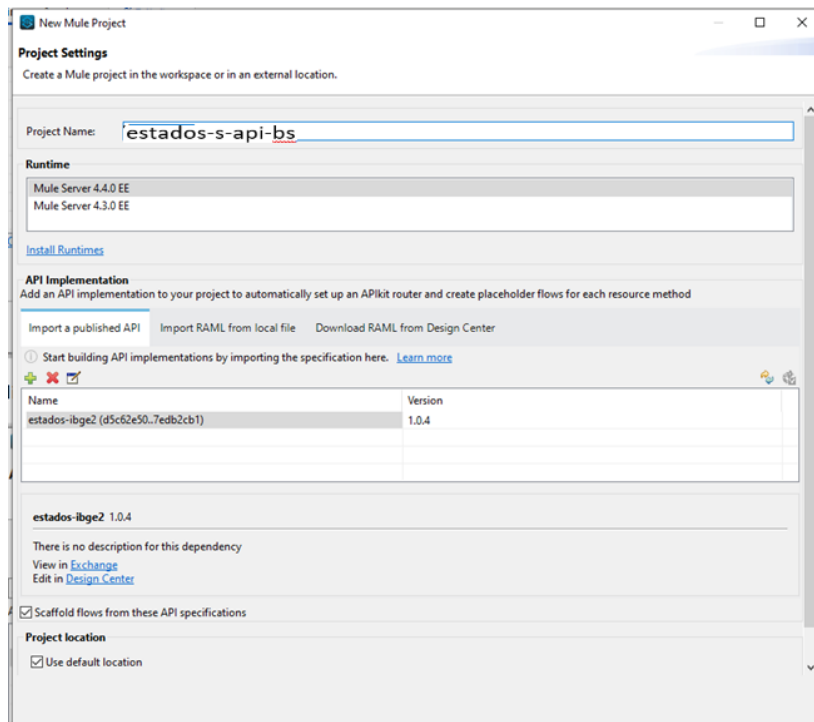
Adicione sua conta da Plataforma Anypoint e clique em OK



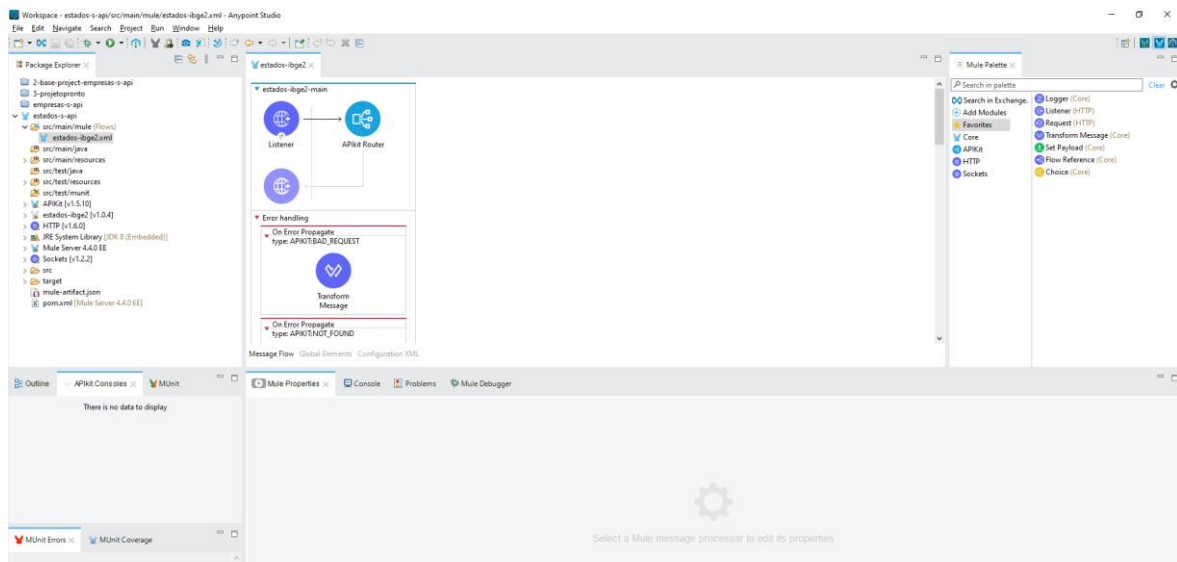
Realize a pesquisa pelo nome do RAML publicado no Exchange



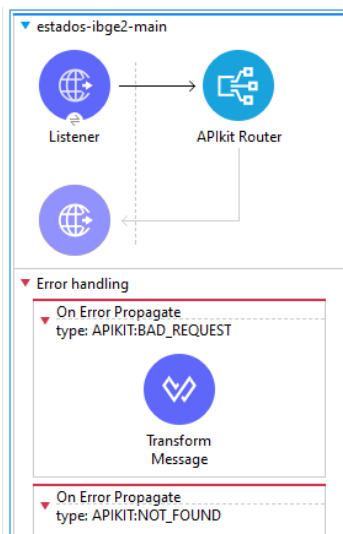
Clique em Add / Finish



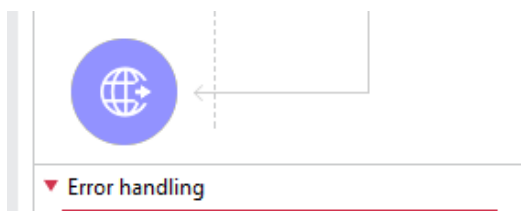
Neste momento são importadas todas as informações referentes ao contrato da API publicado no Exchange e montada uma estrutura de flows e subflows com os componentes que se fazem necessários.



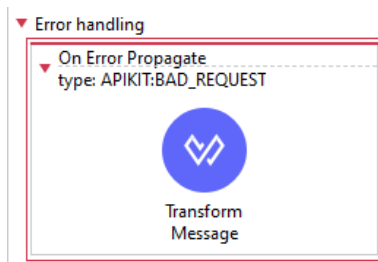
Flows e Subflows: São estruturas de fluxos e subfluxos onde os componentes são inseridos para o tráfego de mensagens ou eventos Mule.



Error handling: Área existente em todos os flows onde serão definidas as manipulações de tratativas de erros.



On Error Propagate: Componente responsável por realizar a tratativa de erros quando acionado pelo error handling.



HTTP Listener connector: Componente responsável por “escutar” solicitações HTTP. Você pode escolher quais métodos. O conector aceita: GET, POST etc.



APIkit Router: Componente responsável por definir qual rota (fluxo) será seguida na API, conforme especificado no RAML.



APIkit Router

APIkit Console: Componente responsável por fornecer com console HTML interativo para a interação com a API, conforme especificado pelo RAML



APIkit Console

Transform Message: Componente responsável por converter (transformar) os dados de entrada em uma nova estrutura ou formato de saída. Estas transformações realizamos através da linguagem DataWeave.



Transform
Message

Logger: Componente responsável por gerar logs.



Logger

HTTP Request: Componente responsável por consumir serviços HTTP, podendo selecionar qual método irá utilizar (GET, POST etc.). Onde pode-se também incluir corpo (body), cabeçalhos (header), e parâmetros para as consultas.



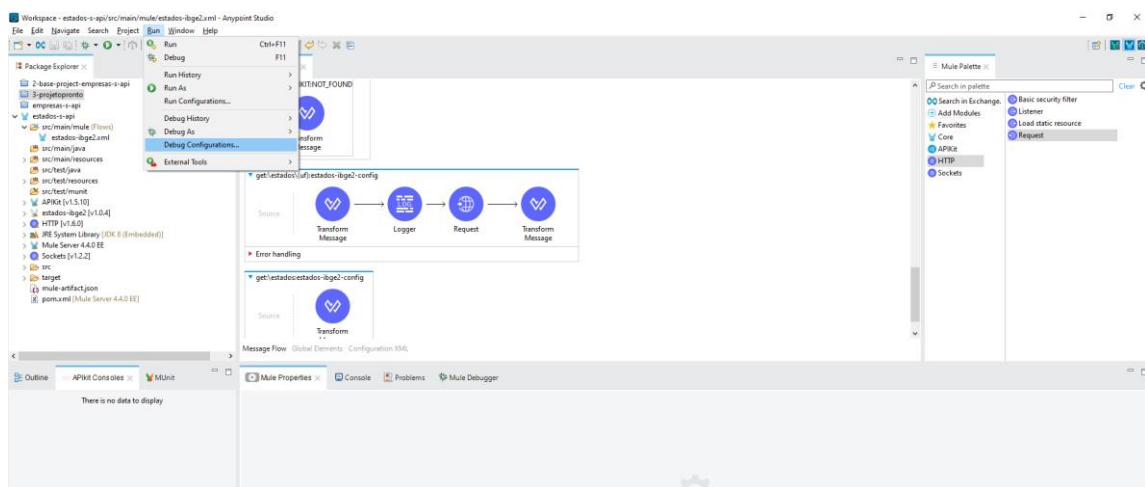
Request

Agora vamos configurar nossa aplicação para que seja realizada a chamada na API do IBGE.

Testando API de Estados

Neste momento iremos testar localmente nossa API desenvolvida através do “Debug”

Clique no Menu Run / Debug Configurations

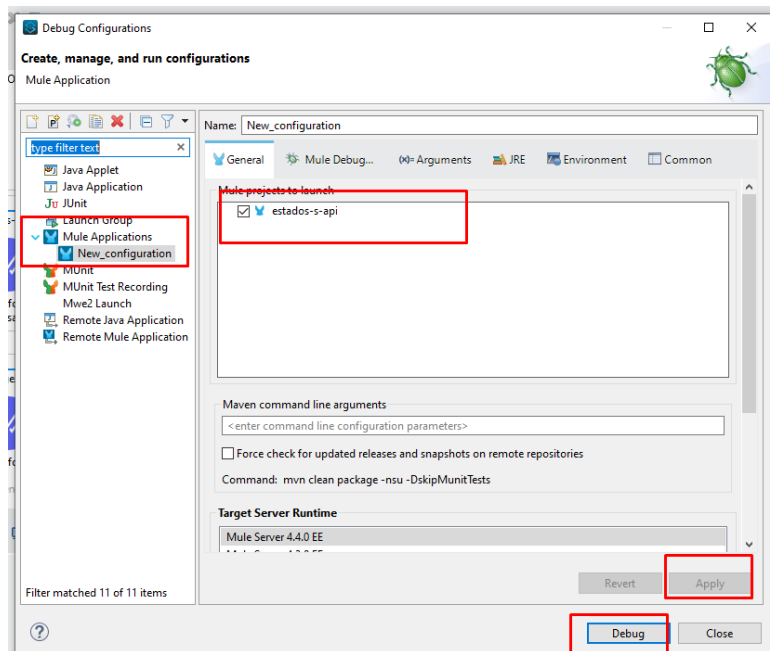


Existe diferenças importantes entre RUN e Debug nesta etapa.

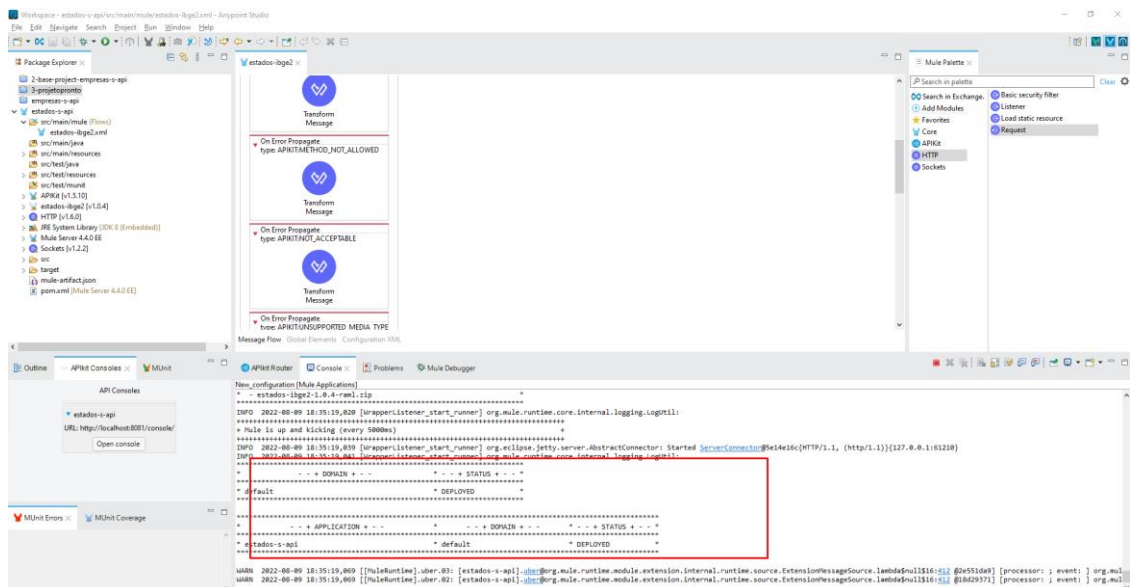
- RUN – Você está colocando sua aplicação para rodar. Sem depuração e sem acompanhar os bastidores dos dados.
- Debug – Aqui sua aplicação está em teste. É possível parar o processamento através de **breakpoints** e acompanhar o status do passo a passo, observando o que ela está retornando no meio do processo.

Clique em “Mule Applications” / “New configuration” (lado esquerdo)

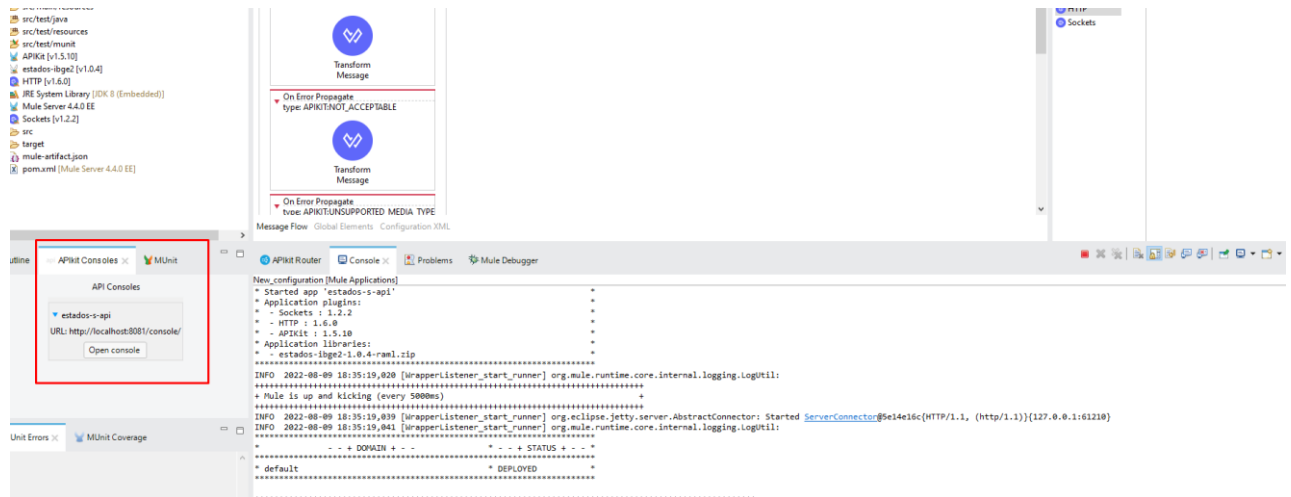
Selecione a API que deseja realizar o debug / clique em Apply / Debug



Caso seu desenvolvimento esteja correto, será exibida na console do Anypoint Studio “Deployed”



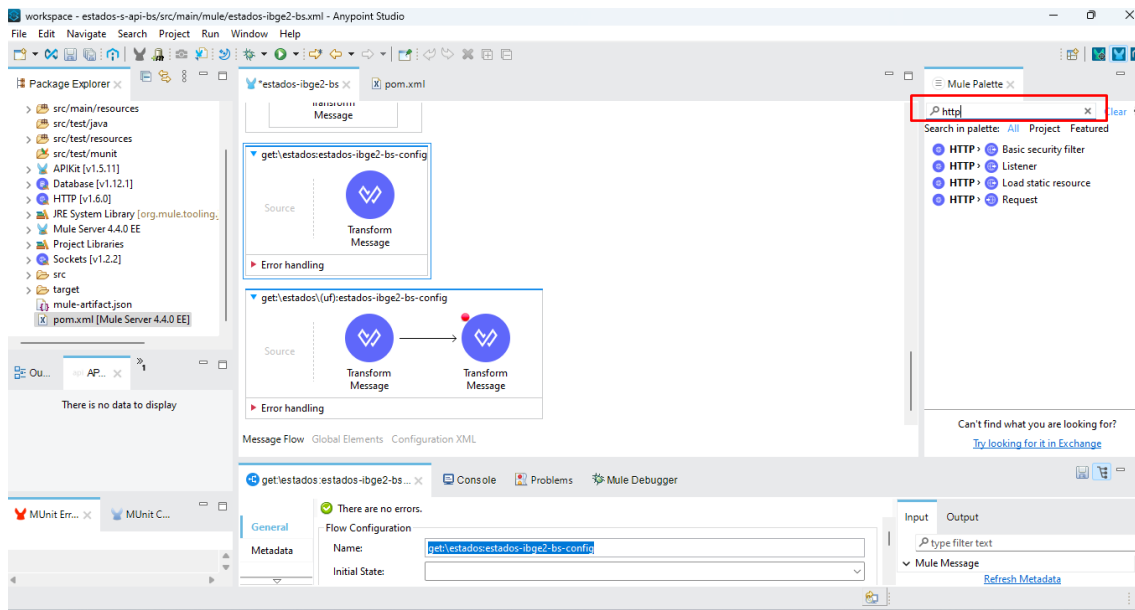
Note que foi criado na aba APIkit consoles a opção para que seja aberta uma console rodando localmente, clicando no botão “Open console”



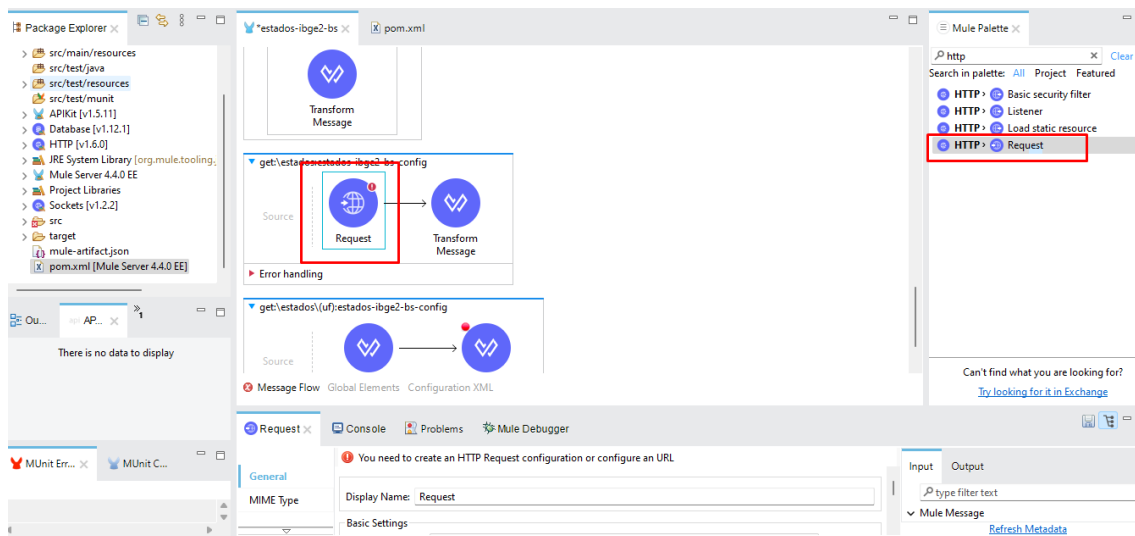
Neste momento iremos melhorar o funcionamento de nossa API, onde faremos que ela consuma os endpoints disponibilizados pelo IBGE e nos retorne as informações corretamente.

No primeiro fluxo (flow) “get:\estados:estados-ibge2-bs-config”, iremos incluir o componente HHT Request que será responsável em realizar o consumo da API do IBGE.

Na mule palette digite na pesquisa “http”

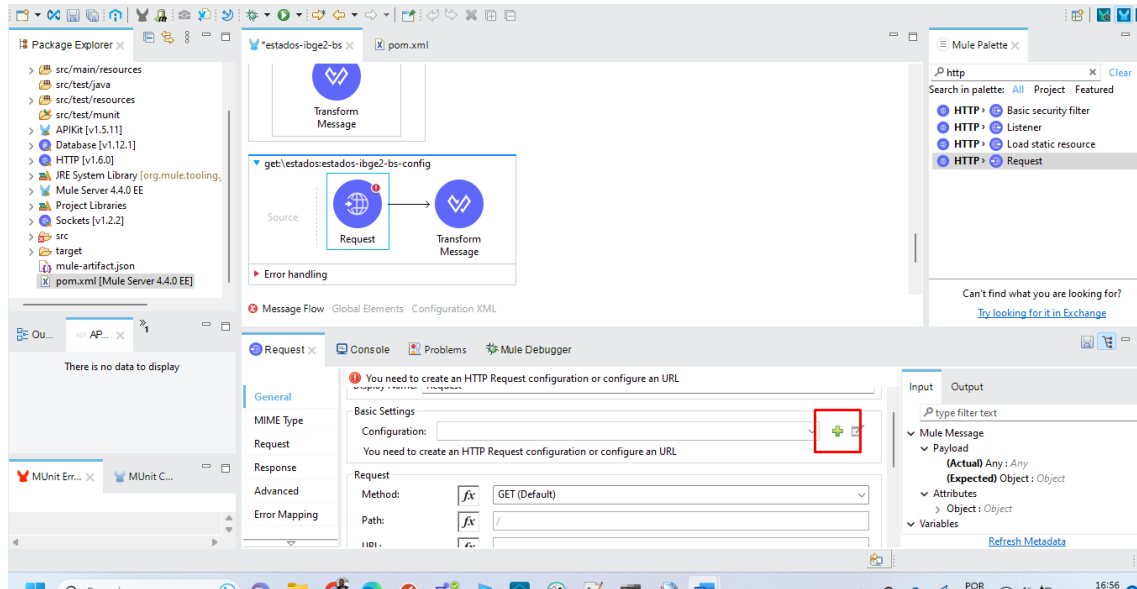


Clique no componente “HTTP Request” e arraste até o flow deixando-o como primeiro componente



Agora iremos alterar as configurações deste componente conforme abaixo:

Clique no sinal de “+” para adicionar uma nova configuração



Inclua as seguintes informações:

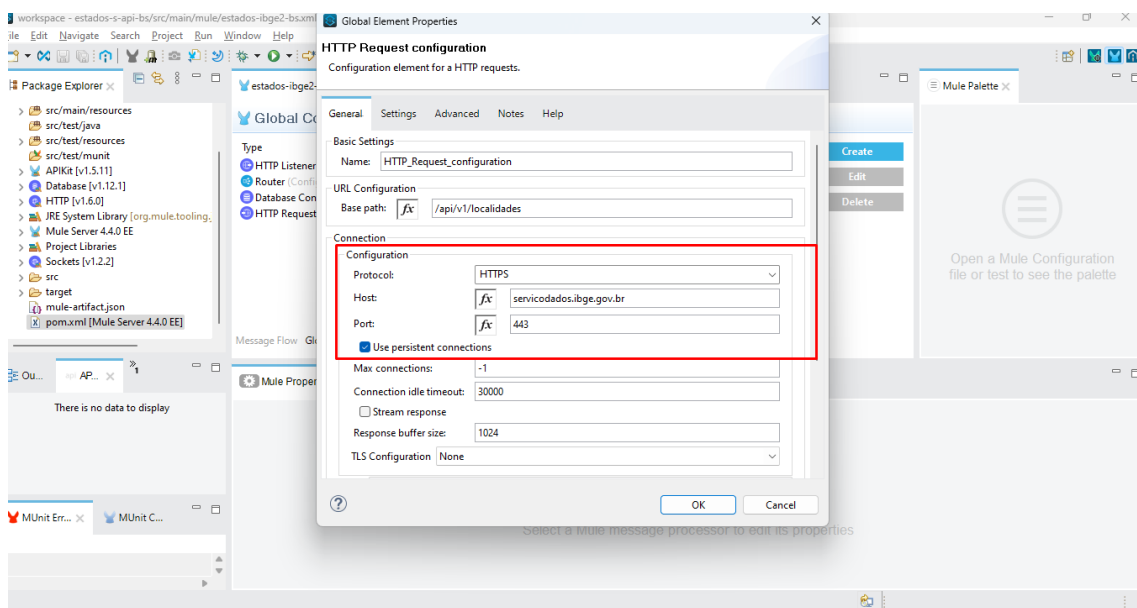
Protocol: HTTPS

host: servicodados.ibge.gov.br/api/v1/localidades

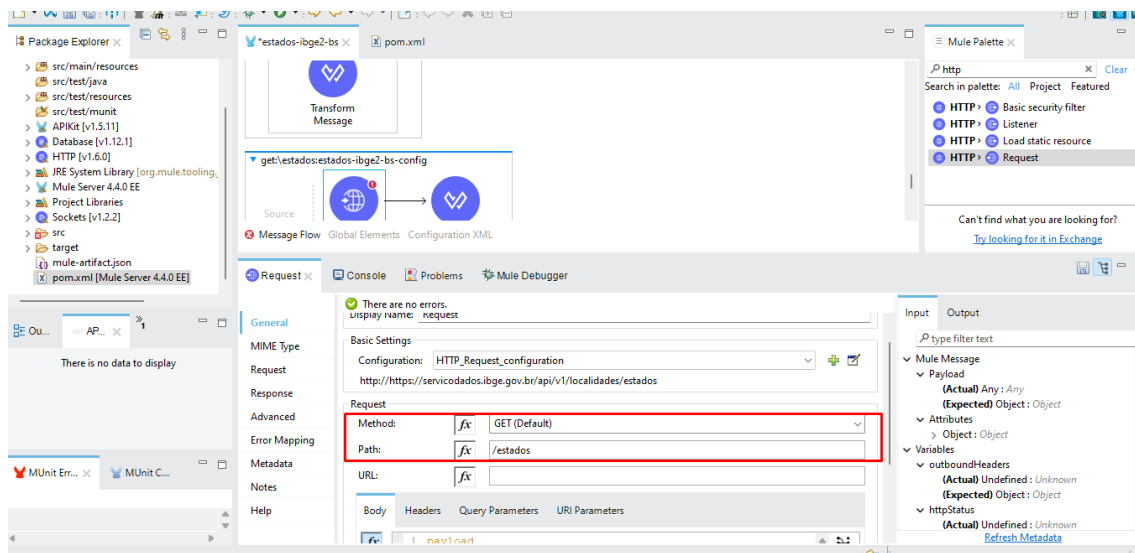
Base path: /api/v1/localidades

Port: 443

e clique em OK

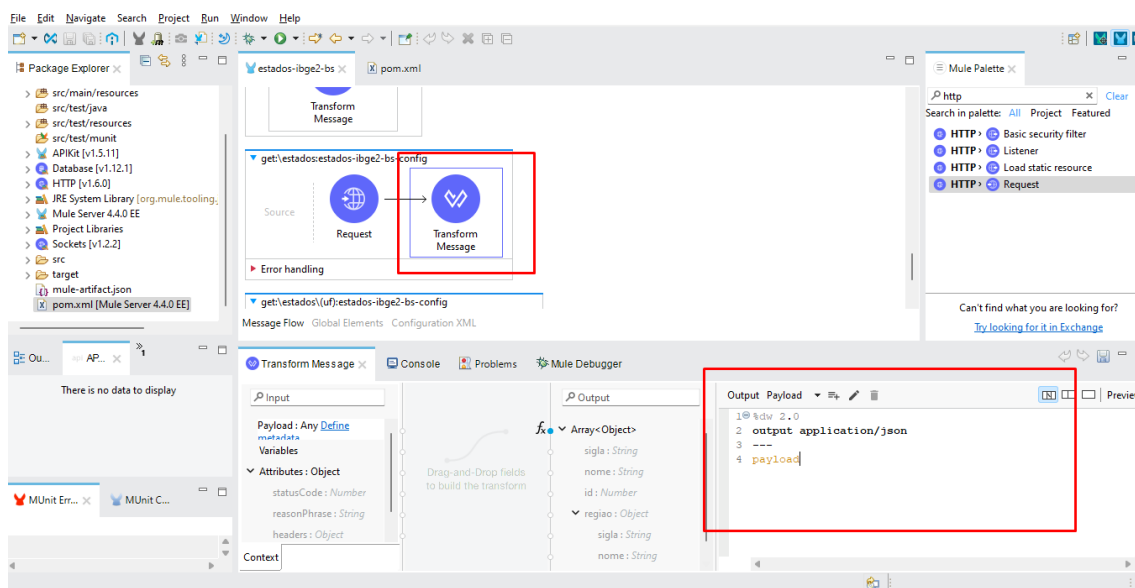


No campo “Method” deve estar como “GET”, pois iremos realizar uma consulta no IBGE e no campo “Path” inclua “/estados” (complemento do endpoint do IBGE)



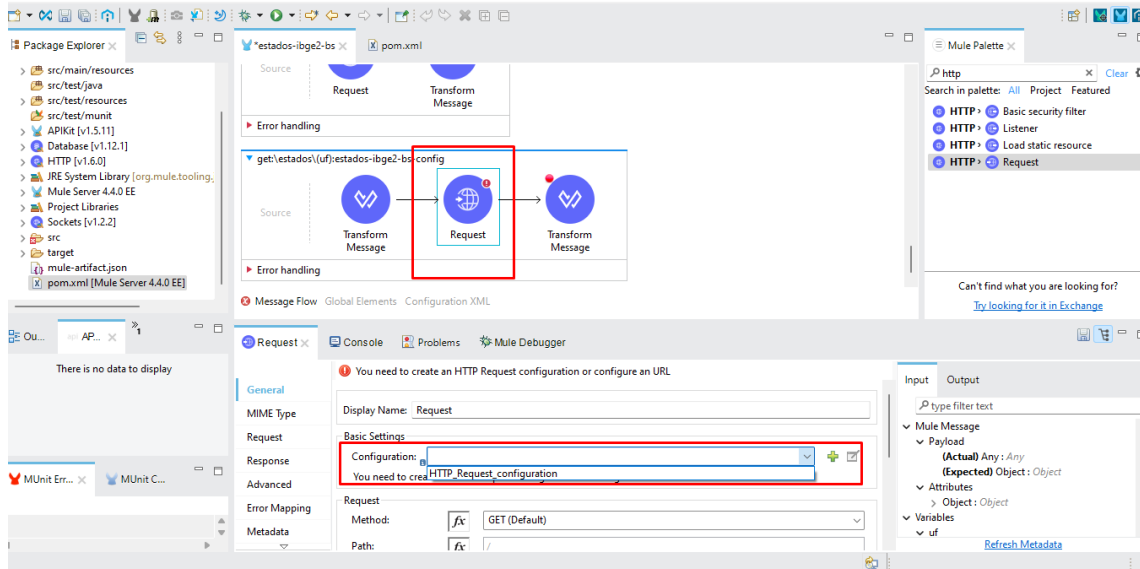
No componente “Transform Message” iremos alterar o código dataweave para o seguinte:

```
%dw 2.0
output application/json
---
Payload
```



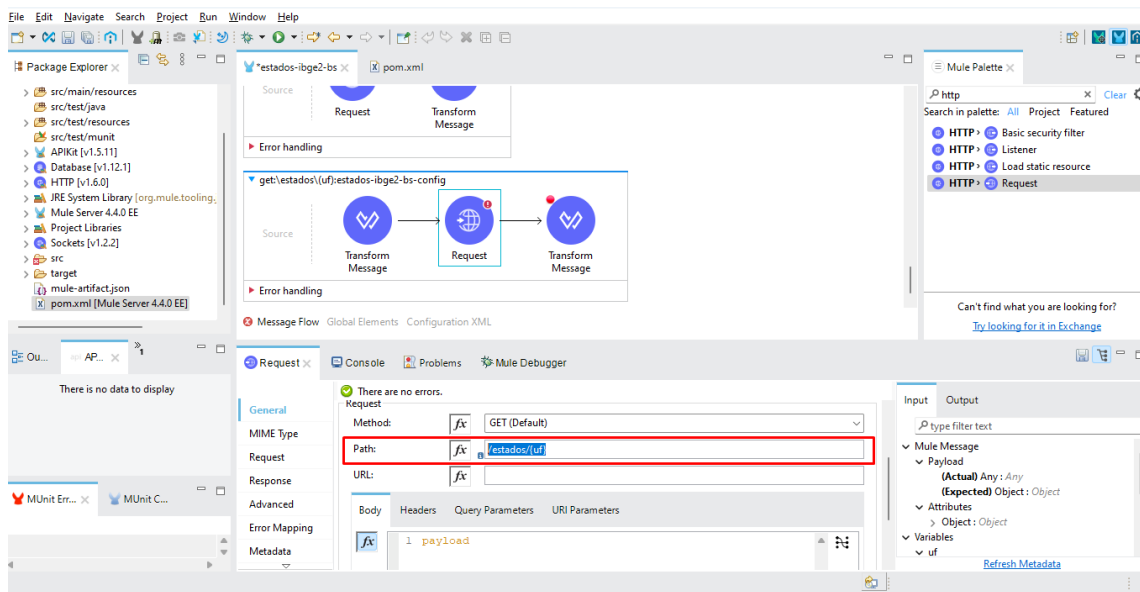
Agora rode o Debug e realize a chamada para verificar o resultado.

No segundo fluxo (flow), iremos incluir um outro componente “HTTP Request” e iremos reaproveitar as mesmas configurações já realizados no mesmo que inserimos no primeiro fluxo.



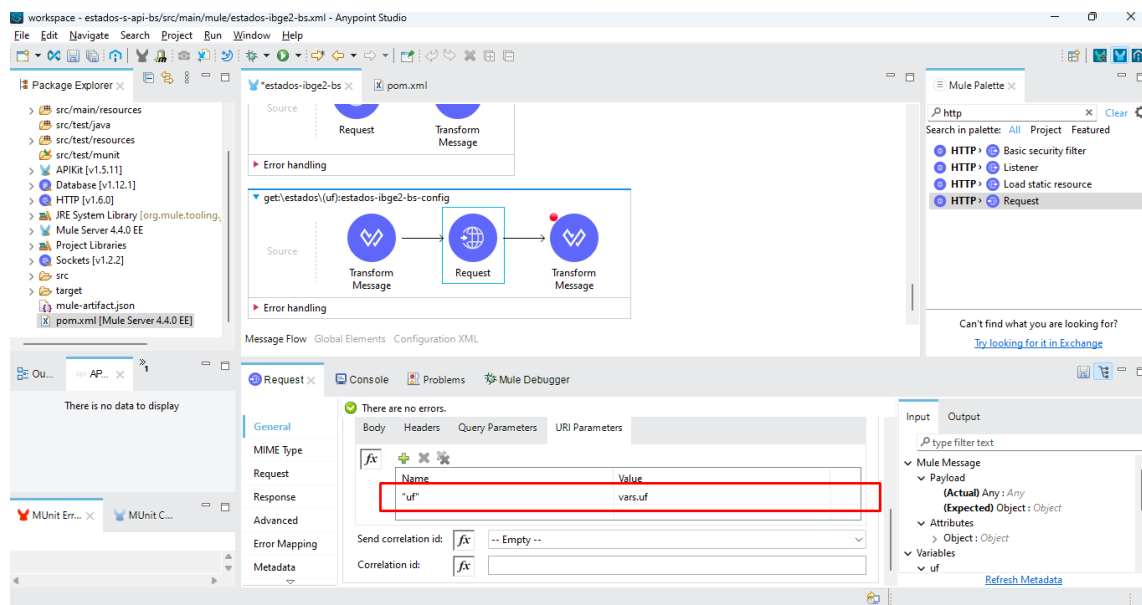
Nas propriedades deste HTTP Request iremos incluir os dados abaixo:

Path: /estados/{uf}

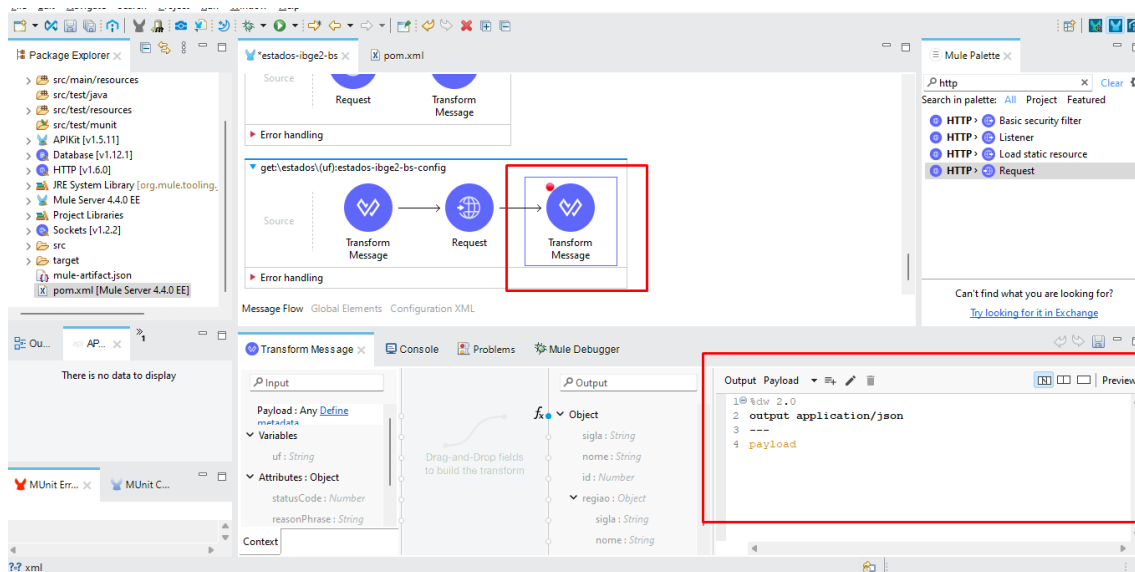


Ainda nas propriedades do HTTP Request, clique na aba “URI Parameters” para que seja incluída a informação da UF que foi passada na chamada da API.

Clique no botão “+” e inclua a nova chave e valor (variável coletada no primeiro transform)



No componente “Transform Message” final, inclua o mesmo código dataweave do último transform message do primeiro fluxo



Rode a aplicação em modo debug e veja o resultado final realizando a chamada do endpoint.

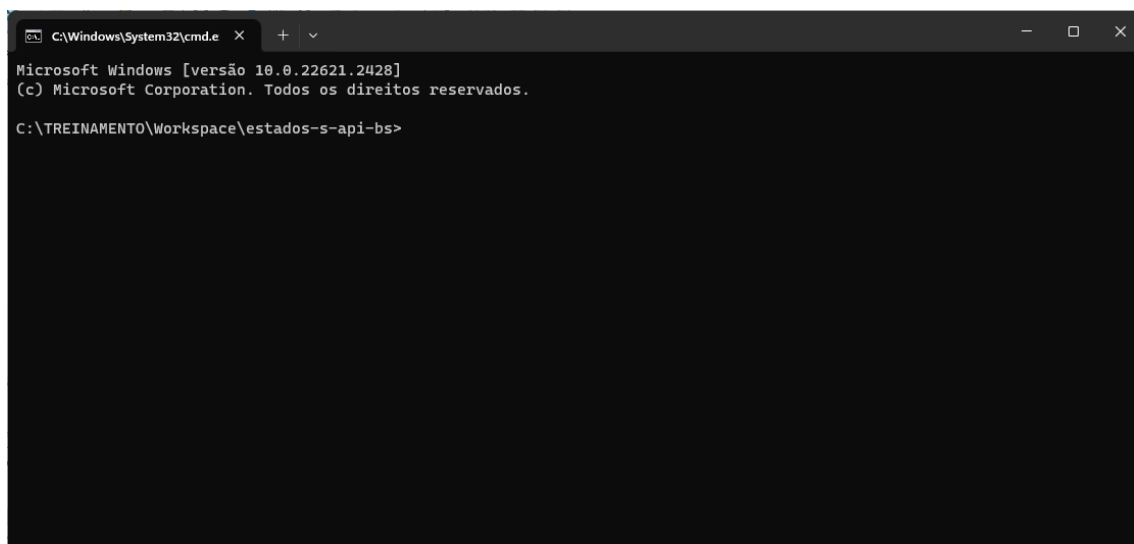
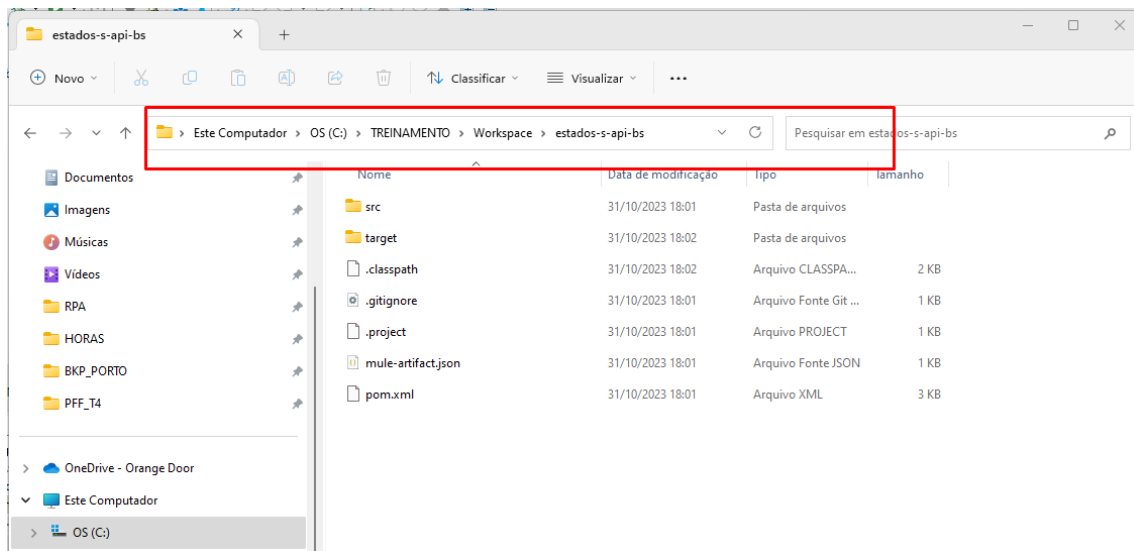
Utilize os “breakpoints” e realize o debug passo a passo para ver o funcionamento da API.

Gerando pacote para a implantação

Após o término de nosso desenvolvimento, iremos agora gerar um pacote para que possamos enfim realizar a o deploy (implantação) de nossa API no Anypoint Studio.

Utilizaremos a ferramenta Maven para realizar o build de nossa aplicação.

Para isto iremos abrir o prompt de comando do DOS, na pasta em que está salvo o nosso projeto “estados-s-api-bs”



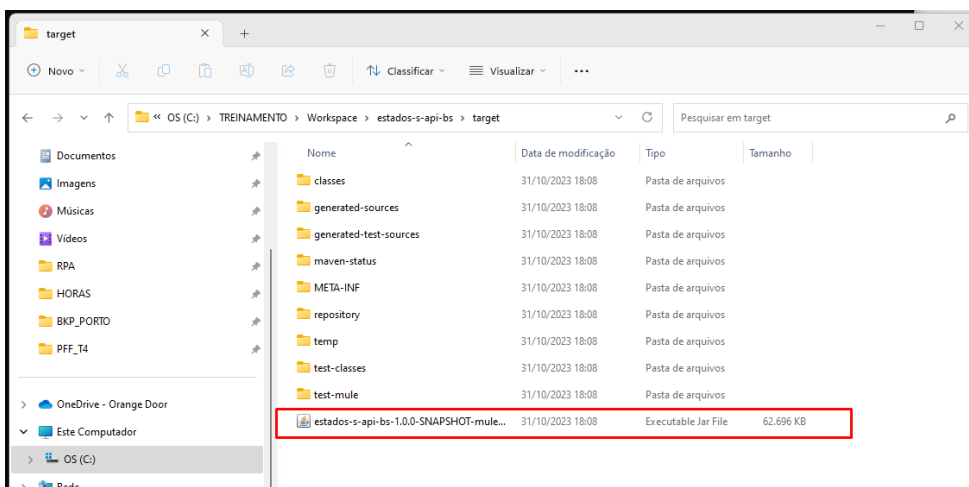
Digite o comando “mvn clean package” e pressione enter
Este comando é responsável por compilar e gerar um
arquivo “.jar” da sua aplicação

```
C:\Windows\System32\cmd.e x + v
C:\TREINAMENTO\Workspace\estados-s-api-bs>mvn clean package
```

Caso ocorra tudo certo, a mensagem “BUILD SUCCESS” será exibida na tela

```
C:\Windows\System32\cmd.e x + v
[INFO] --- mule-maven-plugin:3.5.4:compile (default-compile) @ estados-s-api-bs ---
[INFO] --- mule-maven-plugin:3.5.4:process-classes (default-process-classes) @ estados-s-api-bs ---
[INFO] --- mule-maven-plugin:3.5.4:generate-test-sources (default-generate-test-sources) @ estados-s-api-bs ---
[INFO] --- maven-resources-plugin:3.0.2:testResources (default-testResources) @ estados-s-api-bs ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 1 resource
[INFO] --- mule-maven-plugin:3.5.4:generate-test-resources (default-generate-test-resources) @ estados-s-api-bs ---
[INFO] --- maven-compiler-plugin:3.8.1:testCompile (default-testCompile) @ estados-s-api-bs ---
[INFO] Nothing to compile - all classes are up to date
[INFO] --- mule-maven-plugin:3.5.4:test-compile (default-test-compile) @ estados-s-api-bs ---
[INFO] --- maven-surefire-plugin:2.19.1:test (default-test) @ estados-s-api-bs ---
[INFO] --- mule-maven-plugin:3.5.4:package (default-package) @ estados-s-api-bs ---
[INFO] Building zip: C:\TREINAMENTO\Workspace\estados-s-api-bs\target\estados-s-api-bs-1.0.0-SNAPSHOT-mule-application.jar
[INFO] BUILD SUCCESS
[INFO] Total time: 7.747 s
[INFO] Finished at: 2023-10-31T18:08:15-03:00
[INFO]
```

O arquivo “.jar” está gerado na pasta “target” do seu projeto

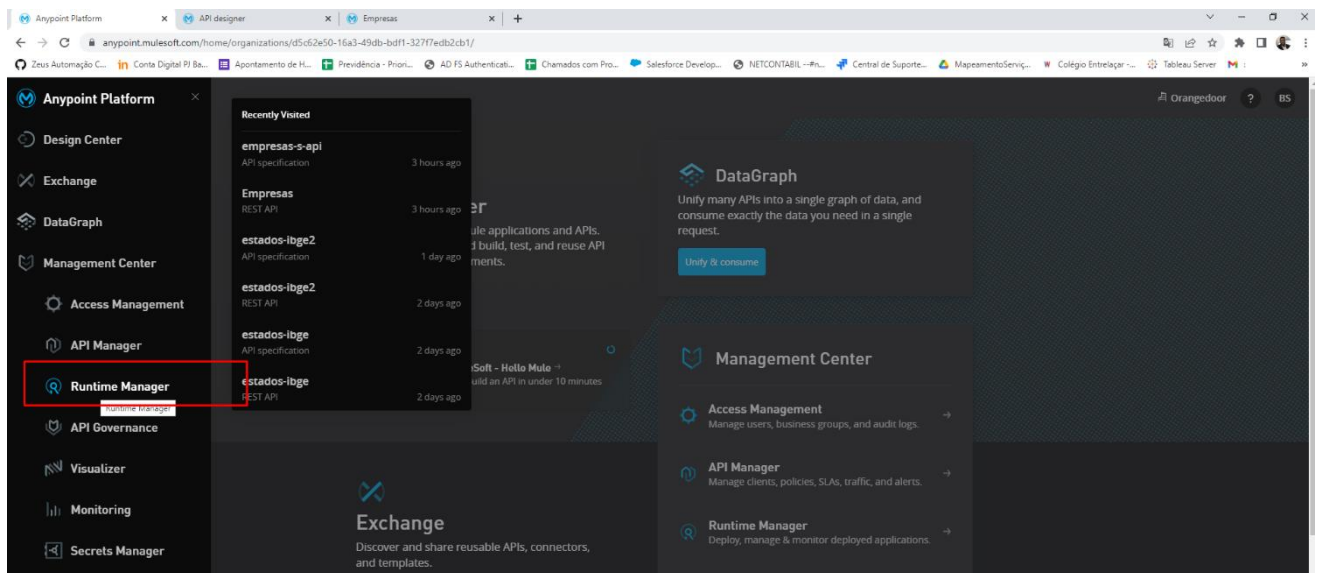


Runtime Manager

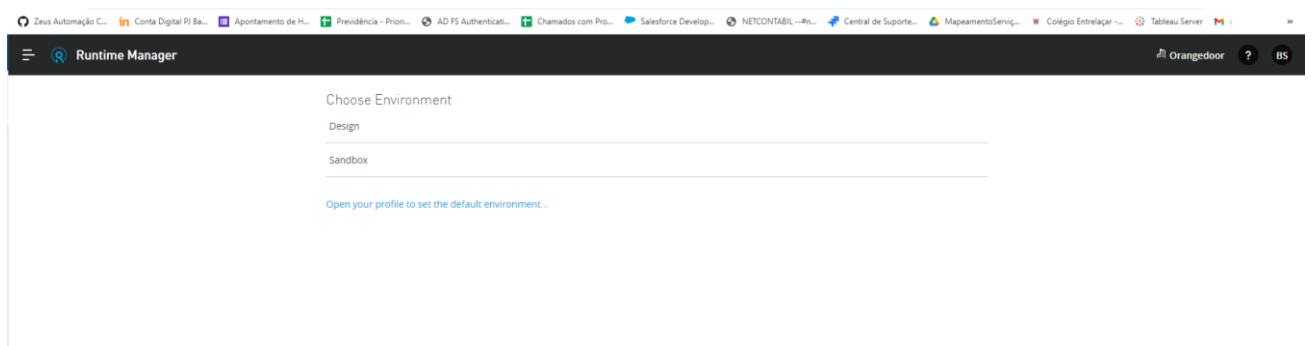
Interface da Anypoint Platform que fornece uma visão unificada de seus ambientes e APIs. Responsável pela implantação, gerenciamento e monitoramento das APIs Mule em ambientes de Design, sandbox ou production, tudo a partir de um local central.

Neste momento iremos realizar o deploy (implantação) de nossa API na Plataforma Anypoint através da ferramenta Runtime Manager

Acesse a Plataforma Anypoint e vá até o Runtime Manager no menu principal

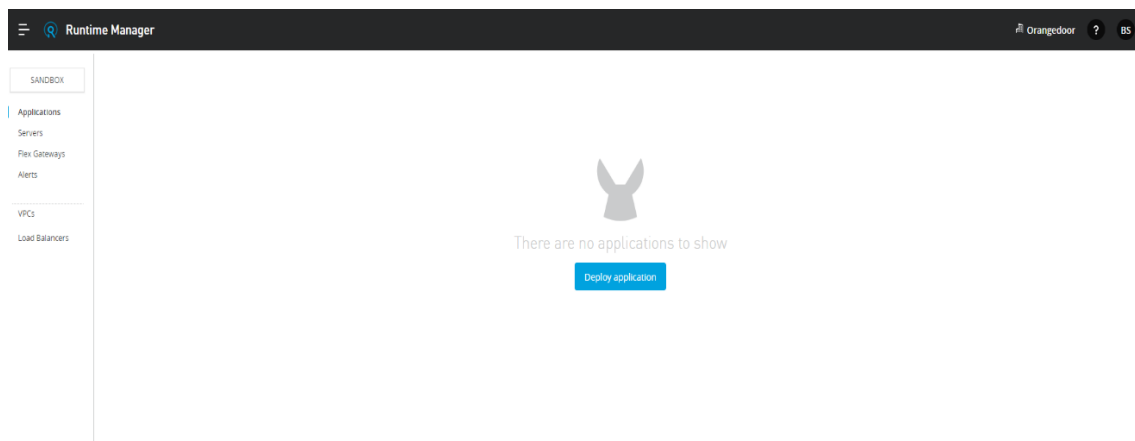


Notem que nesta versão trial disponibilizada pela Mulesoft, temos dois ambientes que podemos utilizar: Design e Sandbox

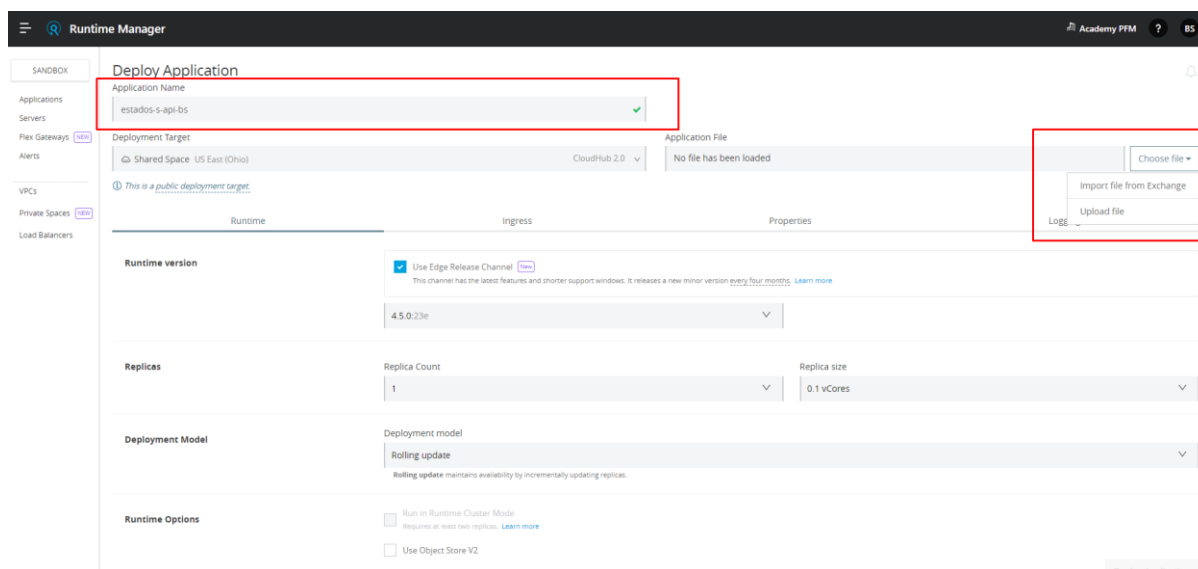


Vamos utilizar o ambiente Sandbox

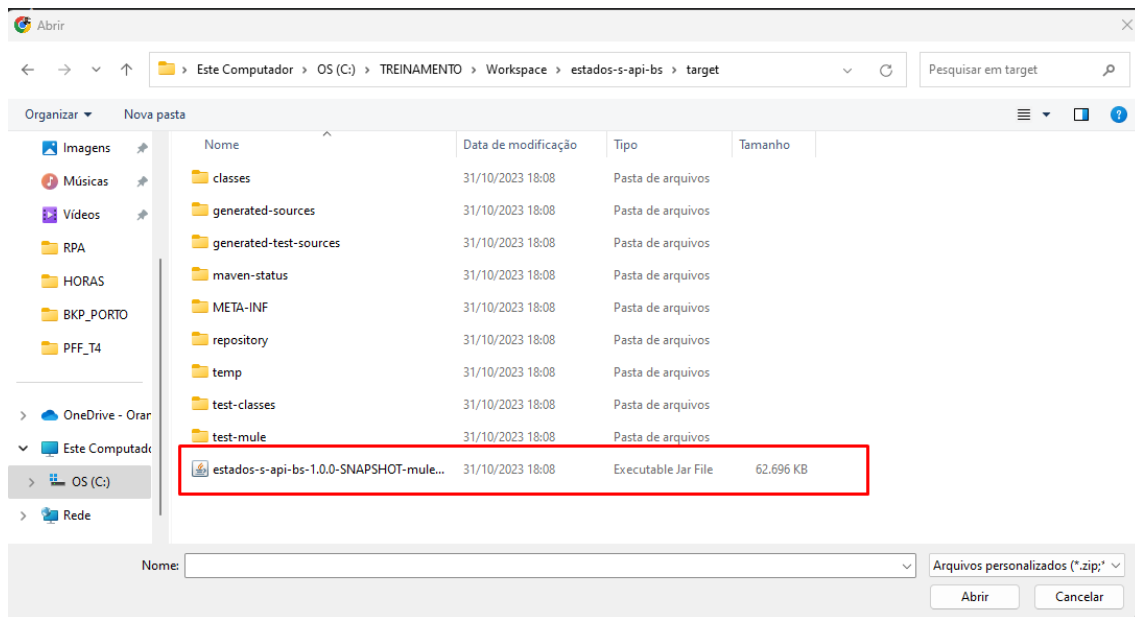
Nesta tela clique em “Deploy application”



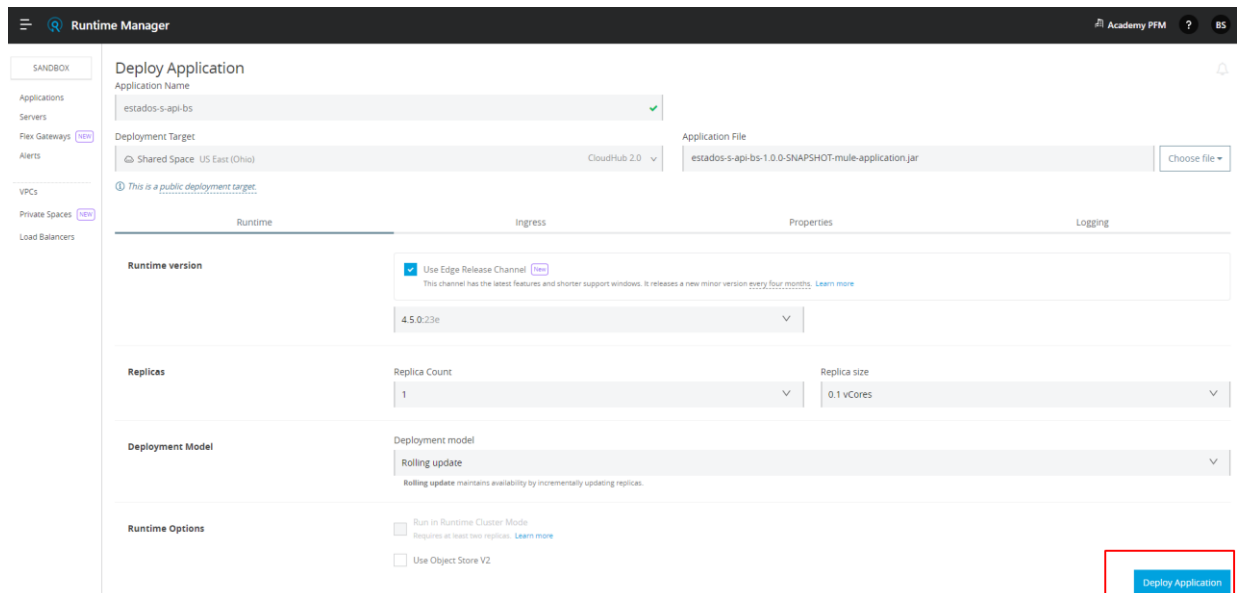
Digite o nome de sua API “estados-s-api-bs” e clique no botão “Choose file” / “Upload file”



Selecione o arquivo “.jar” na pasta target dentro do seu projeto



Clique no botão “Deploy Application”



A API está sendo implantada no ambiente Cloudhub da Mulesoft

Runtime Manager

Academy PFM ? BS

SANDBOX

estados-s-api-bs

Config 35b1c5 (Desired)

Search

Time range

Log Levels (5/5)

Last deployed a few seconds ago - 2023-10-31 18:15 GMT-3

Warn

constraint configuration problems the provider com.mulesoft.agent.external.handlers.metadata.MetadataRequestHandler will be ignored.

a few seconds ago - 2023-10-31 18:16:19.352 GMT-3 - Providers

Listener_start_runner - A provider com.mulesoft.agent.external.handlers.properties.PropertiesRequestHandler registered in SERVER runtime does not implement any provider interfaces applicable in the SERVER runtime. Due to

constraint configuration problems the provider com.mulesoft.agent.external.handlers.properties.PropertiesRequestHandler will be ignored.

Info

a few seconds ago - 2023-10-31 18:16:19.409 GMT-3 - ContextHandler

Listener_start_runner - Started o.e.j.s.ServletContextHandler@8b6751d6(/,/null,AVAILABLE)

Info

a few seconds ago - 2023-10-31 18:16:19.409 GMT-3 - Server

Listener_start_runner - Started @15770ms

Info

a few seconds ago - 2023-10-31 18:16:19.506 GMT-3 - CollectorAgentBase

Listener_start_runner - Starting

Info

a few seconds ago - 2023-10-31 18:16:19.506 GMT-3 - Configuration

Listener_start_runner - Mode: RTF

Info

a few seconds ago - 2023-10-31 18:16:19.507 GMT-3 - Configuration

Listener_start_runner - Premium tier: true

Info

a few seconds ago - 2023-10-31 18:16:19.507 GMT-3 - Configuration

Listener_start_runner - Producer Name: analytics-metrics-collector-mule4

Info

a few seconds ago - 2023-10-31 18:16:19.507 GMT-3 - Configuration

Listener_start_runner - Producer Version: 2.7.0

Info

a few seconds ago - 2023-10-31 18:16:19.507 GMT-3 - Configuration

Listener_start_runner - Prebuffer max size: 10000

Info

a few seconds ago - 2023-10-31 18:16:19.507 GMT-3 - Configuration

Listener_start_runner - Combiner max size: 1000

Info

a few seconds ago - 2023-10-31 18:16:19.507 GMT-3 - Configuration

Listener_start_runner - Aggregation time interval: 60000 ms

Info

a few seconds ago - 2023-10-31 18:16:19.508 GMT-3 - Configuration

Listener_start_runner - Aggregation Top N: 100

Info

a few seconds ago - 2023-10-31 18:16:19.508 GMT-3 - Configuration

Listener_start_runner - Metric log file name: /opt/mule/logs/_metric.log pattern: /opt/mule/logs/_metric-%i.log max size: 10 MB max files: 10

Tudo ocorrendo certo, sua API estará em execução e disponível para ser consumida

Runtime Manager

Academy PFM ? BS

SANDBOX

Deploy application

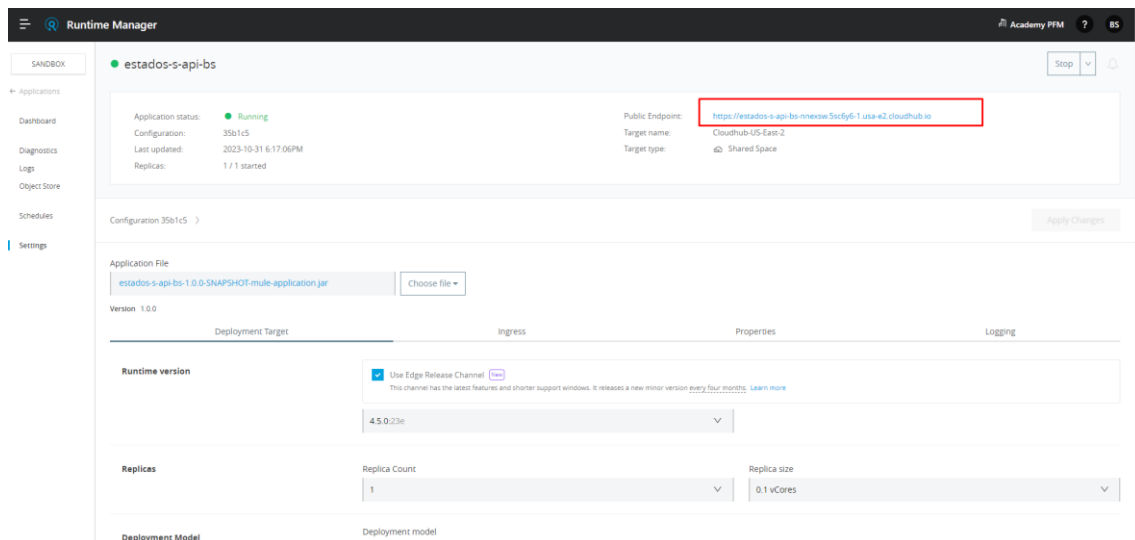
Search Applications

All Applications (1)

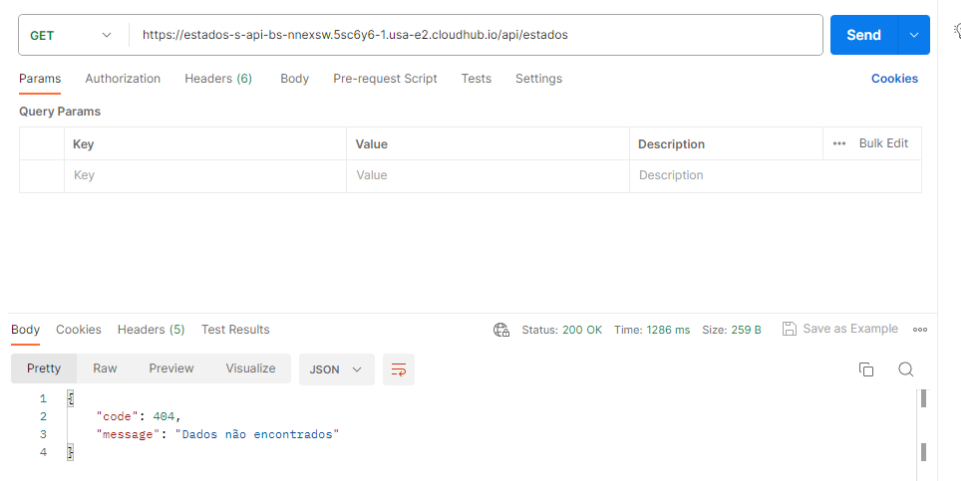
Update Available (0)

Name	Target Name	Target Type	Status	Runtime Version	Date Modified
estados-s-api-bs	Shared Space	CloudHub 2.0	Running	4.5.0-23h	2023-10-31 18:17:06

Este será o domínio para realizar as chamadas de sua API



Agora teste no Postman!



Desafio:

Com base na API Pública abaixo de consulta de CNPJ, desenvolva o contrato e uma API para que seja consumido o serviço de consulta por CNPJ:

Documentação: <https://www.cnpj.ws/docs/intro>

Endpoint a ser consumido: <https://publica.cnpj.ws/cnpj/{cnpj}>

Para ajuda no desenvolvimento do RAML:

<https://danielpoe.github.io/json-to-raml-type/json-to-raml.html>

