

PROGRAMA DE FORMAÇÃO MULESOFT®

Material de Treinamento

SUA TRILHA PARA SE TORNAR UM JEDI COMEÇA AQUI

ORANGEDOOR ACADEMY_

Caro aluno,

Seja bem-vindo! É com muita satisfação que o recebemos na OrangeDoor Academy para execução do Programa de Formação MuleSoft. Estamos juntos nessa jornada de conhecimento e desenvolvimento pessoal e profissional. A seguir você encontrará conceitos importantes para entendimento das aulas, bem como links para download das ferramentas que utilizaremos no decorrer de sua formação.

Caso surjam dúvidas, estamos à sua disposição no grupo exclusivo de alunos no WhatsApp.

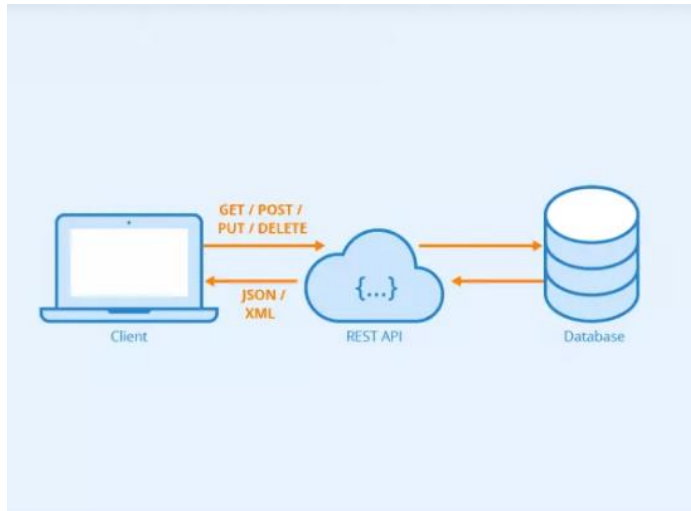
Desejamos a todos excelentes estudos.

Um forte abraço,

Equipe OrangeDoor Academy_

Aula 01

Definição de API



API é uma abreviação para Application Programming Interface. Em português, a sigla significa interface de programação de aplicação.

O conceito de API nada mais é do que uma forma de comunicação entre sistemas. Elas permitem a integração entre dois sistemas, em que um deles fornece informações e serviços que podem ser utilizados pelo outro, sem a necessidade de o sistema que consome a API conhecer detalhes de implementação do software.

Metaforicamente, podemos pensar no que é API como um garçom. Quando estamos em um restaurante, buscamos o que desejamos no menu e solicitamos ao garçom. O garçom encaminha esse pedido à cozinha, que prepara o pedido. No fim, o garçom traz o prato pronto até a gente. Não temos detalhes de como esse prato foi preparado, apenas recebemos o que solicitamos.

Entre os principais benefícios das APIs estão a agilidade e eficácia nas integrações, permitindo que a troca de informações aconteça de forma mais segura e facilitada – tanto para quem disponibiliza os dados quanto para quem utiliza.

Isso acontece porque as APIs otimizam a gestão interna, oferecem ferramentas práticas e base de dados sólidas, facilitando e melhorando a forma de controle dos processos rotineiros e as estratégias da empresa.

Outra grande vantagem é a possibilidade de firmar parcerias e ampliar o negócio. Com o uso de APIs é possível fazer acordos comerciais com intermediadores de pagamentos, redes sociais, blogs, ferramentas de busca, entre outros serviços.

Quando um cliente compra um produto ou serviço em um site, por exemplo, antes que a transação seja confirmada, a API envia os dados do cartão para um aplicativo remoto, que verifica as informações e envia uma resposta para o vendedor confirmar a compra. Tudo isso em apenas alguns instantes, de forma segura e eficaz.

As APIs web normalmente usam o protocolo HTTP/HTTPS para mensagens de solicitação e fornecem uma definição da estrutura das mensagens de resposta. Essas mensagens de resposta geralmente têm o formato de arquivo XML ou JSON. Tanto XML quanto JSON são formatos de preferência porque apresentam os dados de forma simplificada, o que facilita a manipulação por outras aplicações.

SOAP x REST

SOAP (Simple Object Access Protocol) é um protocolo baseado em XML para troca de informações em um ambiente distribuído. É utilizado para troca de mensagens entre aplicativos distribuídos pela rede. Estes aplicativos, ou “Web services”, possuem uma interface de acesso simples e bem definida. Ele se baseia na Linguagem de Marcação Extensível (XML) para seu formato de mensagem.

REST (Representational State Transfer, que significa Transferência Representacional de Estado) é um modelo de arquitetura e não uma linguagem ou tecnologia de programação, que fornece diretrizes para que os sistemas distribuídos se comuniquem diretamente usando os princípios e protocolos existentes da Web. O padrão utilizado para a troca de mensagens neste caso é o JSON (Javascript Object Notation), é um formato compacto, de

padrão aberto independente, de troca de dados simples e rápida (parsing) entre sistemas.

Os recursos dentro do modelo REST permitem mapear qualquer coisa da internet como um elemento de acesso via WEB, assim podemos consumir informações através de seus métodos na internet como se ela fosse nosso próprio banco de dados.

Métodos HTTP comuns

O protocolo HTTP define um conjunto de métodos de requisição responsáveis por indicar a ação a ser executada para um dado recurso. Embora esses métodos possam ser descritos como substantivos, eles também são comumente referenciados como HTTP Verbs (Verbos HTTP).

GET:

O método GET solicita a representação de um recurso específico. Requisições utilizando o método GET devem retornar apenas dados. Como por exemplo, consultar uma base de clientes passando o CPF como critério de busca.

POST:

O método POST é utilizado para submeter uma entidade a um recurso específico, frequentemente causando uma mudança no estado do recurso ou efeitos colaterais no servidor. Como por exemplo, incluir um novo cliente em uma base de clientes.

PUT:

O método PUT substitui todas as atuais representações do recurso de destino pela carga de dados da requisição. Como por exemplo, alterar o endereço de um determinado cliente, passando o CPF como o critério de seleção.

DELETE:

O método DELETE remove um recurso específico. Como por exemplo, excluir um determinado cliente de uma base passando como critério de seleção o CPF.

Glossário de termos utilizados no desenvolvimento de APIs em MuleSoft

Middleware:

Este termo nada mais é que a integração em si. É um serviço ou software que será responsável pela conversão, formatação e transmissão dos dados de um sistema a outro.

Endpoint:

Um endpoint é a URL onde sua API pode ser acessada por uma aplicação cliente.

JSON e XML:

Padrões de formatos de dados, estas são siglas que com certeza você deve ter ouvido diversas vezes. Entenda que para um sistema transmitir ou receber dados através de interfaces públicas, é necessário que estes dados sejam enviados sempre no mesmo formato.

Sistema em camadas:

A arquitetura da API REST inclui várias camadas que operam juntas para construir uma hierarquia que ajuda a gerar um aplicativo mais escalável e flexível. Devido ao seu sistema em camadas, um aplicativo tem melhor segurança, pois os componentes em cada camada não podem interagir fora da camada subsequente.

Request:

Um Request ou requisição traduzindo diretamente para português, é o pedido que um cliente realiza a um servidor.

Response:

A Response (resposta) nada mais é do que a resposta que o servidor envia ao cliente.

Payload:

Carga útil (payload) é a parte dos dados transmitidos que é a mensagem real pretendida.

RAML:

A linguagem de modelagem de API RESTful é uma linguagem baseada em YAML para descrever APIs RESTful.

Plataforma Anypoint:

Plataforma web de gerenciamento das APIs.

Anypoint Studio:

Ferramenta local para desenvolvimento de APIs MuleSoft. Parecido com a IDE Eclipse.

Breakpoint:

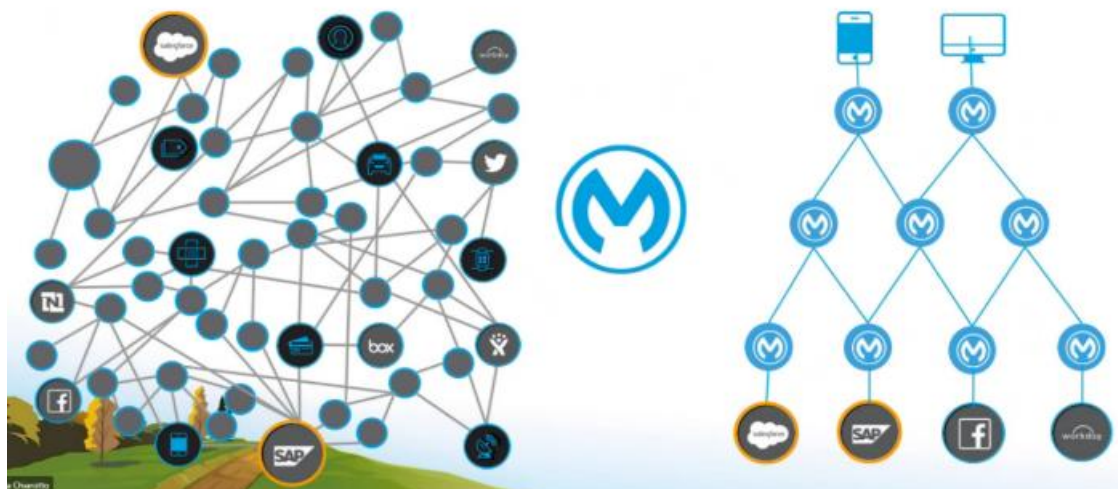
É uma função muito utilizada no Anypoint Studio, onde um ponto de interrupção de dados é colocado em um componente. Assim podemos acompanhar o fluxo de dados em velocidade pausada a partir desse ponto, para obter uma melhor análise dos dados que estão passando.

MuleSoft

Ferramenta de middleware para gerenciamento e desenvolvimento de APIs. A MuleSoft oferece agilidade de negócios excepcional para as empresas, conectando aplicativos, dados e dispositivos on-premises e na nuvem com uma abordagem orientada por APIs.

Os desafios dos clientes

Visão do Ambiente / Application



Ferramentas necessárias para o treinamento

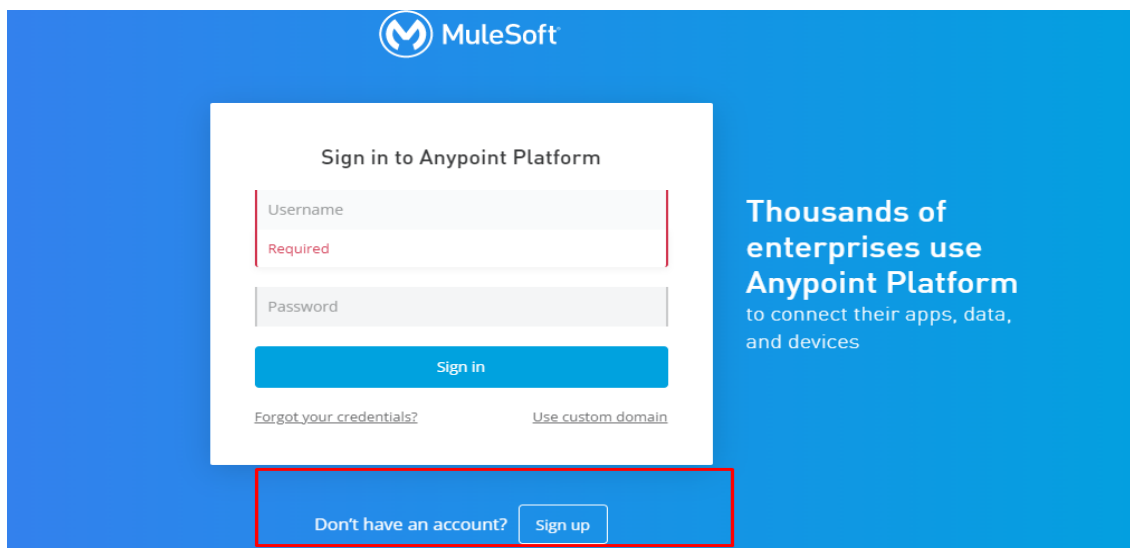
Setup requirements

- A computer with at least 8-16 GB available RAM (16 highly recommended), 2GHz CPU, and 10GB available storage
- Internet access to port 80 (with > 5Mbps download and > 2Mbps upload)
- **Anypoint Studio 7.12.0 or later** with embedded Mule runtime

DOWNLOADS:

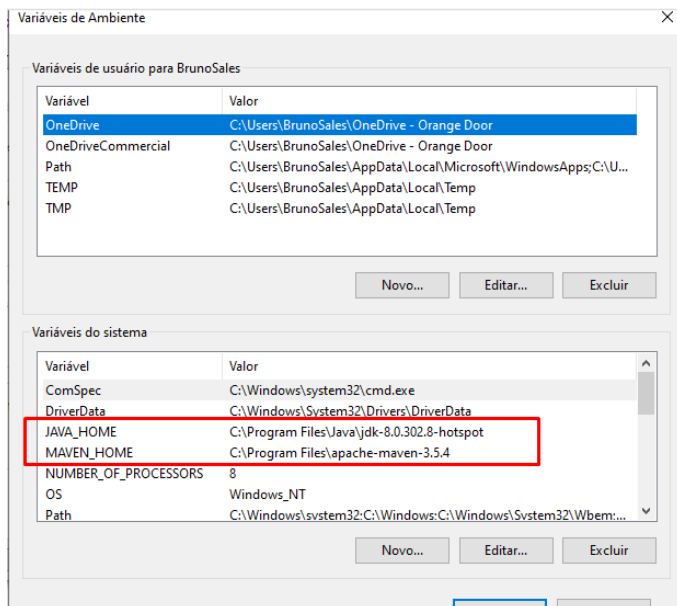
- JDK8 - [jdk-8u202-windows-x64.exe](#)
- Maven 3.8.6 - [apache-maven-3.8.6-bin.zip](#)
- Anypoint Studio 7.12.1 - [AnypointStudio-7.12.1-win64.zip](#)
- Postman - [Postman-win64-8.10.0-Setup.zip](#)
- Cadastro na Plataforma Anypoint - <https://anypoint.mulesoft.com/>

Obs.: Ao se cadastrar na Plataforma Anypoint é disponibilizada uma versão trial de 30 dias de uso. Sendo assim, realizem o cadastro somente no dia de início do treinamento para obter um tempo maior para uso.

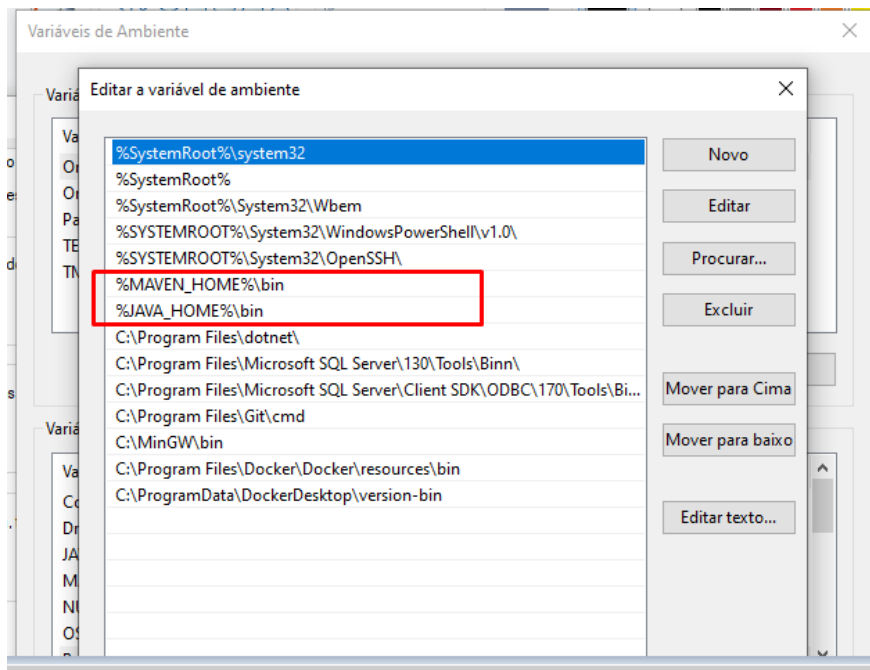


Setup para desenvolvimento MuleSoft:

Criar as variáveis de ambiente JAVA_HOME e MAVEN_HOME e apontar para os seus respectivos diretórios:



Configurar o "Path" destas variáveis apontando para o diretório "bin"



Executar os comandos:

- `java -version`
- `mvn -version`
- `mvn clean`

```
Prompt de Comando
Microsoft Windows [versão 10.0.19043.1826]
(c) Microsoft Corporation. Todos os direitos reservados.

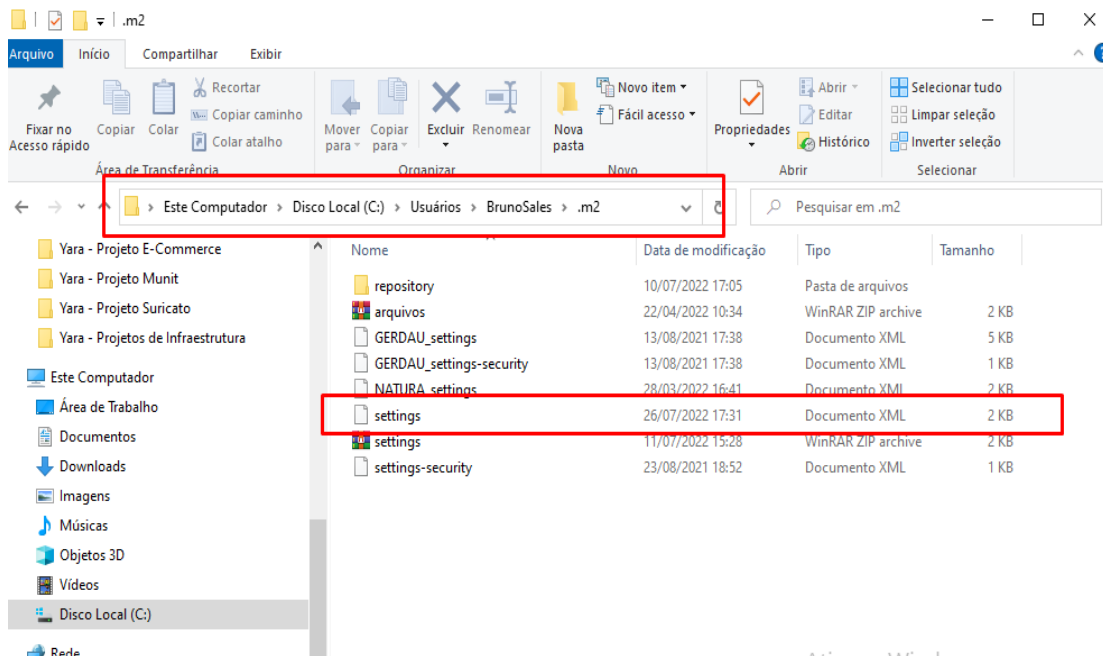
C:\Users\BrunoSales>java -version
openjdk version "1.8.0_302"
OpenJDK Runtime Environment (Temurin)(build 1.8.0_302-b08)
OpenJDK 64-Bit Server VM (Temurin)(build 25.302-b08, mixed mode)

C:\Users\BrunoSales>mvn -version
Apache Maven 3.5.4 (1edda0938998edf8bf061f1ceb3cfdeccf443fe; 2018-06-17T15:33:14-03:00)
Maven home: C:\Program Files\apache-maven-3.5.4\bin\..
Java version: 1.8.0_302, vendor: Temurin, runtime: C:\Program Files\Java\jdk-8.0.302.8-hotspot\jre
Default locale: pt_BR, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"

C:\Users\BrunoSales>
```

No diretório “.m2” precisamos incluir o arquivo settings.xml

- download settings.xml



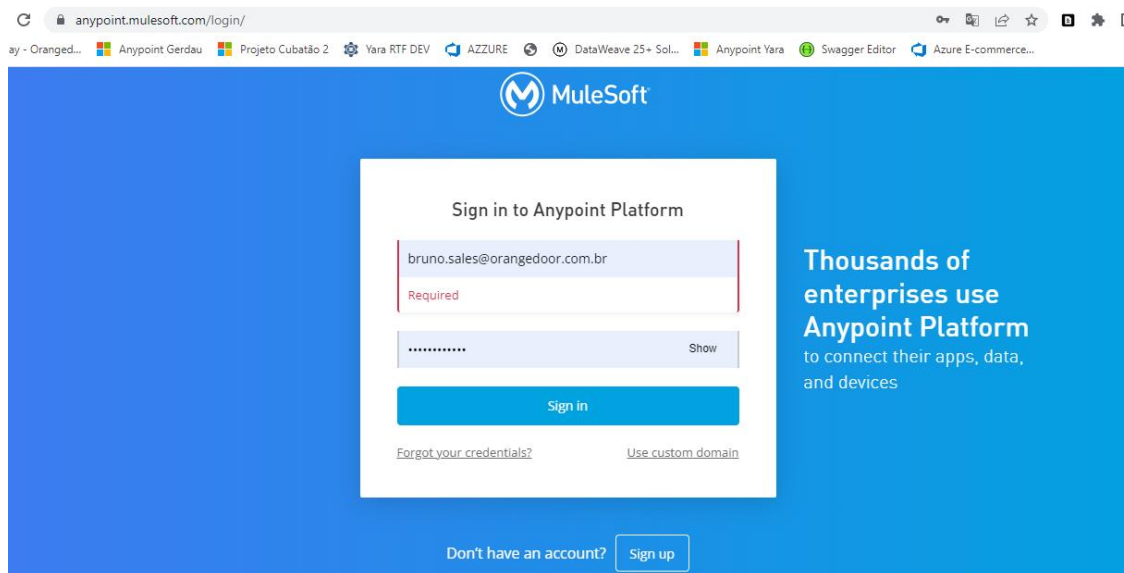
Alterar usuário e senha do arquivo settings.xml conforme suas credenciais criadas na Plataforma Anypoint



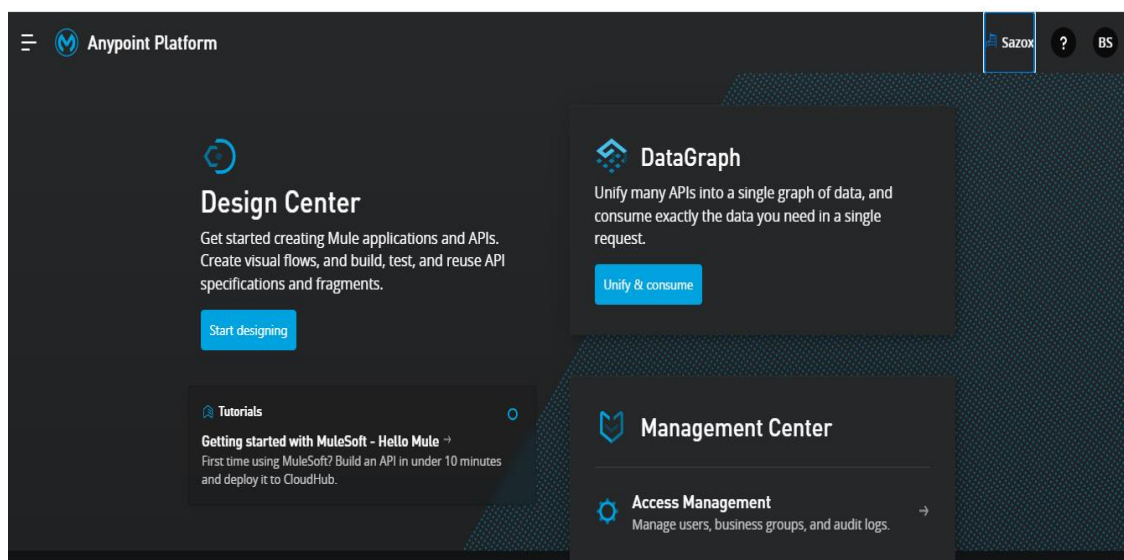
Acessando a Plataforma Anypoint

<https://anypoint.mulesoft.com/login/>

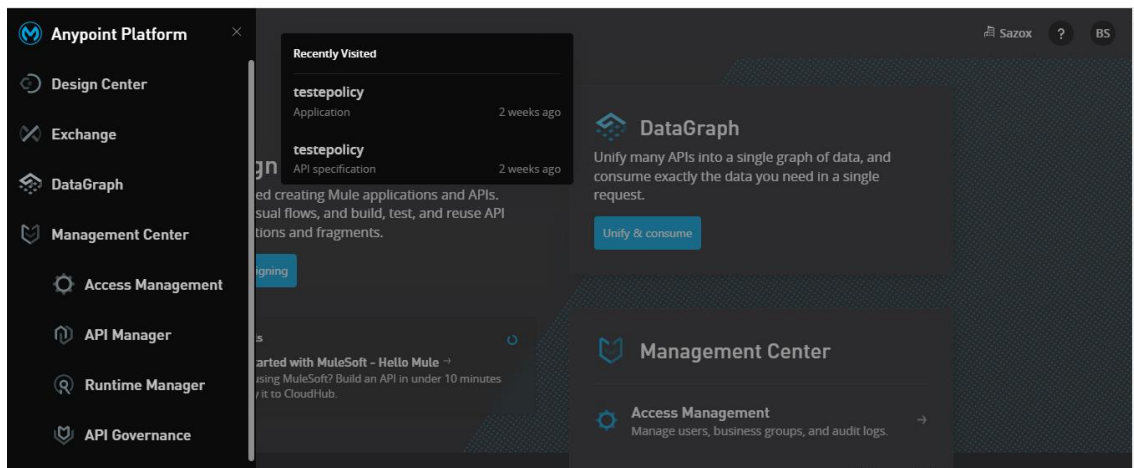
Entrar com o login e senha cadastrados anteriormente



Tela inicial



No menu principal podemos observar as seguintes ferramentas:



Code Builder:

IDE online para desenvolvedores projetarem, desenvolverem e implantarem APIs, integrações e automações em um único ambiente. Ainda está na versão Beta (em testes).

Design Center:

Responsável pela criação de contratos (RAML) das API. Onde definimos as entradas e respostas utilizando um editor web. Possibilitando a reutilização de modelo de dados e esquemas de segurança na forma de fragmentos de API, além da realizar a publicação do contrato e documentação no Anypoint Exchange.

Exchange:

Responsável por armazenar todas as documentações de APIs, conectores, políticas de segurança personalizadas, para que sejam consultados/reutilizados por outros usuários. Podemos também realizar testes de consumo das APIs.

DataGraph:

Permite criar um esquema unificando APIs existentes no Exchange para que seja possível a utilização deste

esquema para outros requisitos de negócio da empresa, não havendo a necessidade de criação de novas APIs.

Access Management:

Permite a criação, manutenção e gerenciamento de usuários, regras, ambientes, auditoria de logs e políticas de segurança na Plataforma Anypoint.

API Manager:

Responsável pelo gerenciamento das APIs, onde é possível bloquear ou gerenciar qualquer serviço com a segurança de um gateway de API, aplicar políticas de segurança pré-definidas ou personalizadas.

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 staging, sandbox ou production, tudo a partir de um local central.

API Governance:

Permite garantir o padrão de conformidades das APIs de sua organização, identificando os problemas de não conformidades e tomando medidas para resolvê-los.

Visualizer:

Ferramenta que fornece uma representação gráfica em tempo real das APIs Mule que estão em execução, onde os dados exibidos no gráfico são atualizados dinamicamente.

Monitoring:

Ferramenta responsável pelo monitoramento das APIs em execução permitido criar painéis gráficos customizados por recurso, através destes painéis é possível detectar anomalias e problemas e corrigi-los de forma rápida. Além da geração de métricas de consumo das APIs

Secrets Manager:

Gerenciador de segredos da Plataforma Anypoint, usa a tecnologia de cofre de senhas para armazenar e controlar acesso a chaves privadas e senhas.

Introdução ao RAML

Quando desenvolvemos uma API temos que levar em consideração que ela irá fornecer uma interface para que outros sistemas consigam se comunicar com nosso sistema através dela, quanto mais claro for o contrato dela, maior será a chance de a API ser bem-sucedida.

Através da abordagem Contract-First é possível lançar uma prévia da API antes mesmo de implementá-la, inclusive fornecer um endpoint que permite a simulação das operações mesmo fase do design, antes mesmo de pensar em questões técnicas como linguagem ou servidor. Com isso é possível antecipar alguma mudança futura, pois é possível obter o feedback de quem for consumi-la.

Com o RAML é possível criar o contrato de uma API utilizando uma sintaxe bem enxuta e objetiva.

Utilizando o RAML com o Anypoint Platform é possível:

- Fazer a especificação do contrato
- Realizar testes através de mock
- Gerar um portal de documentação da API
- Compartilhar essa documentação
- Realizar a geração do código de forma automática a partir do RAML, existem plug-ins para diversas linguagens.

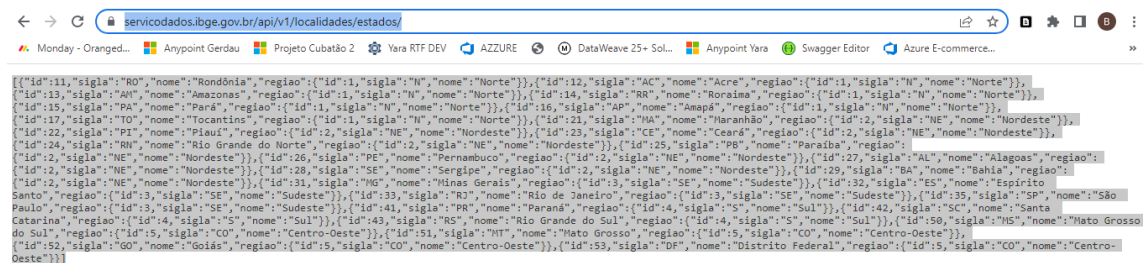
Construindo nosso primeiro Contrato RAML usando a Plataforma Anypoint

Iremos utilizar como base para os nossos exercícios, algumas APIs disponibilizadas pelo IBGE, conforme documentação:

<https://servicodados.ibge.gov.br/api/docs/>

Considere a seguinte API do IBGE, que ao ser consumida retorna os dados dos estados brasileiros:

<https://servicodados.ibge.gov.br/api/v1/localidades/estados/>



Agora com base nesta API e os dados retornados, iremos construir nosso contrato (especificação) utilizando a Plataforma Anypoint.

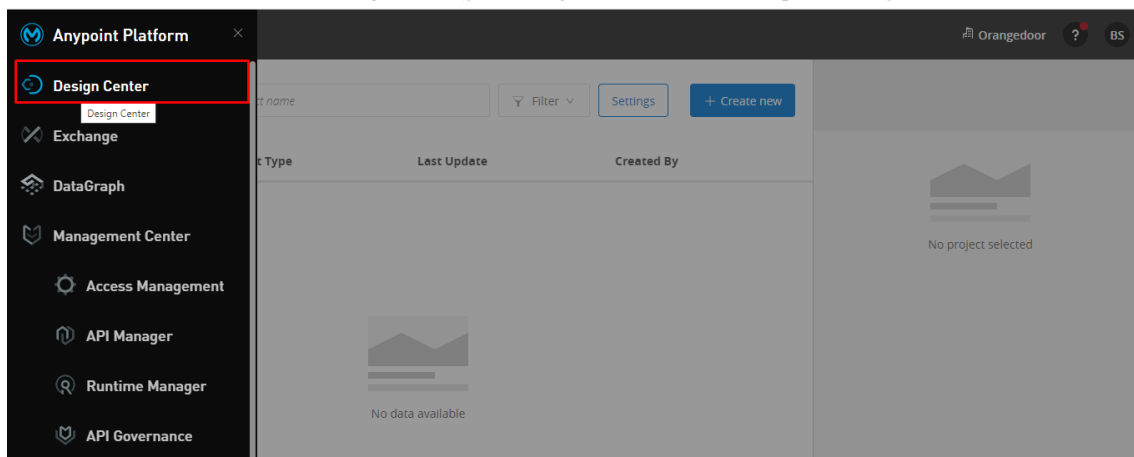
Exercício: 1

Download: [Exercicio1.txt](#)

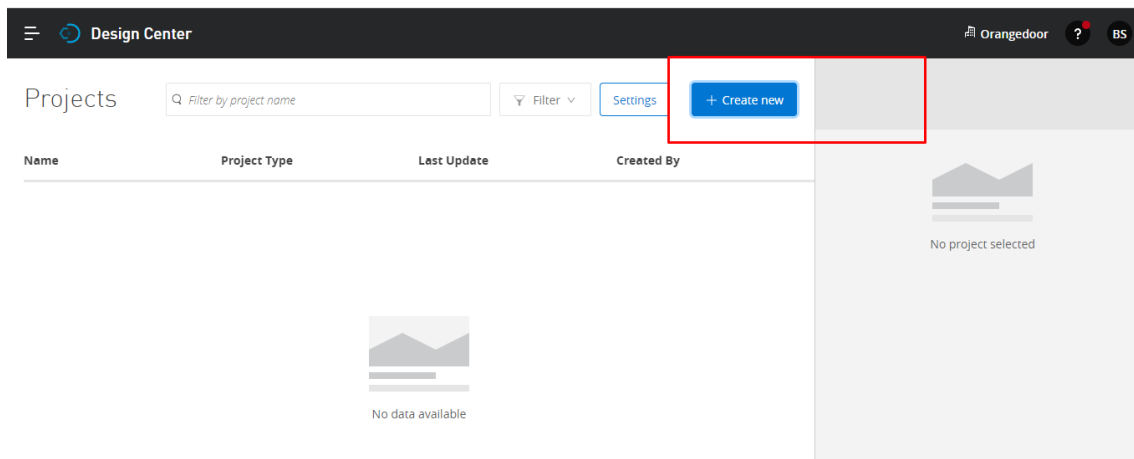
Iremos desenvolver a especificação do contrato de nossa API baseada no RAML. Podemos observar que os dados retornados do IBGE são: id, sigla, nome e região.

Para iniciar o desenvolvimento da especificação, iremos utilizar a ferramenta “Design Center”

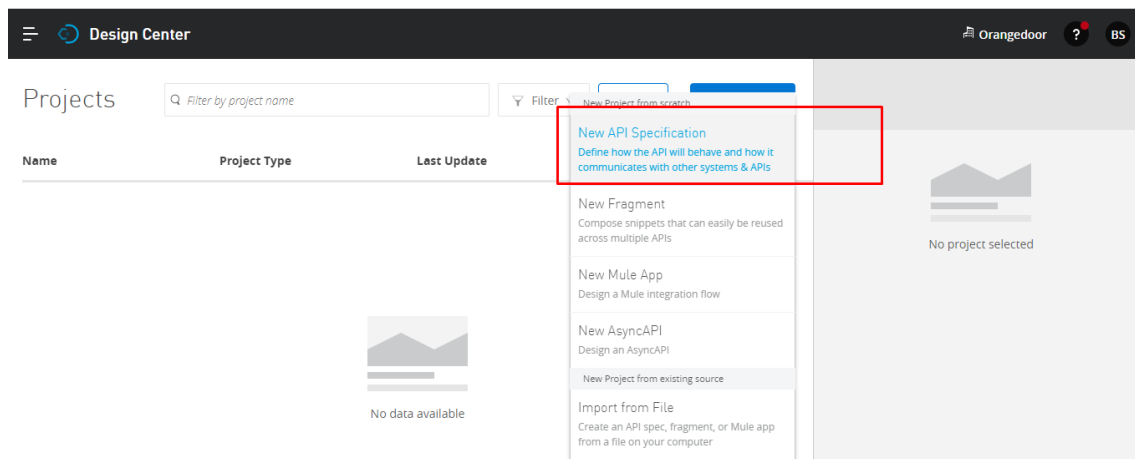
Clique no Menu Principal da Plataforma Anypoint e escolha a opção Design Center.



Clique no botão “Create new”



Selecione “New API Specification”



Digite o nome da sua especificação do contrato “estados-ibge-bs” (BS= Bruno Sales.... utilize as iniciais de seu nome) e clique em “Create API”

New API Specification

Project Name

estados-ibge

How do you want to draft the API Spec?

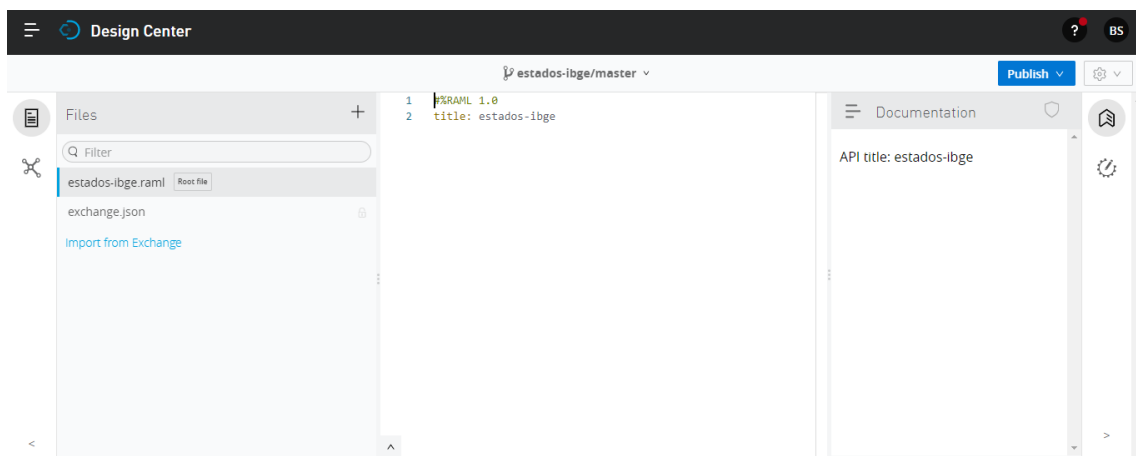
☒ **I'm comfortable designing it on my own**
A complete code editing experience with interactive documentation

Specification Language

RAML 1.0

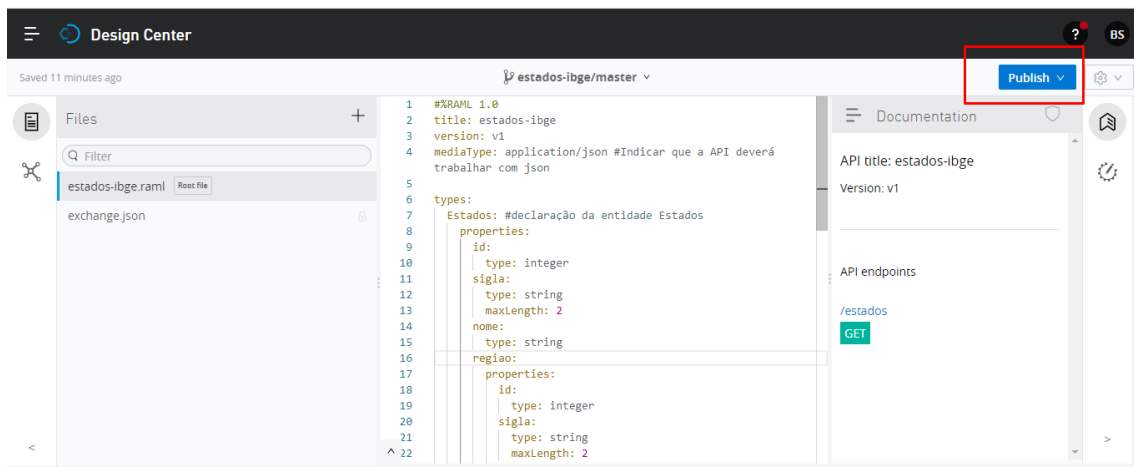
☐ **Guide me through it**
Use a visual interface scaffolding the API Specification (can generate both RAML & OAS)

Neste momento você será direcionado para a tela do edit do RAML:



Agora iremos escrever o seguinte trecho de código para a criação do contrato RAML.

Após terminar o código, caso não tenha erros iremos publicar nosso RAML no “Exchange” clicando no botão “Publish” no canto superior direito.



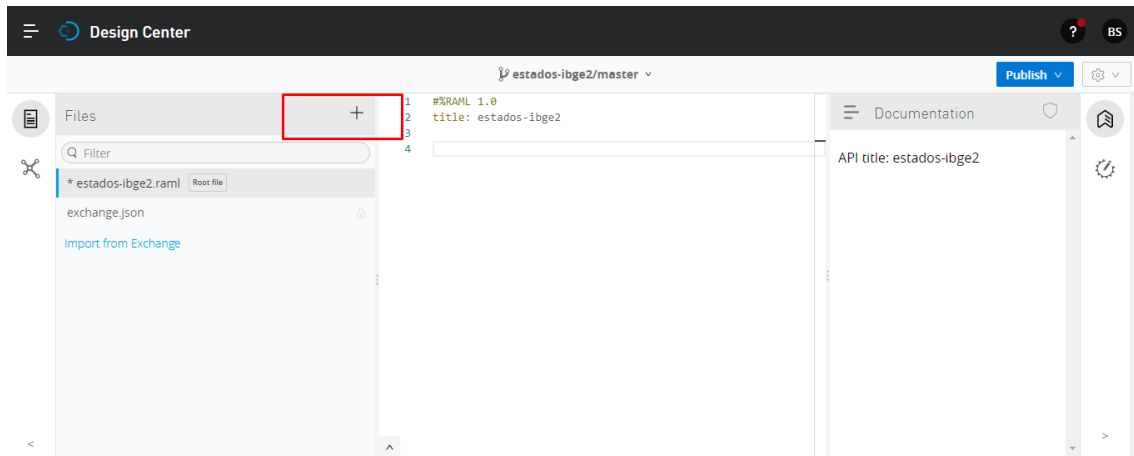
Inclua no campo Asset Version: 1.0.0

A screenshot of the 'estados-ibge' Asset Versioning dialog box. It contains fields for 'Asset version (required)' with the value '1.0.0' and 'API version (required)' with the value 'v1'. Below these are radio buttons for 'Lifecycle State' with 'Stable' selected. To the right, there is explanatory text about SemVer and links for additional help. At the bottom, there are 'Cancel' and 'Publish to Exchange' buttons.

Após publicado nosso RAML, podemos verificar nosso contrato no Exchange

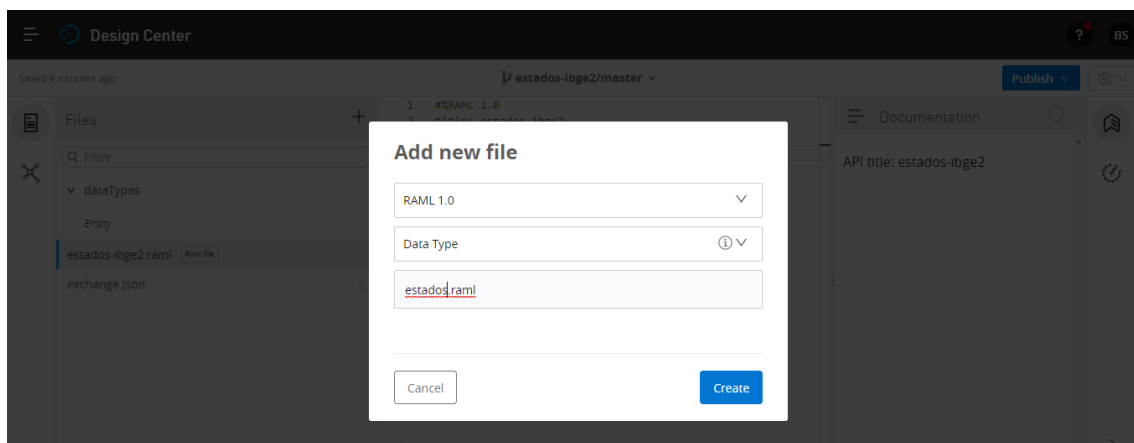


Ao iniciar iremos adicionar uma nova pasta chamada “dataTypes” clicando no sinal “+” / “Folder”

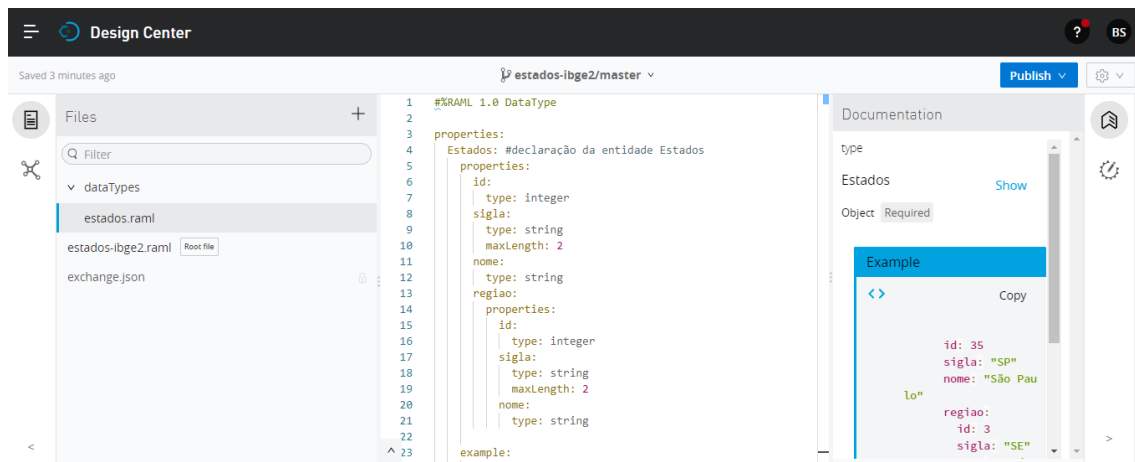


Add new folder

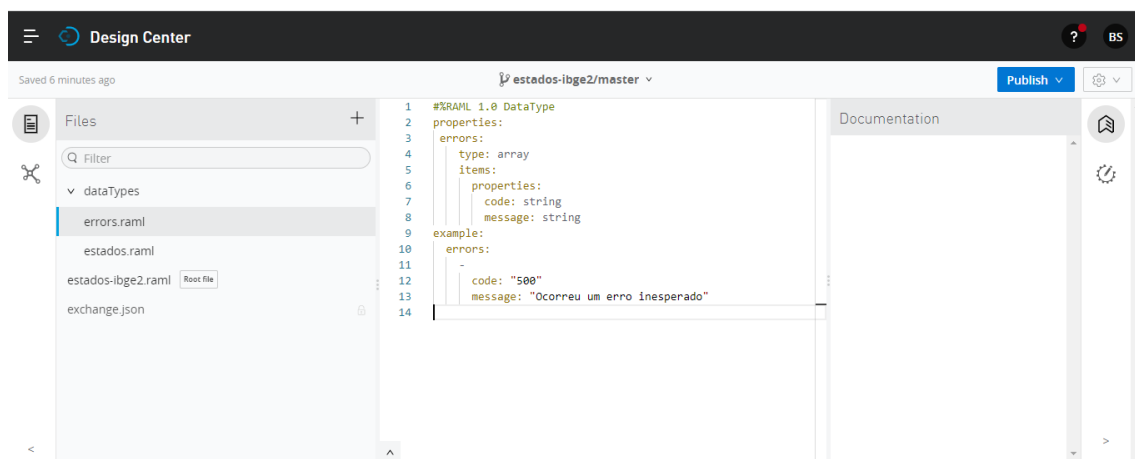
Dentro da pasta “dataTypes” iremos adicionar o arquivo do tipo “Data Type” chamado “estados.raml”



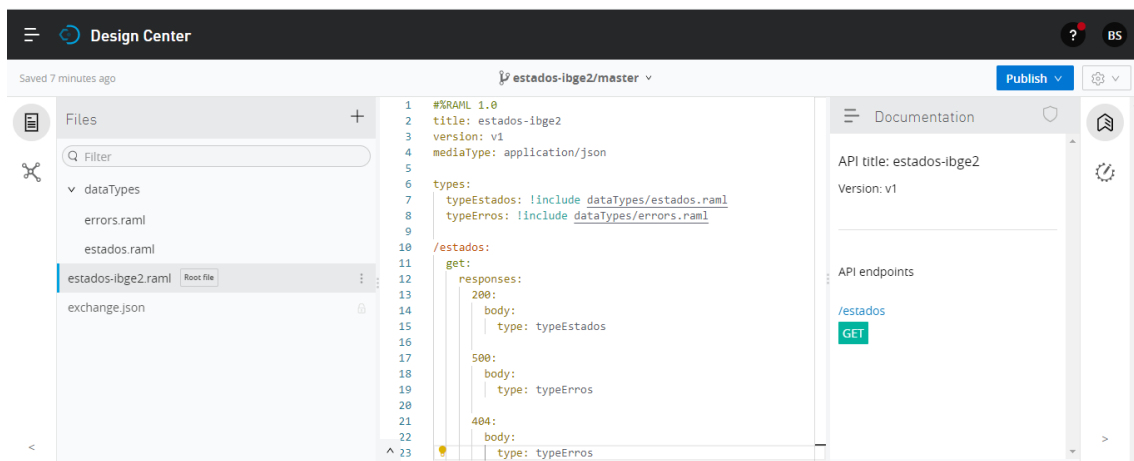
No arquivo *dataType estado.raml*, digite o código abaixo:



Ainda dentro da pasta “dataTypes”, crie um novo arquivo do tipo Data Type com o nome de “errors.raml” e digite o código abaixo:



Por final, no arquivo “estados-ibge2” (root), digite o código abaixo:



Realize a publicação no Exchange e verifique como ficou.

Exercício 3:

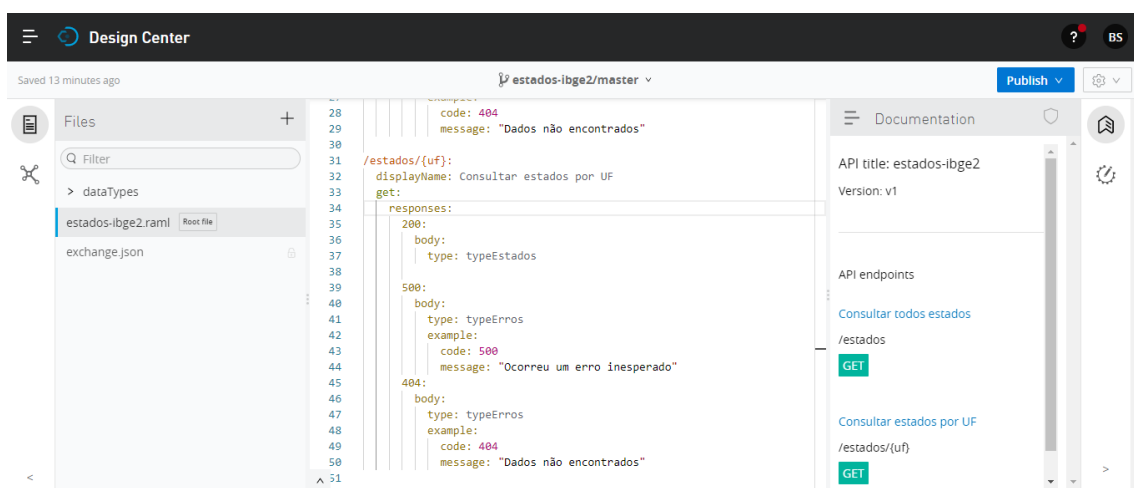
Neste exercício iremos criar um novo endpoint no RAML “estados-ibge2”, onde teremos que passar a sigla da UF como URI para a consulta de um determinado estado:

<https://servicodados.ibge.gov.br/api/docs/localidades#api-UFs-estadosUFGet>

Endpoint:

<https://servicodados.ibge.gov.br/api/v1/localidades/estados/{UF}>

Download [Exercicio3.txt](#)



Realize a publicação no Exchange e verifique como ficou.