

Módulo 1: Design Center

O **Anypoint Design Center** fornece um editor robusto para projetar as especificações da API e criar um serviço simulado (*Mocking Service*) para acelerar os esforços de desenvolvimento.

REST API são muito populares e suportam uma variedade de padrões que ajudam a resolver muitos desafios de confiabilidade, escalabilidade e integração. Usaremos a abordagem de "*API design first*" usando *RESTful API Modeling Language (RAML)*. Seguiremos as práticas recomendadas do REST para promover a adoção de API.

Para mais informações sobre RAML, consulte: www.raml.org.

Etapas do laboratório

Nesse workshop criaremos uma API do zero em questão de minutos.

[Etapa 1: Criar um projeto de API no Design Center](#)

[Etapa 2: Definir os parâmetros iniciais da API](#)

[Etapa 3: Criar a primeira operação da API](#)

[Etapa 4: Criar uma operação de consulta](#)

[Etapa 5: Especificar uma resposta para consulta](#)

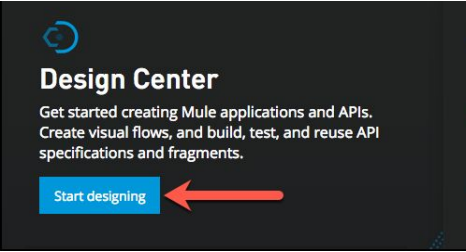
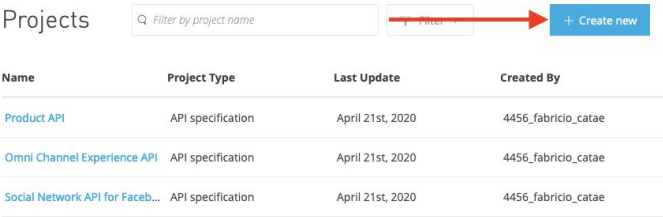
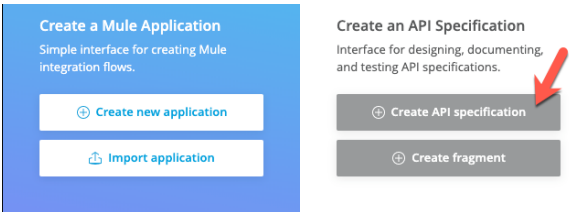
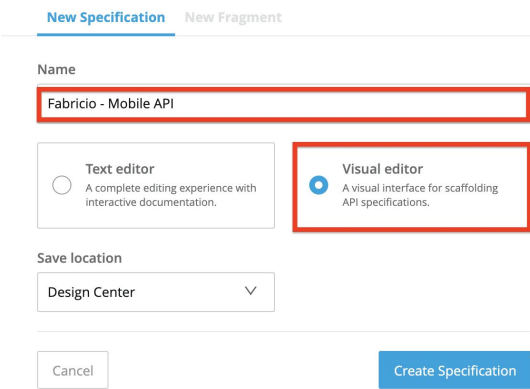
[Etapa 6: Criar uma resposta exemplo](#)

[Etapa 7: Habilitar o Mocking Service](#)

[Etapa 8: Testar o Mocking service](#)

Etapa 1: Criar um projeto de API no Design Center

É na fase de design que definimos a interface da API, ou seja, seus recursos, sub-recursos, parâmetros de entrada e saída, tipo de dados, etc. No **Anypoint Design Center**, você verá uma lista de APIs que já foram definidas pela sua organização.

1. Você pode acessar o Design Center na página inicial da Anypoint Platform .	
2. Para criar uma nova API, clique no botão Criar .	
3. Selecione Create API Specification .	
4. Defina os seguintes valores: <ul style="list-style-type: none">Nome da API:<seu_nome_aqui> - Mobile APISelecione o botão de opção Visual Editor <p><i>Nota: Como estamos em uma ambiente compartilhado, forneça um nome único à sua API. Neste exercício, acrescente o seu usuário ao nome da sua API.</i></p>	
5. Clique no botão Create Specification .	

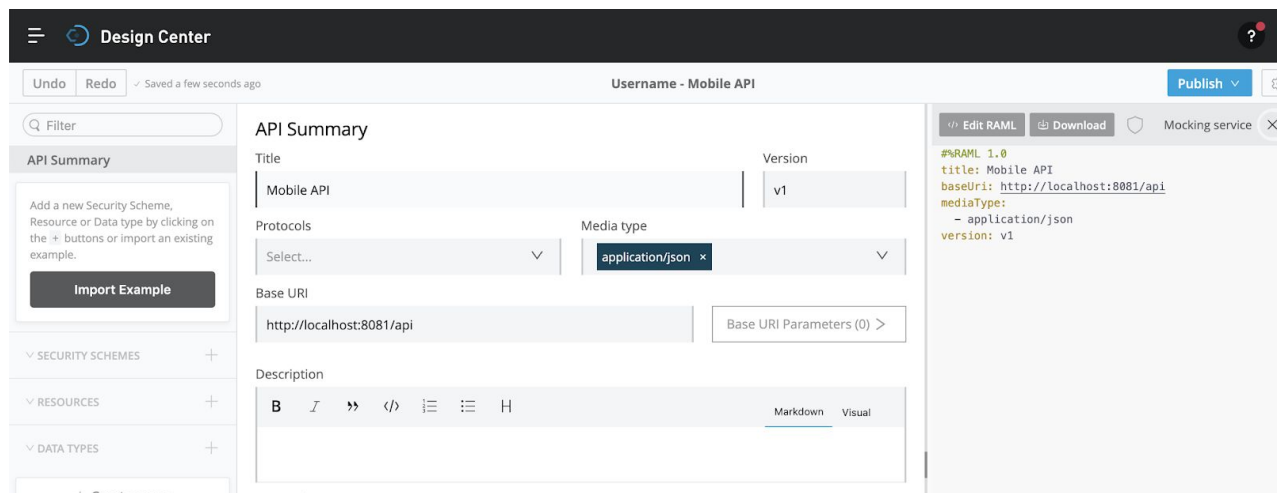
O projeto foi criado no **Design Center** e você está pronto para criar sua primeira API.

Etapa 2: Definir os parâmetros iniciais da API

Começamos configurando os parâmetros iniciais da sua API:

1. Defina um título inicial (pode ser diferente do nome do projeto). Por exemplo, Mobile API	<div>API Summary</div> <div>Title</div> <div>Mobile API</div>
2. Como esse é nosso primeiro design, definiremos essa API como versão v1 .	<div>Version</div> <div>v1</div>
3. Defina o tipo de dado como application/json	
4. A maioria das especificações da API definirá uma URI base. Esse campo informa ao consumidor da API onde está estará acessível. Defina o Base URI como: http://localhost:8081/api	<div>Base URI</div> <div>http://localhost:8081/api</div>

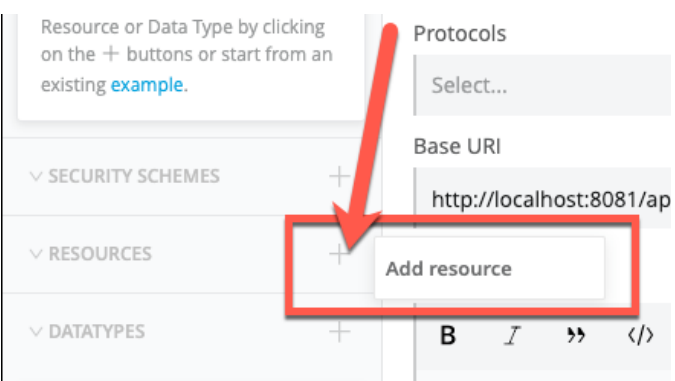
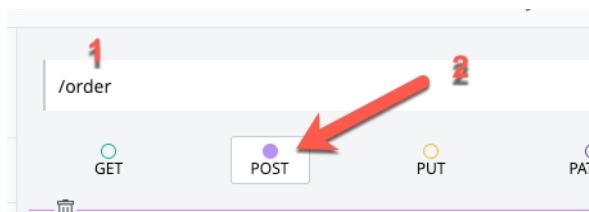
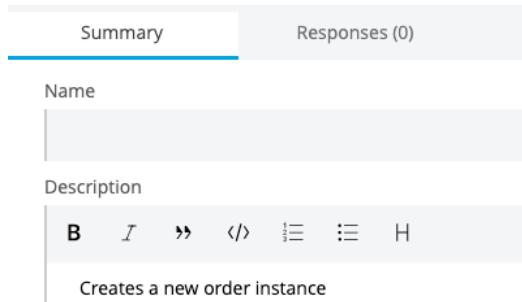
Preenchemos a definição inicial da sua API no **Design Center**:



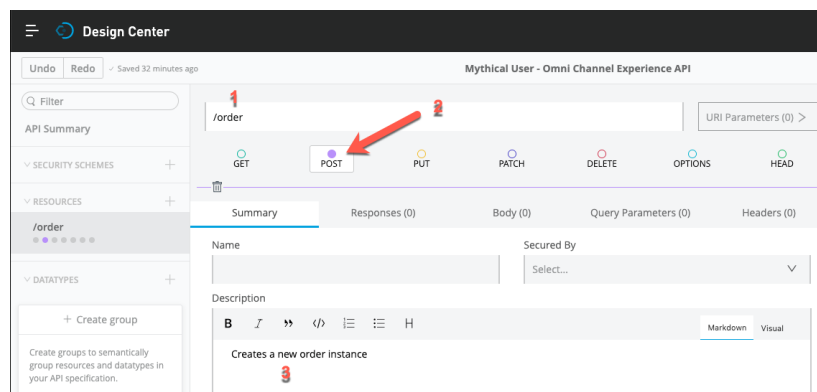
O próximo passo será adicionar os recursos disponíveis na sua API.

Etapa 3: Criar a primeira operação da API

O objetivo da API será o tratamento sobre os pedidos (em inglês, "order"). Nessa etapa vamos criar uma operação de **POST /order**, que corresponde a criação de um pedido.

<p>Utilize o menu lateral à esquerda para adicionar um novo recurso ("resource").</p> <ol style="list-style-type: none">1. Clique no botão + à direita de Resources.2. Após pressionar o botão, escolha Add resource	
<ol style="list-style-type: none">3. Renomeie o recurso de /new-resource para /order4. Selecione a opção POST.	
<ol style="list-style-type: none">5. Preencha os campos de nome ou descrição.6. Por exemplo, na documentação, descreva a operação a ser realizada. Por exemplo, podemos descrever: "Create a new order instance"	

Com estes passos, já temos a primeira operação: **POST /order**, que será usada para criar novos pedidos.



O próximo passo será criar uma operação para consultar os pedidos existentes.

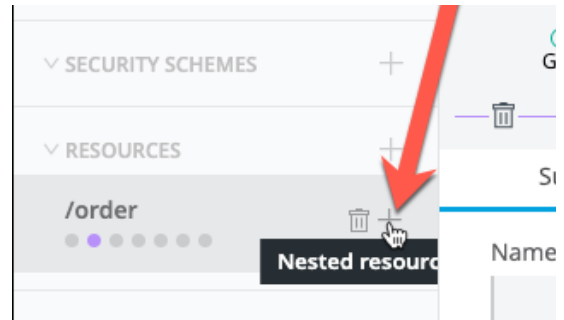
Etapa 4: Criar uma operação de consulta

Na etapa anterior, criamos uma operação para permitir que o aplicativo de comércio eletrônico crie um novo pedido. E se quisermos permitir que o aplicativo de comércio eletrônico obtenha o status de um pedido já existente?

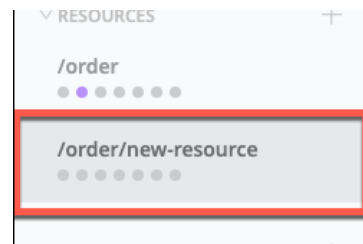
Precisamos criar a operação **GET /order/{order_id}**.

Utilize o menu lateral à esquerda para adicionar um novo recurso ("resource").

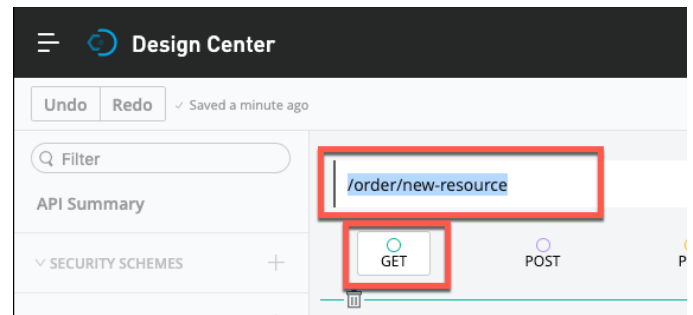
1. Passe o cursor do mouse sobre o recurso **/order**.
Você notará que um botão **+** aparece junto com a mensagem de "nested resource"
2. Pressione o botão **+** para criar um novo sub-recurso



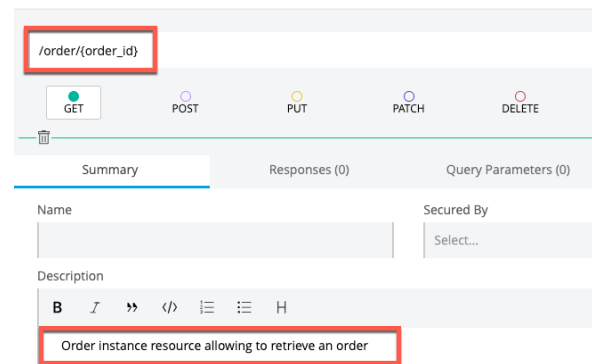
3. Observe que um novo recurso foi criado com o nome **/order/new-resource**.



4. Além disso, o método GET foi selecionado por padrão.

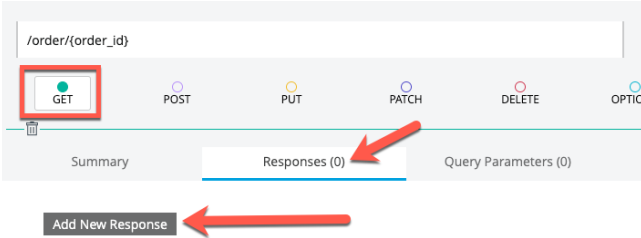
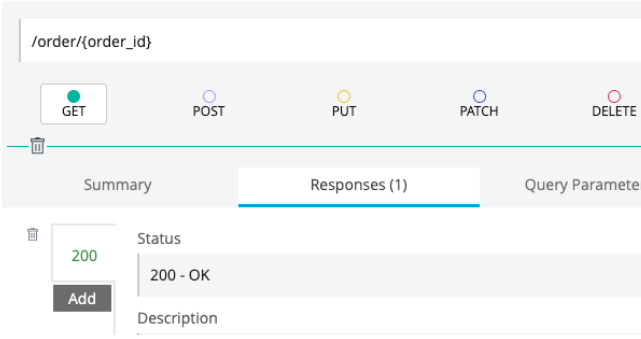
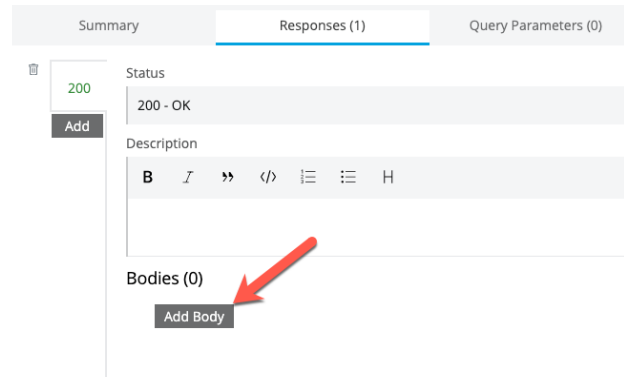
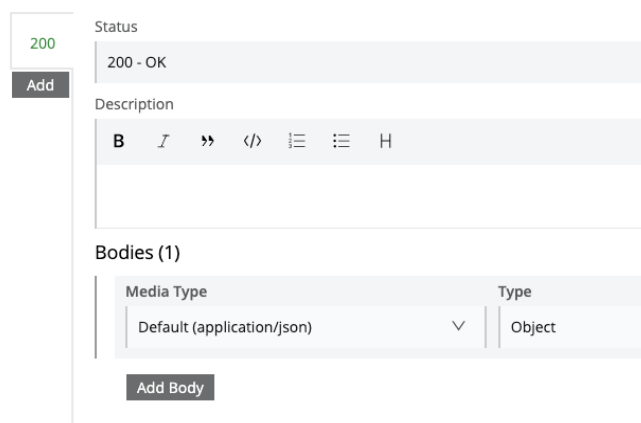


5. Renomeie **/order/new-resource** para o nome **/order/{order_id}**. Essa é a forma de indicar variáveis dinâmicas.
6. Insira uma descrição na documentação. Por exemplo: "Busca um pedido específico através do seu id"



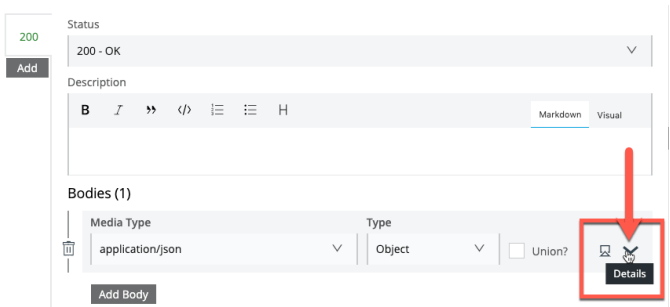
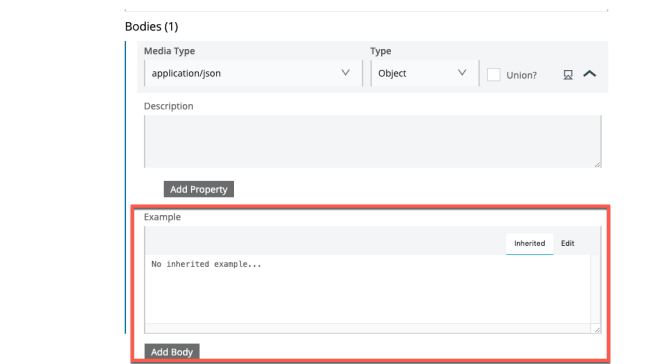

Etapa 5: Especificar uma resposta para consulta

Nessa etapa, vamos definir que o método **GET /order/{order_id}** retorna um objeto **JSON**.

<div>1. No método GET, clique em Responses</div> <div>2. Em seguida, clique em Add New Response.</div>	
<div>3. Agora você deve selecionar uma resposta com status de 200 OK.</div>	
<div>4. Clique em Add Body</div>	
<div>5. Em seguida, especifique o tipo de retorno como JSON (Default application/json) e tipo Object</div>	

Etapa 6: Criar uma resposta exemplo

Adicionalmente, podemos criar um exemplo para nosso método `GET /order/{order_id}`. Essa informação será usada futuramente na documentação e no Mocking Service.

<div>1. Continuando a etapa anterior, precisamos expandir a seção clicando em Details. Assim podemos adicionar mais detalhes em Bodies(1).</div>	
<div>2. Observe que, depois de expandido, aparecerá a seção Example com uma mensagem de "No inherited example..."</div>	
<div>3. Em Example, clique na guia Edit. 4. Em seguida, adicione um código JSON qualquer. Vide os exemplos abaixo (4a e 4b).</div>	
<div>4a. Exemplo 1: JSON simples</div>	<pre>{ "order_id": "pedido-0234" }</pre>
<div>4b. Exemplo 2: JSON com vários elementos. <i>Nota: verifique se a sintaxe está correta (aspas, vírgulas, colchete e chaves).</i></div>	<pre>{ "order_id": "pedido-0234", "items": [{"produto": "121", "nome": "mouse"}, {"produto": "122", "nome": "teclado"}] }</pre>

Etapa 7: Habilitar o Mocking Service

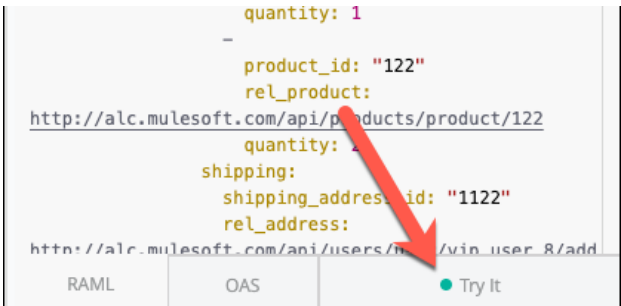
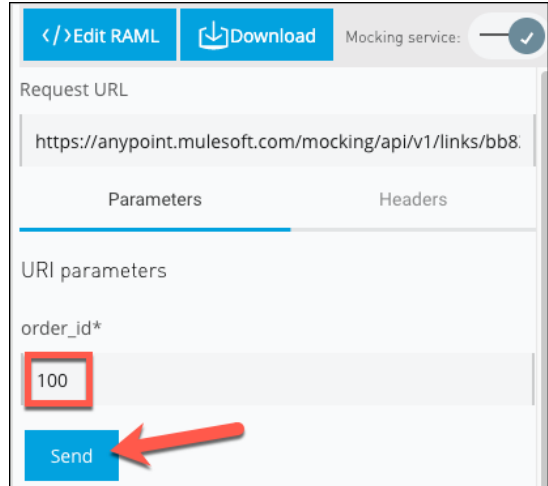
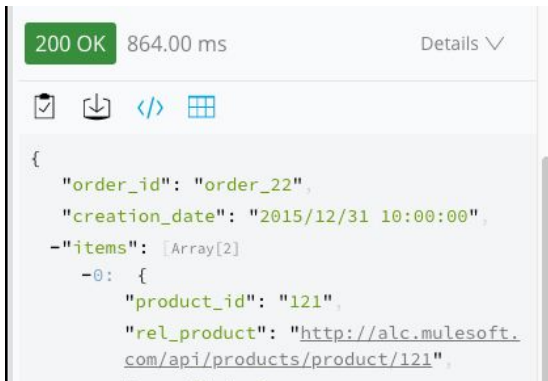
O **Anypoint Design Center** fornece a possibilidade de habilitar **Mocking services**, que permite aos usuários interagir com a API como se ela já estivesse construída e implantada. Esse recurso permite que você disponibilize rapidamente a iteração entre a API e seus consumidores durante a fase inicial de desenvolvimento.

<p>1. No lado direito superior, observe que existe um painel com o código RAML gerado automaticamente.</p>	
<p>2. Observe que o campo baseUri no RAML Viewer corresponde ao valor definido na etapa inicial</p> <p>3. Habilite o botão do Mocking service</p>	
<p>4. Após alguns segundos, o Mocking service será iniciado, criando uma nova URL para a baseUri.</p>	

A partir desse momento, o **Mocking service** está ativo e pode ser utilizado como se fosse uma API disponível.

Etapa 8: Testar o Mocking service

Chegou o momento de testar a API usando o **Mocking service**. Podemos realizar o envio de uma solicitação e ver a resposta da API.

<p>Observe o RAML Viewer no lado direito da tela.</p> <ol style="list-style-type: none">1. Clique em Try it na parte inferior do RAML Viewer.	 <p>The screenshot shows a RAML snippet with the following content:</p> <pre>quantity: 1 - product_id: "122" rel_product: http://alc.mulesoft.com/api/products/product/122 quantity: 1 shipping: shipping_address_id: "1122" rel_address: http://alc.mulesoft.com/api/users/m.../vin_user_8/add</pre> <p>At the bottom, there are three tabs: RAML, OAS, and Try It. A red arrow points to the 'Try It' button.</p>
<ol style="list-style-type: none">2. Digite um valor qualquer em order_id e envie a requisição através do botão Send.	 <p>The screenshot shows the Mocking service interface. At the top, there are buttons for '</>Edit RAML' and 'Download', and a 'Mocking service:' toggle. Below, the 'Request URL' is set to 'https://anypoint.mulesoft.com/mocking/api/v1/links/bb8'. Under the 'Parameters' tab, 'URI parameters' are listed, with 'order_id*' having a value of '100'. A red box highlights the '100' value. At the bottom, there is a blue 'Send' button with a red arrow pointing to it.</p>
<ol style="list-style-type: none">3. Abaixo observamos a resposta 200 OK com a resposta da chamada no formato JSON.	 <p>The screenshot shows the Mocking service response. At the top, it displays '200 OK' and '864.00 ms'. Below, there is a JSON response:</p> <pre>{ "order_id": "order_22", "creation_date": "2015/12/31 10:00:00", "-items": [Array[2]] -0: { "product_id": "121", "rel_product": "http://alc.mulesoft.com/api/products/product/121",</pre>

Concluimos a criação de uma especificação de API e usamos o **Mocking services** para disponibilizar uma simulação do retorno das chamadas.