Your journey to building APIs starts here! Learn the essentials of using fastn.ai API in this comprehensive tutorial. Take control and become an API pro!😎 Follow these simple steps to create your API: From the navigation bar, click on the "API" section. Alternatively, click the "+" icon on your dashboard. The Fastn team is here to help! We've prepared a handy ready-to-use API for you to explore, or if you're feeling adventurous, you can build your own from scratch! It's time to name your API route! Add the type and the endpoint if necessary. Let's create a GET route using the REST method.

Setting up your API's configuration is indeed the first step. Curious about its importance? It helps structure the input and output data, ensuring you know exactly how your data will be displayed. You can begin by exploring existing models or create new ones. For more information, check out the Models tutorial. In our case, we don't need input models; we can create the structure directly for the output by defining models from And add title and price.

The datasource is the heart of our API, where you can utilize existing platform APIs or create new ones to use. We're going to create a 'get product' API. First, drag and drop the Datasource element to the canvas. Next, we'll see all the datasources that we have. Go to "My Datasources." Add new datasource. You need to include some details like the method and your URL. In our case, we used the fakestoreapi. There are some additional things you can learn about in the Create datasource section for more information. Do not forget to click your datasource. Next, we'll show you the mapper, where you can view the input from the datasource and map it to other data formats! What's the purpose? In this simple example, you may not need it, but later, when we delve into more complex examples, understanding this will be valuable. Read more about it in the mapper tutorial. As you can see, we can't find any data input to map over our datasource structures. If we need to, for example, add some taxes to our prices, we'll need to include a mapper/function element. Click "Next," then add a name to your datasource. The rest is configuration, which you can learn about in the API datasource configuration.

Let's speed up debugging. Click on the debugging CTA. Map your Data Overview Deal With Mapper Debug Your API Overview In this tutorial, we will learn how to deal with the Mapper and how to map the data from one Flow Node to another. In the last tutorial, we stopped at creating our Datasource called 'getproductsFromfakestore' from the Fake Store. Deal With Mapper Data Mapping is the process by which you transfer data from one step to another or from one node to another. What we need to do now is add the taxes to the product price. So far, these are the steps we have. There are two steps to add the taxes: one is by making it dynamic, where you will pass through the Header or by creating the variable and making it static. We are going to use the first step, making it dynamic. To do that, we need to click into our old step, getProductsFromfakestore. Add the taxes value and map the data from input.taxes. This value will actually come from the variable called taxes through the input and assign or mapping into our variable called taxes too, do not forget to save and name your step. Let's create our expression. First, drag the data mapping tool. You'll notice that the design is very similar to the datasource mapping steps. Yes, it's the same because we don't know when you'll need it. If you open the code test, you'll see it. You will see the JSON format that we already have from the first node inside the steps. You'll also see our previous step getProductsFromfakestore. We'll create

an object containing numbers to represent our expectations. Notes Avoid the spacing between the words “TotalPrice” will make it easier for you to call it in the mapper. Here in fastn, to create a mathematical expression or any expression related to the data, we need to use the action feature. Here we can Multi our Price with 0.5, which it is be our Taxes rate. To do that, we need to create an action and choose the Math expression option. Then, select the steps and the key with which we need to perform the expression, like this. Make sure that the output is in Float. You can Learn more about Action. Save that, then run the test again. You'll note that the expression is added and the result is correct. Let's name our steps and begin debugging.

It's time for debugging. Let's see if everything goes correctly. Click on the debugging button. Then run the test. The debugging section will appear at the bottom of the page, with the steps or nodes colored. If both steps are white without errors, it means that they have passed successfully. The last step will be passed as the result. Now, let's write the taxes value in the input to pass the value to step 1, and test the result by clicking on the run button. The result of this expression in step 2 will show as "Total Price: 109.95." If you want to see the default value in the Price from our previous steps, you can click on Mission Success. Congratulations, you created your dynamic taxes and learned how to use the inputs and mapping. If you want to make the output for a specific structure, in the next tutorial, we'll see how to do that.

We want to make sure that our documentation is easy for you to use. However, we know that sometimes you might need extra help. Here are several ways you can get in touch with us if you're facing any issues. Email: To contact us, simply send an email to us at contact@fastn.ai. We'll get back to you as soon as we can.