**Tutorial 44: Where to Fetch Data**

In the tutorial we are going to focus on fetching the data from the endpoint from the JSON Server.

Make sure **Tutorial 43** is complete to be able to move on to this tutorial or clone the [vue3-firebase-course](https://github.com/younggrasshopperza/vue3-firebase-course) repository and go to branch called **Tutorial-43** and navigate to the young-grasshopper-jobs folder.

**Step 1:** In the Jobs.vue file, in the data property, remove the objects inside the jobs data property, make sure that the jobs data property is an empty array.

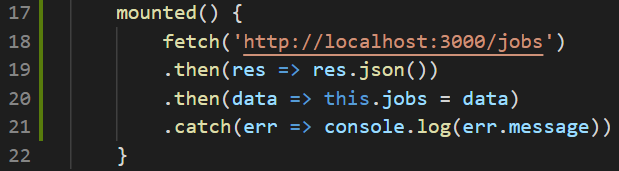
Look at the example below (**Line 14**):



**Step 2:** In the Jobs.vue file, create mounted lifecycle hook, remember the mounted component executes when the component mounts to the dom which is when we want to fetch the data.

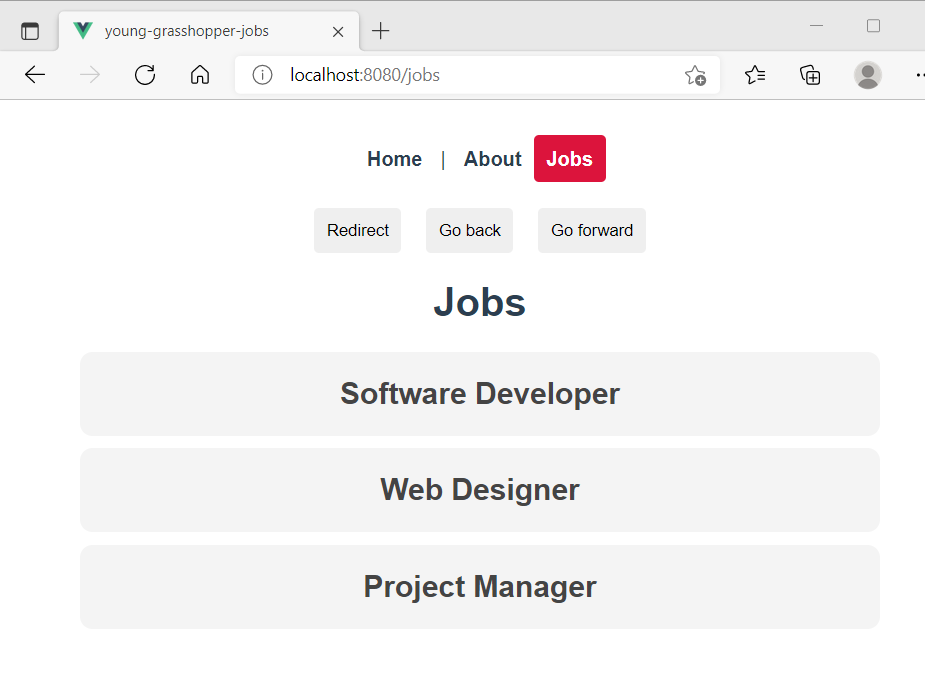
We will be using the fetch API. Inside the mounted component call fetch function and pass the JSON server endpoint as an argument in the fetch function. The fetch returns a promise which we can use to create a then method. We get a response and we need to use the JSON method on it. We then get another callback function and create another then method, that gets access to the actual passed data, assigned the data we receive to the jobs data property. We will call an error method to check if there are any errors. The argument can provide a message of the error.

Look at the example below (**Between** **line 17 and line 22**):



The data is still displaying as normal and we can see the job titles on the page.

**Output:**



**Step 3:** In the JobDetails.vue file, we want to be able to see all of the information related to that jobs stored in the JSON file. Create a mounted lifecycle hook. Create a fetch request that calls the individual job. In order for us to do that we to type forward slash and then whatever the id of the job is. Remember we have a prop called id. Pass the prop as a parameter at the end of the URL.

We then need to create a data property called job saved the data we get back from fetch to job.

Look at the example below (**Line 10**):



Currently the screen will be blank because when the component renders on the page by default the value will stay null until the fetch executes. By then it would already be showing as null on the screen because of the delay.

In the next tutorial we will work on conditionally showing data

**You can view the code snippets from the** [**vue3-firebase-course**](https://github.com/younggrasshopperza/vue3-firebase-course) **repository under the Tutorial-44 branch**.