

Retrieve Data from API(.json)

In this case we are not making an api, we are practicing on how to retrieve data from an api, hence we are using an api placeholder; **JSONPlaceholder**.

The aim for the lesson is fetch the post in the json file, and display all the titles.

NOTE On Start of new project make sure npm run watch is running, so as to compile the js.

```
npm run watch
```

Instructions

1. Install **axios**

Note on its own, does not make api calls, therefore we are going to use axios(its a library)

To install axios, run

```
npm install axios
```

2. To pull in the axios into the project.

We pull it globally into our ./src/app.js

```
window.axios = require('axios');
```

To make something attached to our entire js environment we attach it to window, window can be thought as the lowest js environment.

3. Mounting the data/pulling in the data from the json.

In vue life-cycle, **mounted** is the only life-cycle thing needed, **mounted** is called when the vue component is ready to go, hence it does not hold up the operation of the other parts of the component.

mounted is placed anywhere in the root instance, or also in the component.

```
mounted: function(){  
  
},
```

a) To make the request, to get the api

in the mounted function;

```
axios.get('https://jsonplaceholder.typicode.com/posts')
```

this retrieves the **posts** resources into our project.

It does not hold up the operation of the other parts of the component. It pulls in the resources in the background, and once the required resources are pulled in, the next function in mounted is carried out.

b) To check if we got the data.

Resources we get is in form of a **response**, hence to confirm that we received the response,

```
.then(response => console.log(response));
```

through the console.log we are returned a sub-property of **data**

c) Prepare **posts** variable in our data

```
data: {  
  posts: null,  
},
```

d) Pass the data into our posts array.

The **posts** array contains object arrays, in our case we are interested in the data array.

```
.then(response => this.posts = response.data)
```

e) To display the pulled in data. i.e since the data is an object array on its own, we are to display the titles part of it

```
<li v-for="post in posts" v-text="post.title" ></li>
```

f) Other functionality of axios,

To catch errors to fetching of the resources,

```
.catch(error => this.posts = [{title: 'No post Found'}])
```

To finalize

```
.finally(() => console.log('Data Loading Complete'));
```

The ./src/app.js file

```
import Vue from 'vue';  
  
window.axios = require('axios');  
  
new Vue({  
  el: '#app',  
  
  components: {  
  
  },  
  
  mounted: function(){  
    axios.get('https://jsonplaceholder.typicode.com/posts')  
      .then(response => this.posts = response.data)  
      .catch(error => this.posts = [{title: 'No post Found'}])
```

```
        .finally(() => console.log('Data Loading Complete'));
    },

    data: {

        posts: null,

    },

    //To carry out computations
    computed: {

    },

    methods: {

    },

    },
}
})
```