# Weather API call:

We will create a component that will call a weather API. We will NOT format the data just prove that we can call it.

We will use JavaScript FETCH/PROMISE (no Axios library)

1. Create a **WeatherReport.vue** component:

```
<template>
  <div>
    <h2>Weather:</h2>
    <p>{{ weather }}</p>
    <div>

    </div>
  </div>
</template>

<script>


export default {
  name: "weather-report",

  data() {
    return {
      weather: ''
    };
  },
  created() {

fetch('http://api.openweathermap.org/data/2.5/weather?q=London,uk&APPID=2f7bdcc26bc9669ab55f8341c4521509')
  .then((response) => {
      if (response.status !== 200) {
        console.log('Looks like there was a problem. Status Code: ' +
          response.status);
        return;
      }

      // Examine the text in the response
      response.json().then((data) => {
```

```
        console.log(data);
        this.weather = data;
      });
    }
  )
  .catch((err) => {
    console.log('Fetch Error', err);
  });
  },
};
</script>

<style>
</style>
```

2.  Create a Weather.vue  view and import the WeatherReport component

```
<template>
  <div>
     <weather-report />
  </div>
</template>

<script>

import WeatherReport from '@/components/WeatherReport.vue';

export default {
    components: {
        WeatherReport
    }
}
</script>

<style>

</style>
```

3. Now set up the route:

```
import Weather from '../views/Weather.vue'

  {
    path: "/weather",
    name: "weather",
    component: Weather,
    meta: {
      requiresAuth: false
    }
  },
```

4. Now set up the Menu's Router Link:

```
<router-link v-bind:to="{ name: 'weather' }">Weather Report</router-link>
```

Now the menu should return the weather and display….

**Call a Web Service with API key using Axios:**

## We will use NASA's pic of the day API

api.data.gov
`Gt53JZCtJPHVfecMOgQg4n0fRAW64TBPBrydnl4W`

1. Create an Axios service call: NASAService.js

```
import axios from 'axios';

const apiKey = '?api_key=Gt53JZCtJPHVfecMOgQg4n0fRAW64TBPBrydnl4W';

export default {

    getNASAData(){

        return axios.get('https://api.nasa.gov/planetary/apod' + apiKey);
    }
}
```

2. Create **NASAData.vue**

```
<template>
  <div>
     <button v-on:click="getData" >Click Here To Win A Prize </button>
     {{info}}

     <p>
        <img v-bind:src="info.url">
     </p>
  </div>
</template>

<script>

import nasaService from "../services/NASAService.js";
export default {
   name: 'nasa-data',
    data() {
       return {
         info: ''
       }
    },
    methods: {
       getData() {
        nasaService.getNASAData().then(response => {
       this.info = response.data;
      })
       }
```

```
    },
}
</script>

<style>

</style>
```

3. Create **Nasa.vue** view page

```
<template>
  <div>
     <nasa-data />
  </div>
</template>

<script>

import NasaData from '@/components/NasaData.vue';
export default {
    components: {
        NasaData
    }
}
</script>

<style>

</style>
```

4. Add a route

4. Add to Nav bar….

**Authentication:**

```
      <div id="nav">
        <router-link v-bind:to="{ name: 'home' }">Home</router-link> | 
        <router-link v-bind:to="{ name: 'search' }">Home Search</router-link> | 
         <router-link v-bind:to="{ name: 'addHome' }">Add Home</router-link> | 
         <router-link v-bind:to="{ name: 'weather' }">Weather Report</router-link> | 
         <router-link v-bind:to="{ name: 'login' }" v-if="$store.state.token == ''">Login</router-link>
        <router-link v-bind:to="{ name: 'logout' }" v-if="$store.state.token !=
''">Logout</router-link>
      </div>

      <p v-if="$store.state.token != '' ">Welcome {{$store.state.user.username}}</p>
   </div>
```

CORS