

## USER STUFF FIRST:

How could we add a welcome Steve (user) message to the page when a user logs in?

Let's add this to the header..

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

1. What happened? How could we fix this?

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

Don't forget to lock down the controller with the authentication checks and role privileges using the `@PreAuthorize("isAuthenticated()")` and `@PreAuthorize("PermitAll")` and `@PreAuthorize("hasRole('Admin')")` ... Review Module 3 material if you need a refresher... See Authentication and Authorization lecture code or capstone...

## Calling a 3rd party web service with an API key....

1. Using <https://www.weatherbit.io/api/weather-forecast-16-day>

Test using the browser:

[http://api.weatherbit.io/v2.0/forecast/daily?key=db76e9cee13649cfb36dad82389a6671&postal\\_code=43035&country=US](http://api.weatherbit.io/v2.0/forecast/daily?key=db76e9cee13649cfb36dad82389a6671&postal_code=43035&country=US)

2. Signed up for the free account and got an API key
3. KEY: db76e9cee13649cfb36dad82389a6671

#### 4. Create a WeatherService.js

```
import axios from 'axios';

const BASE_URL = 'http://api.weatherbit.io/v2.0';
const API_KEY = 'key=db76e9cee13649cfb36dad82389a6671'
const UNITS = 'I'

export default {
  getForecast(zip, countryCode) {
    return
    axios.get(`${BASE_URL}/forecast/daily?${API_KEY}&postal_code=${zip}&country=${countryCode}&units=${UNITS}`);
  }
}
```

#### 5. Create a DailyWeather.vue component in the Components directory

#### 6. Now we build the first part of the template that has a search form for zip code

```
<template>
  <div class="search-field">
    <label for="zip">Search For Weather By Zip Code: </label>
    <input type="text" v-model="zip" placeholder="Enter zip code!" />
    <div class="btn-container">
      <button type="submit" class="search-btn">
        Fetch!
      </button>
    </div>
  </div>
</template>
```

#### 7. Now we are going to add a <script> section:

Zip: we will v-model to this from the form

Result: will hold the result object from the web service

```
<script>
```

```

import WeatherService from "../services/WeatherService.js";

export default {
  name: "daily-weather",
  data() {
    return {
      zip: "",
      result: {},
    };
  }
}

</script>

```

8. Then dump in some <style> stuff. Explain that not a lot of thought went into this, I just copied it.

```

<style>
.btn-container {
  display: flex;
  flex-direction: row;
  justify-content: center;
  align-items: center;
  margin-top: 20px;
}
.search-btn {
  color: #fff;
  border-color: rgba(90, 90, 90, 0.75);
  background-color: rgba(45, 4, 61, 0.5);
  border-radius: 10px;
  font-size: 30px;
  text-shadow: 2px 2px rgba(0, 0, 0, 0.8);
  width: 200px;
  height: 60px;
  box-shadow: 1px 1px rgba(0, 0, 0, 0.5);
  transition: all 0.2s ease-in-out;
  display: flex;
  flex-direction: column;
  margin-left: auto;
  margin-right: auto;
  text-align: center;
  align-items: center;
  justify-content: center;
  margin-bottom: 40px;
  margin-right: 5px;
}

```

```

.search-btn:hover {
  box-shadow: 10px 10px rgba(0, 0, 0, 0.5);
  transform: scale(1.05);
  background-color: rgb(0, 175, 0);
}

.search-field {
  margin-top: 20px;
  margin-bottom: 20px;
  font-family: "Miriam Libre";
  font-size: 25px;

  display: flex;
  flex-direction: column;
  justify-content: center;
  align-content: center;
  align-items: center;
}
.search-field label {
  margin-top: 20px;
}
.search-field input {
  width: 518px;
  height: 40px;
  border-radius: 5px;
  font-size: 17px;
}
</style>

```

9. Now we are going to go build a function that is going to call the web services,. We will load it on the click of the button... so we will put it in the methods sections

```

methods: {
  getWeatherForecast(zip) {
    WeatherService.getForecast(zip, "US")
      .then((response) => {
        this.result = response.data;
        console.log(this.result);
      })
      .catch((error) => {

```

```

    if (error.response) {
      this.errorMsg =
        "Error searching. Response received was: " +
        error.response.statusText +
        "!";
    }
    else if (error.request) {
      this.errorMsg = "Server unavailable";
    }
    else {
      this.errorMsg = "General Error";
    }
    console.log(error.status);
  }); //end of catch
}, //end of get weather forecast method
} //end of methods section

```

10.

11. Now that we have the method, we can add it to the button as follows:

```
@click.prevent="getWeatherForecast(zip)"
```

12. Now let's go add the DailyWeather component to the Home page

```
import DailyWeather from '@components/DailyWeather';
```

Add to components  
Add to template

13. Now let's verify everything is working....

Is the form on the homepage?  
Can we see the response printing out in the log?

Can we see the response in Google Dev Tools → Network

14. Now Let's build the results container using the sample JSON from [http://api.weatherbit.io/v2.0/forecast/daily?key=db76e9cee13649cfb36dad82389a6671&postal\\_code=43035&country=US](http://api.weatherbit.io/v2.0/forecast/daily?key=db76e9cee13649cfb36dad82389a6671&postal_code=43035&country=US)

Start by adding in just the city/state

```
<div v-if="result.data">
  <div>City/State: {{ result.city_name }} , {{ result.state_code }}</div>
  <div v-for="forecast in result.data" v-bind:key="forecast.ts">
    <div>Date: {{ forecast.datetime }}</div>
    <div>-- High Temp: {{ forecast.high_temp }} F</div>
    <div>-- Low Temp: {{ forecast.low_temp }} F</div>
  </div>
</div>
```

Now DEBUG and Troubleshoot!