



Dynamic Page Content

This weather site assignment has two major parts. The first part of the assignment follows the activity that you completed in the *Prepare* module of this lesson by reading [JSON data from an external source](#) to populate a weather site **home** (index.html) page with town data such as population and a list of events. The second part has you gather current weather data, in JSON format, from a free weather API at Weather Underground to use instead of the static data we provided as placeholders for the high temperature, etc., on the Franklin page.

Requirements

- ☐ Create the weather site home page (*index.html*) using the [weather site planning document](#) as a guide. Store this page in the *lesson-9* folder of your *assignments* directory and be sure to include the supportive folders and files that support this weather site assignment update.
- ☐ Read in the town statistical data for all three towns from the given JSON data source. Display the individual town data with the **motto**, **year founded**, **population**, and **annual rainfall** from the JSON file on your home page. Use the lesson's activity as your guide to read in the data and display on the page. The town data location is:

`https://byui-cit230.github.io/weather/data/towndata.json`

Hint: You can manipulate the DOM (edit the HTML page) by placing the content directly into the page using the `getElementById()` method or similar and affecting the `innerHTML` attribute.

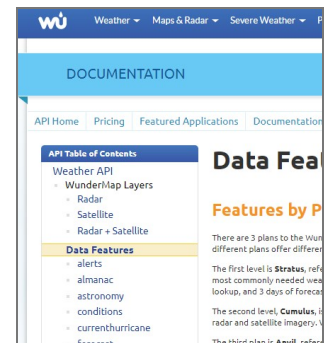
• **Challenge:** List this same town data on the town's respective page in addition to the home page and include the town events.

- ☐ Get the current weather data for Franklin (use Franklin, MN) from Weather Underground and display the results in appropriate locations on the page that fits your design and interface. You will be asked to have at least five, live data elements displayed on the page from Weather Underground.

Additional Help: The following video by Dr. Paul Cheney will help you understand the process of using AJAX to retrieve JSON data at weather underground.

 [Use AJAX to access JSON data from the weather underground](#)

- Go to [wunderground.com](https://www.wunderground.com) and register for a Basic (free) account.
- Once registered, log in using your username and password.
- [Apply for an API key](#) for a "Stratus Developer" plan (this should be free).
- Make sure you read the "API Terms of Use" agreement, particularly the first item under "Conditions and Restrictions" and everything under "Rate Limits; Overage Policy."
- Once you have received your API key, make a copy of it and store it where you can access it easily on your own computer.
- Go to



<https://www.wunderground.com/weather/api/d/docs> which documents how to get started and shows examples of the response data in **JSON** format. In the **API Table of Contents** on the left, click **Data Features** to see what weather features the *Stratus plan* provides. Note how to combine features into a single request under the *Standard Request URL Format* section.

- We will be using the **conditions** and **forecast** features. Select these menu items and then scroll to the bottom of the page to view the example JSON formatted response using the *Show Response* button.

ob_url

Examples

`http://api.wunderground.com/api/Your_Key/conditions/q/CA/San_Francisco.json`

Show Response

The example given is for San Francisco, California. We will use **Franklin, MN** for our Franklin town page.

- Now that you have reviewed the basics of the API, create a JavaScript file named `wu-weather.js` and save that file in your scripts folder within lesson-9. Use an `XMLHttpRequest` object to request the JSON content (see example below) of the the Weather Underground API and get the following data:

1. The current weather string, e.g., Partly Cloudy, from conditions.
2. The current temperature in Fahrenheit from conditions.
3. The current wind speed in miles per hour from conditions.
4. The image icon representing the current weather from conditions.
5. The text description forecast for the most current period in Fahrenheit, e.g., Partly cloudy. Lows overnight in the upper 70s.

```
1 var requestURL = 'http://api.wunderground.com/api/ ...| fill in the correct URL components ... /MN/Franklin.json';
2 var request = new XMLHttpRequest();
3 request.open('GET', requestURL, true);
4 request.send();
5
6 request.onload = function() {
7   var franklinWeather = JSON.parse(request.responseText);
8   document.getElementById('desc').innerHTML = franklinWeather.current_observation.weather;
```

Assessment

- Update your assignment portal page to link to this weather site home page.
- Update your remote repository accordingly.
- Submit your URL in the assignment drop-box.

Assessment Criteria

30 points possible - Weather Site Project Category

- Ensure that the home page is semantically structured and that all HTML markup and CSS are valid.
- PARC and typography design principles are evident in all views.
- The page load weight does not not exceed 400 Kb.
- The site is responsive in all three views and has an acceptable layout given the planning document sketches.
- The town data is retrieved and dynamically displayed on the home page for all three towns.
- The current weather data is consumed from Weather Underground for the the Franklin town page with all five, required data elements.
- There are no spelling or grammatical errors.