

Assignment: [Tutorial] Make an AJAX Request

Due Date: June 17th by 11:59pm

Points: 10

Description: Let's write a simple web app that will display the most salient forecast details for us. You should have two files open in the first panel - an *index.html* file that sets up a simple web page, and a blank *forecast.js* that will (once we're done), display the forecast on the index page.

Overall Summary:

The screenshot shows the Codio web interface. At the top, there's a navigation bar with 'Codio', 'Project', 'File', 'Find', 'View', 'Tools', 'Education', and 'Help'. Below this is a browser-like address bar showing 'codio.com - ...' and a 'Guide' tab. The main URL is 'https://codio.com/home/student/feedback/nboonyasiti/f761e96808b2252d8f4a6de2233b3a55'. The page title is 'Web Development / Week 2'. The main content area features a card for 'Making an AJAX Re...' from Nina Boonyasiti. It shows a 'Grade' of 100, 'Points (%)' of 100, 'Graded' status with a green circle and '2', 'Answered' status with a green circle and '2', and 'Assessments' with a grey circle and '2'. Below this card are tabs for 'Assessments', 'Comments', and 'Code Comments'. The 'Assessments' tab is active, showing a table with two rows of assessment data.

Section	Assessment Name	Points	Type	Correct
Working with the Weather Data	Check displayForecast() details	1	★	✓
XmlHttpRequest	Check useXHR() details	1	★	✓



Student Submission:

displayForecast takes in data as a parameter, parses the data and then creates an "img" and "p" element. These elements are appended to display, which is identified as "forecast". The useXHR() function is used as an AJAX Request to the HTTP to request for the most up-to-date information for the weather app. In the window.addEventListener, we specify that these updates are to be done by the hour.


```
index.html  forecast.js  x
1  function displayForecast(data) {
2      var current = data.properties.periods[0];
3
4      var display = document.getElementById("forecast");
5      display.textContent = "";
6
7      var icon = document.createElement("img");
8      icon.src = current.icon;
9
10     var desc = document.createElement("p");
11     desc.textContent = current.detailedForecast;
12
13     display.appendChild(icon);
14     display.appendChild(desc);
15 }
16
17 function useXHR() {
18     const xhr = new XMLHttpRequest();
19     xhr.addEventListener('load', () => {
20         displayForecast(JSON.parse(xhr.responseText));
21     });
22
23     const url = "https://api.weather.gov/gridpoints/
TOP/48,54/forecast";
24     xhr.open("GET", url);
25     xhr.send();
26 }
27
28 // useXHR();
```

```
index.html  forecast.js  x
17 function useXHR() {
18     const xhr = new XMLHttpRequest();
19     xhr.addEventListener('load', () => {
20         displayForecast(JSON.parse(xhr.responseText));
21     });
22
23     const url = "https://api.weather.gov/gridpoints/
TOP/48,54/forecast";
24     xhr.open("GET", url);
25     xhr.send();
26 }
27
28 // useXHR();
29
30 // Or we can use the following for multiple script
files
31 // window.addEventListener('load', function() {
32 //     useXHR();
33 // });
34
35 // Option 3: invoke useXHR() and update it at one
hour intervals!
36 window.addEventListener('load', function() {
37     useXHR();
38     const oneHour = 60 * 60 * 1000;
39     setInterval(useXHR, oneHour);
40 })
41
```

Result:

  <https://annuallocal-optionshake.codio.io/weather-demo/index.html>

Weather




Mostly sunny, with a high near 92. Southeast wind 10 to 15 mph.

Auto-Grader Response:

3. Working with the Weather Data

Now we're ready to fetch our data and display the forecast!



Check displayForecast()

Check It!

✓ LAST RUN on 6/16/2022, 4:10:06 PM

✓ `displayForecast()` adds an `` element with `src` attribute set to `data.icon` to `<div id="forecast">`

✓ `displayForecast()` adds a `<p>` element with `textContent` attribute set to `data.detailedForecast` to `<div id="forecast">`

4. XMLHttpRequest



Check useXHR()

Check It!



LAST RUN on 6/16/2022, 4:02:16 PM

✓ Expected and event listener to be attached to the `XMLHttpRequest` object for the `load` event

✓ Expected an instance of `XMLHttpRequest` to be constructed

✓ Your `XMLHttpRequest.open()` should use the `GET` HTTP Method

✓ Your `XMLHttpRequest.open()` should use the url `https://api.weather.gov/gridpoints/TOP/48,54/forec`

✓ Expected `XMLHttpRequest.send()` to be invoked

✓ Expected `XMLHttpRequest` methods to be called in the order `addEventListener()`, `open()`, `send()`