

IMY 210

Theme Assignment 6: JSON and AJAX

This practical aim to provide you with a better understanding of JSON data structure as well as provide a basic framework for creating AJAX calls using both XML (rss format) and JSON.

Provided files

update.rss – A demo file to access XML data

update.json – A demo file to access JSON data

screenshot.png – A demo of the output

Task

1. In this practical you create a simple function that will update a page occasionally from data found in an XML and JSON file.

2. Creating an **asynchronous** method to get data from an XML file:

Start by creating a JavaScript function `getUpdateXML()`.

Create an `XMLHttpRequest` object.

Set the `onreadystatechange` of the `XMLHttpRequest` object as a function that does the following:

- i. Add a conditional statement that test for the object's `readyState` and `status`.
- ii. If the `readyState` is equivalent to `4` (the request has been completed) **AND** the `status` is equivalent to `200` (HTTP status of OK) ...
- iii. ... retrieve the document with `xml.responseXML` and process the document accordingly to be displayed.

* You can check [this resource](#) to understand more about and how to use `responseXML`.

* To access the data in the XML file you can look at the HTML DOM functions (`getElementsByTagName` and `getElementById`). A good resource is the [Mozilla developer API](#) under web technologies.

* The most ideal way of printing large chunk of data, to construct the format on how it should be displayed before appending it to the HTML with the `innerHTML` found as part of the HTML DOM.

Outside the `onreadystatechange` function, create an `open` call with the `XMLHttpRequest` object. Pass the variable "GET" the provided "update.xml".

Call the `send` function with the object.

3. Creating an asynchronous method to get data from a JSON file:

- a. Everything will be the same from *Acquiring data from XML* with a few minor differences:
 - i. Instead of using `getElementsByTagName`, you can access the json object's nodes directly.
e.g. `jsonObject.items[0].title`;
 - ii. Remember to change the file being `opened` by the `XMLHttpRequest` object.
 - iii. Before `sending` the request object, add a `responseType` to the object and set it as json.
e.g. `xmlhttprequestObject.responseType = 'json'`;

4. Lastly, you can simply test this by linking the function to a button in HTML **but**, for this practical we will run this update function automatically.

In any regular instances we can simply call this function after a user action or page update **but**, for this practical will simply run a time out and link the update function within our timeout.

- a. A timeout can be created in JavaScript by setting an interval.
e.g. `windows.setInterval(do me!, duration in milliseconds);`
- b. We can pass a function to the interval as a variable.
e.g. `{function(){ myFunctionName(); }, 1000};`

Bonus: Make it pretty. (CSS,1994)

Submission

- **Compress** your final HTML file (TA6.html) into an archive named **TA6.zip**.
- Submit your ZIP file to the upload slot provided on clickUP before the deadline.
- Ensure that you have a backup of your final submission.