

Pulling Down Tweets:

I did this in PHP. I ended up using this guy's twitter feed parser.

<http://jonathannicol.com/blog/2010/10/07/display-recent-twitter-tweets-using-php/>

Twitter has its own API, so in order to interact with it, I had to create and use my own API key.

The Http magic takes place in the following code. (This can all be found underneath the "Request Twitter Timeline" comment in the TweetPHP.php file)

```
$response_code = $this->tmhOAuth->request('GET', $this->tmhOAuth->url('1.1/statuses/user_timeline.json'), $params);
```

This sets the response_code variable which is immediately checked against 200. If the response_code is equal to 200, it means that the request was successful. If the request was successful, then we iterate through the tweets and process them.

The file that actually pulls content from twitter into the webpage is the pullingDown.php file. Here, I just call the TweetPHP constructor and set all the requisite keys and the screen name of the person whose tweets I want to pull. I chose Darth Vader for this example. Then I use echo to output the results into the webpage.

JSON Language File:

I did this one in JavaScript. I referenced a lot of sources in order to figure this one out. Here are the main ones.

http://www.w3schools.com/json/json_http.asp

<http://jaskokoyn.com/2013/07/22/nesting-objects-and-arrays-json-tutorial/>

<http://stackoverflow.com/questions/2488148/parse-2-dimensional-json-array-in-javascript>

<https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest>

The XMLHttpRequest used in this one makes it possible to retrieve the JSON data from the URL.

Using the XMLHttpRequest, I defined a function that uses the onreadystatechange property, meaning that the function will call whenever the readyState attribute changes.

I then use an if statement which asks if the readyState property is equal to 4, meaning that the operation is complete. It then checks if status property is equal to 200, meaning that the request was successful.

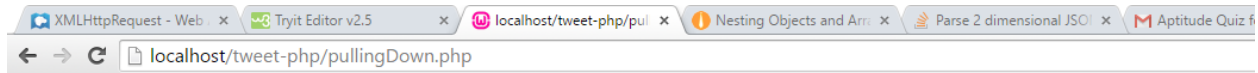
I then set a variable to the parsed JSON Object and call another function on it, which simply iterates through the JSON object and outputs the languages into the html of the webpage.

I then call open on the XMLHttpRequest, which initializes the request, and then I call send, which sends the request.

Other:

I used wamp server to develop and test this code. This all runs great on my localhost. Here are some screenshots of how it looked for me.

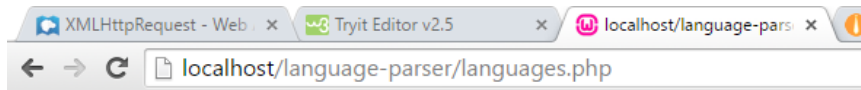
Tweets:



Latest tweets

- To the haters: The Supreme Court just altered the deal, pray they don't alter it further. [#LoveWins](#)
- With a special appearance by your favorite Sith Lord - <https://t.co/ro9bq90rf5>
- Palpatine / Vader 2016
- Celebrating Sitho de Mayo w/ the 501st today by blowing up a small rebel enclave. I love a good party. [#RevengeOfTheFifth](#)
- Arrived too late to witness my evil Empire trounce [#RedSox](#), but enjoyed my [@NESN](#) interview none-the-less <http://t.co/LCuiDFXzEJ>
- I find your lack of gifts disturbing.
- I want those rebels dead, not excuses - <http://t.co/IXXV1cswIT>
- "Preparing for shipment", excellent. Glad I went with the Sith Gray model [#StarWars](#) <http://t.co/7Govwf3saK>
- A cuter son I never did see. The kid w/ the ice cream isn't bad either [#StarWarsCelebration](#) [#StarWars](#) <http://t.co/FLEZGiQyxHâ€>
- Not going to lie, Iâ€™ve looked better. [#StarWars](#) <http://t.co/jZurFXNQqr>

Languages:



Arabic
Catalan
Czech
Danish
German
Greek
English
Spanish
Finnish
French
Hungarian
Indonesian
Italian
Hebrew
Japanese
Korean
Malay
Dutch
Persian
Polish
Portuguese
Russian
Swedish
Thai
Turkish
Chinese