

Hui Xin Data Collection 3/13/20

Notebook: Animation Templates

Created: 3/13/2020 11:50 AM

Author: Alan Tram

URL: <https://stackoverflow.com/questions/25515936/perform-curl-request-in-javascript>

Updated: 3/15/2020 6:55 AM

JavaScript/JSON Methods/Libraries to Know:

Perform **curl** request in javascript?

- <https://stackoverflow.com/questions/25515936/perform-curl-request-in-javascript>

Let in JS

- <https://stackoverflow.com/questions/762011/whats-the-difference-between-using-let-and-var?page=1&tab=votes#tab-top>

JSON.parse()

- A common use of JSON is to exchange data to/from a web server
- When receiving data from a web server, the data is always a string
- Parse the data with JSON.parse(), and the data becomes a **JS object**

```
'{"name":"John", "age":30, "city":"New York"}'  
  
// Use the JavaScript function JSON.parse() to convert text into a JS object  
  
var obj = JSON.parse('{"name":"John", "age":30,"city":"New York"}');
```

JSON.stringify()

- Data to a web server must be as a string

```
var obj = { name: "John" , age: 30 ,city: "New York"};  
  
//convert to string  
  
var myJSON = JSON.stringify(obj);  
  
// myJSON is now the obj as string. Send to server  
  
document.getElementById("demo").innerHTML = myJSON;  
  
Output:  
  
{"name":"John","age":30,"city":"New York"}
```

```
//Stringify a JS Array  
  
var arr = [ "John", "Peter", "Sally", "Jane" ];  
  
var myJSON = JSON.stringify(arr);  
  
document.getElementById("demo").innerHTML = myJSON;  
  
output:  
  
["John","Peter","Sally","Jane"]
```

Fs (File System) module

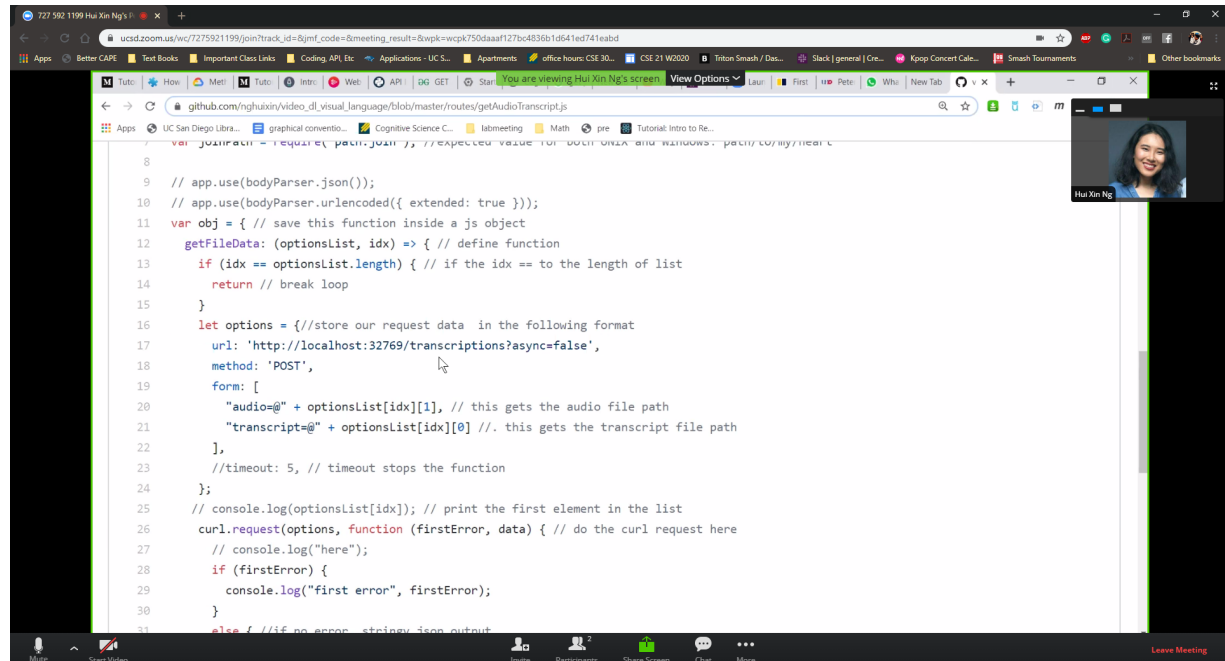
- Node.js includes fs module to access physical file system. The fs module is responsible for all the asynchronous or synchronous file I/O operations.
- <https://www.tutorialsteacher.com/nodejs/nodejs-file-system>

Good Coding Practice:

Code basic example of your intended project and confirm your knowledge of the documentation by seeing if your basic example works.

Then code your actual project

getTranscripts.js:

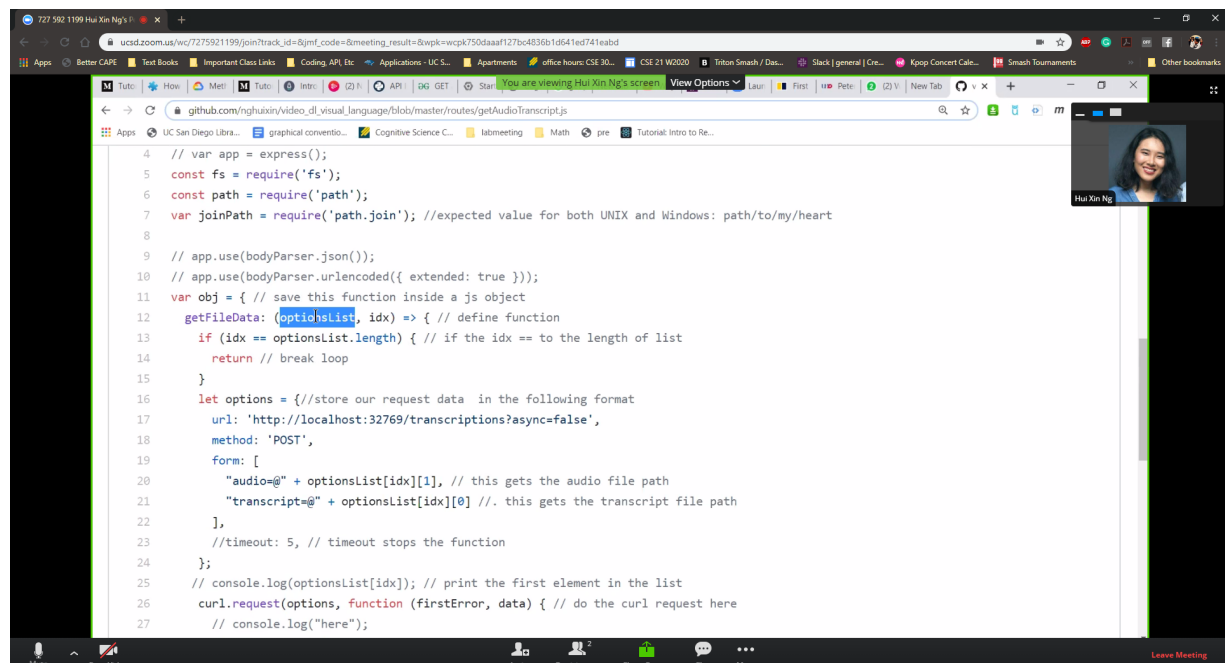


```

8
9 // app.use(bodyParser.json());
10 // app.use(bodyParser.urlencoded({ extended: true }));
11 var obj = { // save this function inside a js object
12   getFileData: (optionsList, idx) => { // define function
13     if (idx == optionsList.length) { // if the idx == to the length of list
14       return // break loop
15     }
16     let options = { //store our request data in the following format
17       url: 'http://localhost:32769/transcriptions?async=false',
18       method: 'POST',
19       form: [
20         "audio=@" + optionsList[idx][1], // this gets the audio file path
21         "transcript=@" + optionsList[idx][0] // this gets the transcript file path
22       ],
23       //timeout: 5, // timeout stops the function
24     };
25     // console.log(optionsList[idx]); // print the first element in the list
26     curl.request(options, function (firstError, data) { // do the curl request here
27       // console.log("here");
28       if (firstError) {
29         console.log("first error", firstError);
30       }
31     });
32   }
33 }

```

- optionslist: contains list of the audios and transcripts
 - inside Hui Xin's youtube-dl/downloads/audio
 - All the audios of the youtube videos extracted



```

4 // var app = express();
5 const fs = require('fs');
6 const path = require('path');
7 var joinPath = require('path.join'); //expected value for both UNIX and Windows: path/to/my/heart
8
9 // app.use(bodyParser.json());
10 // app.use(bodyParser.urlencoded({ extended: true }));
11 var obj = { // save this function inside a js object
12   getFileData: (optionsList, idx) => { // define function
13     if (idx == optionsList.length) { // if the idx == to the length of list
14       return // break loop
15     }
16     let options = { //store our request data in the following format
17       url: 'http://localhost:32769/transcriptions?async=false',
18       method: 'POST',
19       form: [
20         "audio=@" + optionsList[idx][1], // this gets the audio file path
21         "transcript=@" + optionsList[idx][0] // this gets the transcript file path
22       ],
23       //timeout: 5, // timeout stops the function
24     };
25     // console.log(optionsList[idx]); // print the first element in the list
26     curl.request(options, function (firstError, data) { // do the curl request here
27       // console.log("here");
28     });
29   }
30 }

```

getFileData(...):

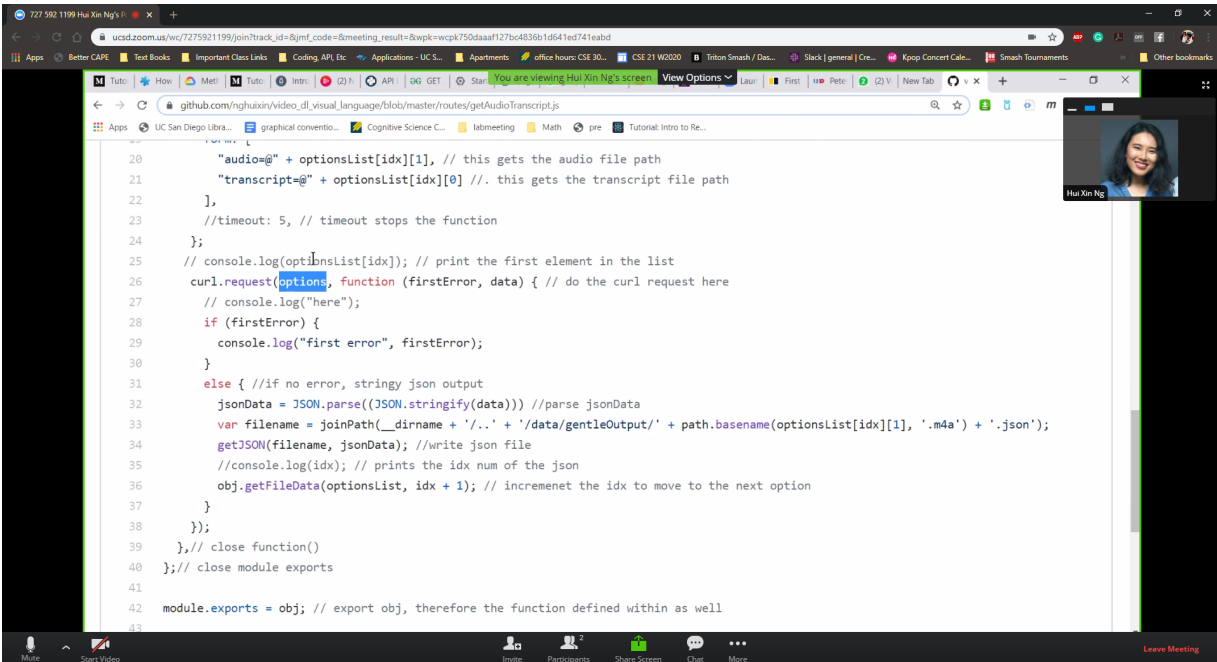
idx = argument Hui Xin created in order for her to move to the next element in the list of lists.

```

var obj = JSON object
// recursive function that calls obj.getFileData

var filename = joinPath...
//She defined the output as gentleOutput folder
//getJson saves the Gentle Jsons into an output file on her local computer

```



what is an asynchronous function?

https://www.google.com/search?q=what+is+an+asynchronous+function&rlz=1C1CHBF_enUS815US815&eq=what+is+an+asynchro&aqs=chrome.4.0j69i57j0l6.5637j1j1&sourceid=chrome&ie=UTF-8

first if statement is a catch

else statement toString() json output

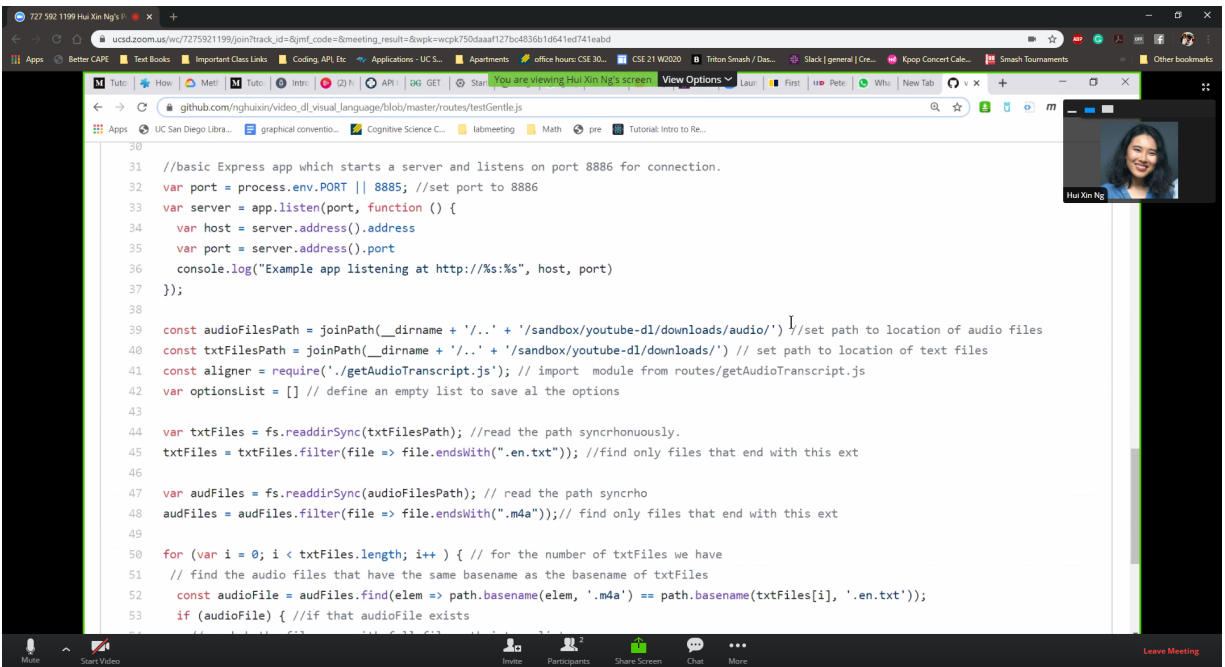
var filename indicates

```

getJSON(filename,data) {
  //stringify data and parse the data
  //uses fs to write the file
  //try catch for failure and success cases

```

testGentle.js:



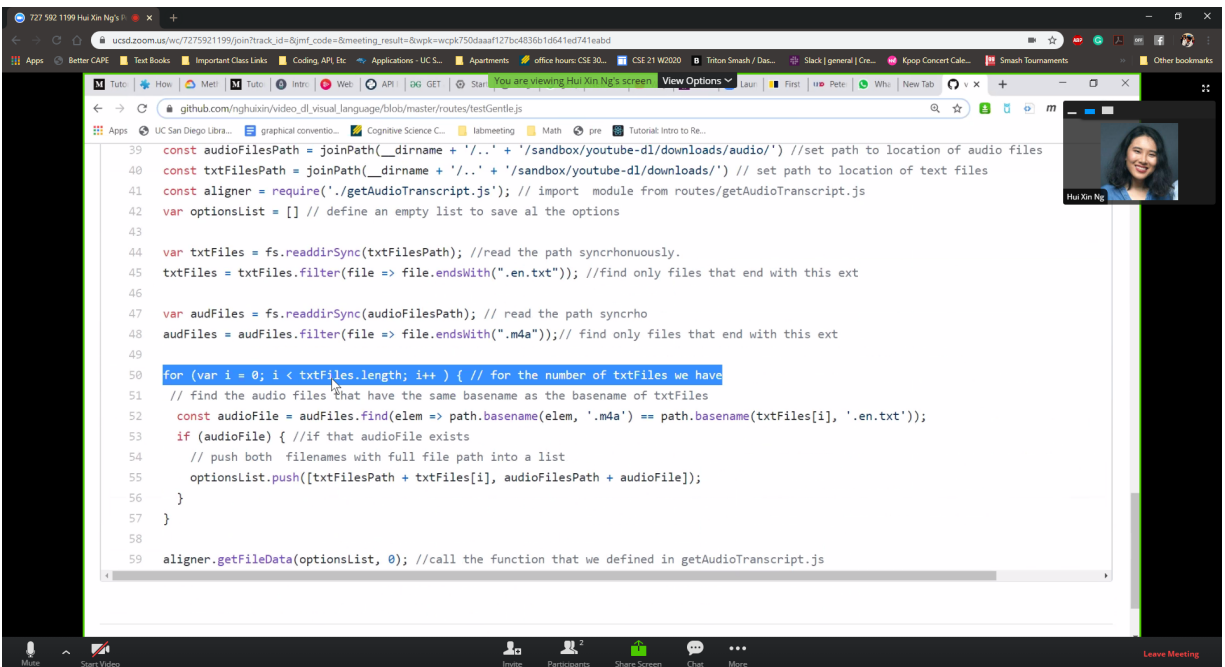
- How do I call the function getFileData from the other js?
- Define an aligner variable that requires that js file

// Read up on fs js documentation

```

txtFiles = txtFiles.filter(file => file.endsWith(".en.txt"));
// find files that end with this extension
// this is to filter out non-english transcript files

```



The for loop:

```

if(audiofilename = videoId):
  push both filenames into a list
  then insert this into a list of lists

```

path.basename drops the extensions so you can compare just the videoIds

Hui Xin:

Drop the extensions to do the name comparisons then reinsert the extensions to complete the curl request

```
34 var host = server.address().address
35 var port = server.address().port
36 console.log("Example app listening at http://%s:%s", host, port)
37 });
38
39 const audioFilePath = joinPath(__dirname + '../' + '/sandbox/youtube-dl/downloads/audio/') //set path to location of audio files
40 const txtFilePath = joinPath(__dirname + '../' + '/sandbox/youtube-dl/downloads/') // set path to location of text files
41 const aligner = require('./getAudioTranscript.js'); // import module from routes/getAudioTranscript.js
42 var optionsList = [] // define an empty list to save all the options
43
44 var txtFiles = fs.readdirSync(txtFilePath); //read the path synchronously.
45 txtFiles = txtFiles.filter(file => file.endsWith(".en.txt")); //find only files that end with this ext
46
47 var audFiles = fs.readdirSync(audioFilePath); // read the path synchrho
48 audFiles = audFiles.filter(file => file.endsWith(".m4a")); // find only files that end with this ext
49
50 for (var i = 0; i < txtFiles.length; i++) { // for the number of txtFiles we have
51 // find the audio files that have the same basename as the basename of txtFiles
52 const audioFile = audFiles.find(elem => path.basename(elem, '.m4a') == path.basename(txtFiles[i], '.en.txt'));
53 if (audioFile) { //if that audioFile exists
54 // push both filenames with full file path into a list
55 optionsList.push([txtFilePath + txtFiles[i], audioFilePath + audioFile]);
56 }
57 }
```

gentle_output_processing:

```
pandas/_libs/index_class_helper.pxi in pandas/_libs/index/Int64Engine._check_type()
KeyError: False

In [27]: sns.barplot(x=wordFreq[25:50].index, y=wordFreq[25:50].values)
wordFreq.set_xticklabels(wordFreq.get_xticklabels(), rotation=45)
# start, end = wordFreq.ymin()
# wordFreq.yaxis.set_ticks(np.arange(start, end, 100))

Out[27]:
```

	categoryName	channel	moments	transcript	videoid	numMoments
0	Education	UCsooayRKGN_zEEBklngZAJTED-Ed	[["word": "essential", "index": 3, "startTime": 10.94]]	Language is an essential part of our lives tha...	-GvVbmecJA	168.0
1	Education	UCsooayRKGN_zEEBklngZAJTED-Ed	[["word": "in", "index": 2, "startTime": 10.94]]	This just in: "Thief robs town with world's l...	-dX34V_XGcQ	74.0
2	News & Politics	UCLx67UDz6yWzQzQCufhAVox	[["word": "in", "index": 3, "startTime": 8.16]]	This is Casablanca, in Morocco, as the US War...	-W6zDQJeh8	367.0
3	News & Politics	UCLx67UDz6yWzQzQCufhAVox	[["word": "magazine", "index": 1, "startTime": 7.98]]	TIME magazine named "The Silence Breakers" Per...	-W6v2_0evJUJ	79.0
4	News & Politics	UCLx67UDz6yWzQzQCufhAVox	[["word": "in", "index": 0, "startTime": 0.45]]	In the last half of 2015, around 500 people in...	0u8CyrtbQ	146.0
5	Education	UCsooayRKGN_zEEBklngZAJTED-Ed	[["word": "in", "index": 0, "startTime": 7.98]]	In 2008, something incredible happened: a man...	0TpToqQT3E	158.0
6	Education	UCsooayRKGN_zEEBklngZAJTED-Ed	[["word": "job", "index": 2, "startTime": 8.84]]	There's a job out there with a great deal of p...	0knzGSV68v4	172.0

```
In [69]: ax = sns.boxplot(x="categoryName", y="numMoments", data=df)
ax.set_xticklabels(ax.get_xticklabels(), rotation=27)

Out[69]: [Text(0, 0, 'Education'),
Text(1, 0, 'News & Politics')]
```

numMoments: number of synset rating words that appear in that transcript