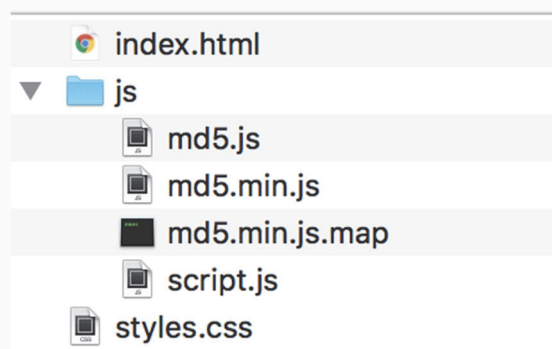


Putting together HTML/CSS with JavaScript + crash course on JSON and Marvel API

Sam Siu, Yilan He, Daniel Li
09/27/2018

What your folder will start as



- Your index.html file will include a .css file and other .js files
- Index.html will call functions from the .js files to use
- .js files can also use functions from those files

Example index.html

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" integrity="
sha384-BVYiiSIFeK1dGmJRAKycuHAHRg320mUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u" crossorigin="anonymous">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.0/jquery.min.js"></script>
  <script src="js/md5.min.js"></script>
  <script src="js/script.js"></script>
</head>
<body>

<h1 id="title">Marvel comics</h1>

<form id="search-form" >
  <label for="name">Title starts with:</label>
  <input type="text" id="name">
  <label for="startYear">Start year:</label>
  <input type="text" id="startYear">
  <button type="button" class="btn btn-primary" onclick="getMarvelResponse();">Submit</button>
</form>

<div id="result">
</div>

</body>
</html>
```

Example Page

Marvel comics

Title starts with: Start year:

Example script.js

```
// you will also have to setup the referring domains on your marvel developer portal
var PRIV_KEY = "YOUR_PRIVATE_KEY";
var PUBLIC_KEY = "YOUR_PUBLIC_KEY";

function getMarvelResponse() {
  // you need a new ts every request
  var ts = new Date().getTime();
  var hash = md5(ts+PRIV_KEY + PUBLIC_KEY).toString();

  var url = 'http://gateway.marvel.com:80/v1/public/comics';

  var titleStartsWith = document.getElementById('name').value; //js way
  var startYear = $('#startYear').val(); // jquery way

  // console.log(titleStartsWith, startYear); test
  $.getJSON(url, {
    ts: ts,
    apikey: PUBLIC_KEY,
    hash: hash,
    limit: 100,
    titleStartsWith: titleStartsWith,
    startYear: startYear
  })
  .done(function(data) {
    // sort of a long dump you will need to sort through
    console.log(data);
    for (var i = 0; i < data.data.count; i++) {
      $('#result').append('<p>' + data.data.results[i].title + '</p>') // print all the titles
    }
  })
  .fail(function(err){
    // the error codes are listed on the dev site
    console.log(err);
  });
};
```

Example Result

Marvel comics

Title starts with: Start year:

Avengers: The Crossing (1995) #1

Amazing X-Men (1995) #4

Astonishing X-Men (1995) #4

Amazing Spider-Man Super Special (1995) #1

Amazing X-Men (1995) #3

Astonishing X-Men (1995) #3

Amazing X-Men (1995) #2

Astonishing X-Men (1995) #2

Age of Apocalypse: The Chosen (1995) #1

Amazing X-Men (1995) #1

Astonishing X-Men (1995) #1

Navigating the Marvel API

- Go to <https://marvel.com/>
- Make an account
- Go to <https://developer.marvel.com/>
- Log in
- Click “Get an API key”
 - It should lead you to your account page with your public and private key infos
- Scroll down to “Your authorized referrers”
 - Add “localhost” to be a referrer
- Click Update at the bottom

JSON Objects

What is JSON?

- `{ "name": "John", "age": 30, "car": null }`
 - Surrounded by curly braces
 - Written in key:value pairs
 - Keys must be strings, and values must be a valid JSON data type (string, number, object, array, boolean or null).
 - Keys and values are separated by a colon.
 - Each key:value pair is separated by a comma.
 - Basically a Python dictionary

How to access

- Dot Notation
 - `myObj.name`
- Bracket Notation
 - `myObj["name"]`

Nested JSON Objects

```
myObj = {  
  "name": "John",  
  "age": 30,  
  "cars": {  
    "car1": "Ford",  
    "car2": "BMW",  
    "car3": "Fiat"  
  }  
}
```

`myObj.name.car1` or `myObj.name["car1"]`