

Lab 8

[Start Assignment](#)

Due Sunday by 11:59pm **Points** 100 **Submitting** a file upload **Available** Nov 5 at 12am - Nov 19 at 11:59pm

CS-546 Lab 8

Template Time

For this lab, you will be using HTML, CSS, and Handlebars to make your first simple templated web application! You will be building a form that allows you to search through characters in the Marvel API.

You will not need to use a database for this lab.

You **must** use the `async/await` keywords (not Promises). You will also be using `axios` (<https://github.com/axios/axios>), which is a HTTP client for Node.js; you can install it with `npm i axios`.

Marvel API

You will be using the [Marvel API \(https://developer.marvel.com\)](https://developer.marvel.com). You will need to register and sign up for an API key. You will not be able to make requests to the API without signing up and getting an [API key \(https://developer.marvel.com/account\)](https://developer.marvel.com/account). Please look at the data returned so you know the schema of the data and the objects it returns (the links to Characters above work but using my API key). **PLEASE MAKE SURE YOU READ THE DOCUMENTATION FOR THE API! IT IS SUPER IMPORTANT THAT YOU UNDERSTAND HOW AN API WORKS AND THE FORMAT OF THE DATA SO YOU CAN USE IT EFFECTIVELY!**

You can use the following code to construct the URL. You can read more about AUTHORIZING AND SIGNING REQUESTS from the link below

[https://developer.marvel.com/documentation/authorization \(https://developer.marvel.com/documentation/authorization\)](https://developer.marvel.com/documentation/authorization).

```
import md5 from 'blueimp-md5' //you will need to install this module;
const publicKey = 'your_public_key(API KEY) from Marvel dev portal';
const privateKey = 'your private key from Marvel dev portal';
const ts = new Date().getTime();
const stringToHash = ts + privateKey + publicKey;
const hash = md5(stringToHash);
const baseUrl = 'https://gateway.marvel.com:443/v1/public/characters';
const url = baseUrl + '?ts=' + ts + '&apikey=' + publicKey + '&hash=' + hash;
```

You will be using two endpoints of the Marvel API which is an API about Marvel for your Axios calls. The search character endpoint where you pass the search term as a query string parameter: https://gateway.marvel.com:443/v1/public/characters?nameStartsWith=SEARCH_TERM_HERE and then you'll get an individual character using the endpoint

<https://gateway.marvel.com:443/v1/public/characters/:id>

:id where :id is the ID of the character you are looking up.

You will use these two endpoints to make your axios.get calls depending on which route is called.

YOU MUST use the directory and file structure in the code stub or points will be deducted. You can download the starter template here: [Lab8_Stub.zip \(https://sit.instructure.com/courses/68229/files/12330975?wrap=1\)](https://sit.instructure.com/courses/68229/files/12330975?wrap=1) ↴ (https://sit.instructure.com/courses/68229/files/12330975?download_frd=1)
<https://sit.instructure.com/courses/64643/files/11372587?wrap=1>

PLEASE NOTE: THE STUB DOES NOT INCLUDE THE PACKAGE.JSON FILE. YOU WILL NEED TO CREATE IT! DO NOT ADD ANY OTHER FILE OR FOLDER APART FROM PACKAGE.JSON FILE. DO NOT FORGET THE START COMMAND OR THE "type": "module" property

You will be making three routes/pages in your application:

- <http://localhost:3000/> the main page of this application will provide a search form to start a search of characters for a keyword.
- <http://localhost:3000/searchmarvelcharacters> this page will make the axios call to the search endpoint and return up to 15 matching results that contain the provided request form param, `searchCharacterByName`
- <http://localhost:3000/marvelcharacter/{id}> this page will show all the details of the character with the id matching the provided URL param, `id`

All other URLs should return a 404

```
## `GET http://localhost:3000/`
```

This page will respond with a valid HTML document. The title of the document should be "Marvel Character Finder". You should have the title set as the `<title>` element of the HTML document and as an `h1` in your document.

Your page should reference a CSS file, </public/site.css>; this file should have *at least 10 rulesets* you must use at least 5 of them on this page.

You should have a `main` element, and inside of the `main` element have a `p` element with a brief (2-3 sentence description) of what your website does.

Also inside the `main` element, you will have a `form`; this `form` will `POST` to `/searchmarvelcharacters`. This `form` will have an `input` and a `label`; You will give the input and name AND id of "`searchCharacterByName`". The `label` should properly reference the same `id` as the `input`. You should also have a `input` with a `type of submit` that submits the form.

```
POST http://localhost:3000/searchmarvelcharacters
```

This route will read the `searchCharacterByName` parameter and then make an axios call to the Marvel API endpoint searching for that keyword. For example, if the user typed `spider` in the input field, you would make the axios call to: `https://gateway.marvel.com/v1/public/characters?nameStartsWith=spider&ts=TIME_STAMP_HERE&apikey=API_KEY_HERE&hash=HASH_HERE`

This route will respond with a valid HTML document with the results returned from the API. The title of the document should be "*Marvel Characters Found*". You should have the title set as the `<title>` element of the HTML document and as an `h1` in your document. In an `h2` element, you will print the supplied `searchCharacterByName`.

Your page should reference a CSS file, `/public/site.css`; this file should have *at least 10 rulesets* you must use at least 5 of them on this page.

You should have a `main` element, and inside of the `main` element have a `ol` tag that has a list of up to 15 characters matching the `searchCharacterByName` found in the request body in the following format (after searching `spider`). **DO NOT SHOW MORE THAN 15 Characters. (HINT: READ THE API DOCUMENTATION!)**

```
<ol>
  <li>
    <a href="/marvelcharacter/1010727">Spider-dok</a>
  </li>
  <li>
    <a href="/marvelcharacter/1009157">Spider-Girl (Anya Corazon)</a>
  </li>
  <li>
    <a href="/marvelcharacter/1009609">Spider-Girl (May Parker)</a>
  </li>
</ol>
```

You must also provide an `a` tag that links back to your `/` route with the text "`Search for Another Character`".

If no matches are found, you will print the following HTML paragraph:

```
<p class="not-found">We're sorry, but no results were found for {searchCharacterByName}.</p>
```

If the user does not input text into their form or enters just spaces into the input field, make sure to give a response status code of 400 on the page, and render an HTML page with a paragraph class called `error`; this paragraph should describe the error.

```
GET http://localhost:3000/marvelcharacter/{id}
```

This route will query the Marvel API using the the `id` parameter in the URL (for example: `https://gateway.marvel.com/v1/public/characters/1009609?ts=TS_HERE&apikey=API_KEY_HERE&hash=HASH_HERE`) and will respond with a valid HTML document with some of the character details. The title of the document should be the `name` of the character. You should have the title set as the `<title>` element of the HTML document.

Your page should reference a CSS file, `/public/site.css`; his file should have *at least 10 rulesets* you must use at least 5 of them on this page.

You should have a `main` element, and inside of the `main` element, you will have an `article` tag that has an `h1` with the character `name`, an `img` which the `src` is set to the value read from `thumbnail.path` in the data which is a URL to an image for the character. **NOTE: The path in the data is NOT the full image path. You will need to concatenate "/portrait_uncanny.jpg" to the end of the image path in the data (see HTML example below).** Please see the [documentation on images](https://developer.marvel.com/documentation/images) (`https://developer.marvel.com/documentation/images`). You will have a `p` element that contains the character `description`, a `h2` that's content says "Comics" and a `ul` for the list of comics. You only need to display the comic `name` as the list items: `li`

Matching Character Data Returned from API (We will not be using all the fields, just the ones noted above):

```
{
  "code": 200,
  "status": "Ok",
  "copyright": "© 2023 MARVEL",
  "attributionText": "Data provided by Marvel. © 2023 MARVEL",
  "attributionHTML": "<a href='http://marvel.com'>Data</a> provided by Marvel. © 2023 MARVEL",
  "etag": "bcc59558320b781e4fb550f7da167fbc342c0a5a",
  "data": {
    "offset": 0,
    "limit": 20,
    "total": 1,
    "count": 1,
    "results": [
      {
        "id": 1009609,
        "name": "Spider-Girl (May Parker)",
        "description": "May \"Mayday\" Parker is the daughter of Spider-Man and Mary Jane Watson-Parker. Born with all her father's powers-and the same silly sense of humor-she's grown up to become one of Earth's most trusted heroes and a fitting tribute to her proud papa.",
        "modified": "2016-03-02T11:04:46-0500",
        "thumbnail": {
          "path": "http://i.annihil.us/u/prod/marvel/i/mg/1/70/4c003adccbe4f",
          "extension": "jpg"
        },
        "resourceURI": "http://gateway.marvel.com/v1/public/characters/1009609",
        "comics": {
          "available": 200
        }
      }
    ]
  }
}
```

```

"collectionURI": "http://gateway.marvel.com/v1/public/characters/1009609/comics ↳_(http://gateway.marvel.com/v1/public/characters/1009609/comics)" ,
"cs" ,
"items": [
{
"resourceURI": "http://gateway.marvel.com/v1/public/comics/5286 ↳_(http://gateway.marvel.com/v1/public/comics/5286)" ,
"name": "Amazing Spider-Girl (2006)" ,
},
{
"resourceURI": "http://gateway.marvel.com/v1/public/comics/5281 ↳_(http://gateway.marvel.com/v1/public/comics/5281)" ,
"name": "Amazing Spider-Girl (2006) #1" ,
},
{
"resourceURI": "http://gateway.marvel.com/v1/public/comics/5591 ↳_(http://gateway.marvel.com/v1/public/comics/5591)" ,
"name": "Amazing Spider-Girl (2006) #2" ,
},
{
"resourceURI": "http://gateway.marvel.com/v1/public/comics/5701 ↳_(http://gateway.marvel.com/v1/public/comics/5701)" ,
"name": "Amazing Spider-Girl (2006) #3" ,
},
{
"resourceURI": "http://gateway.marvel.com/v1/public/comics/5843 ↳_(http://gateway.marvel.com/v1/public/comics/5843)" ,
"name": "Amazing Spider-Girl (2006) #4" ,
},
{
"resourceURI": "http://gateway.marvel.com/v1/public/comics/5997 ↳_(http://gateway.marvel.com/v1/public/comics/5997)" ,
"name": "Amazing Spider-Girl (2006) #5" ,
},
{
"resourceURI": "http://gateway.marvel.com/v1/public/comics/6130 ↳_(http://gateway.marvel.com/v1/public/comics/6130)" ,
"name": "Amazing Spider-Girl (2006) #6" ,
},
{
"resourceURI": "http://gateway.marvel.com/v1/public/comics/6270 ↳_(http://gateway.marvel.com/v1/public/comics/6270)" ,
"name": "Amazing Spider-Girl (2006) #7" ,
},
{
"resourceURI": "http://gateway.marvel.com/v1/public/comics/13478 ↳_(http://gateway.marvel.com/v1/public/comics/13478)" ,
"name": "Amazing Spider-Girl (2006) #8" ,
},
{
"resourceURI": "http://gateway.marvel.com/v1/public/comics/15856 ↳_(http://gateway.marvel.com/v1/public/comics/15856)" ,
"name": "Amazing Spider-Girl (2006) #9" ,
},
{
"resourceURI": "http://gateway.marvel.com/v1/public/comics/15966 ↳_(http://gateway.marvel.com/v1/public/comics/15966)" ,
"name": "Amazing Spider-Girl (2006) #10" ,
},
{
"resourceURI": "http://gateway.marvel.com/v1/public/comics/16151 ↳_(http://gateway.marvel.com/v1/public/comics/16151)" ,
"name": "Amazing Spider-Girl (2006) #11" ,
},
{
"resourceURI": "http://gateway.marvel.com/v1/public/comics/16522 ↳_(http://gateway.marvel.com/v1/public/comics/16522)" ,
"name": "Amazing Spider-Girl (2006) #12" ,
},
{
"resourceURI": "http://gateway.marvel.com/v1/public/comics/16523 ↳_(http://gateway.marvel.com/v1/public/comics/16523)" ,
"name": "Amazing Spider-Girl (2006) #13" ,
},
{
"resourceURI": "http://gateway.marvel.com/v1/public/comics/17249 ↳_(http://gateway.marvel.com/v1/public/comics/17249)" ,
"name": "Amazing Spider-Girl (2006) #14" ,
},
{
"resourceURI": "http://gateway.marvel.com/v1/public/comics/17385 ↳_(http://gateway.marvel.com/v1/public/comics/17385)" ,
"name": "Amazing Spider-Girl (2006) #15" ,
},
{
"resourceURI": "http://gateway.marvel.com/v1/public/comics/17629 ↳_(http://gateway.marvel.com/v1/public/comics/17629)" ,
"name": "Amazing Spider-Girl (2006) #16" ,
},
{
"resourceURI": "http://gateway.marvel.com/v1/public/comics/70668 ↳_(http://gateway.marvel.com/v1/public/comics/70668)" ,
"name": "Amazing Spider-Girl (2006) #17" ,
},
{
"resourceURI": "http://gateway.marvel.com/v1/public/comics/20868 ↳_(http://gateway.marvel.com/v1/public/comics/20868)" ,
"name": "Amazing Spider-Girl (2006) #18" ,
},
{
"resourceURI": "http://gateway.marvel.com/v1/public/comics/21003 ↳_(http://gateway.marvel.com/v1/public/comics/21003)" ,
"name": "Amazing Spider-Girl (2006) #19" ,
}
],
"returned": 20
},
"series": {
"available": 38,
"collectionURI": "http://gateway.marvel.com/v1/public/characters/1009609/series ↳_(http://gateway.marvel.com/v1/public/characters/1009609/series)" ,
"es" ,
"items": [
{
"resourceURI": "http://gateway.marvel.com/v1/public/series/1126 ↳_(http://gateway.marvel.com/v1/public/series/1126)" ,
"name": "Amazing Spider-Girl (2006 - 2009)" ,
},
{
"resourceURI": "http://gateway.marvel.com/v1/public/series/2234 ↳_(http://gateway.marvel.com/v1/public/series/2234)" ,
"name": "AMAZING SPIDER-GIRL VOL. 1: WHATEVER HAPPENED TO THE DAUGHTER OF SPIDER-MAN TPB (2007)" ,
},
{
"resourceURI": "http://gateway.marvel.com/v1/public/series/3101 ↳_(http://gateway.marvel.com/v1/public/series/3101)" ,
"name": "AMAZING SPIDER-GIRL VOL. 2: COMES THE CARNAGE! TPB (2007)" ,
},
{
"resourceURI": "http://gateway.marvel.com/v1/public/series/5376 ↳_(http://gateway.marvel.com/v1/public/series/5376)" ,
"name": "Amazing Spider-Man Family (2008 - 2009)" ,
},
{
"resourceURI": "http://gateway.marvel.com/v1/public/series/1143 ↳_(http://gateway.marvel.com/v1/public/series/1143)" ,
}
]
}

```

```

    "name": "Avengers Next (2006 - 2007)"
},
{
  "resourceURI": "http://gateway.marvel.com/v1/public/series/1995 ↳ (http://gateway.marvel.com/v1/public/series/1995)",
  "name": "Cable (1993 - 2002)"
},
{
  "resourceURI": "http://gateway.marvel.com/v1/public/series/5260 ↳ (http://gateway.marvel.com/v1/public/series/5260)",
  "name": "Counter X Vol. 1 (2008)"
},
{
  "resourceURI": "http://gateway.marvel.com/v1/public/series/19 ↳ (http://gateway.marvel.com/v1/public/series/19)",
  "name": "DAREDEVIL/ECHO: PARTS OF A HOLE PREMIERE HC (2010 - 1999)"
},
{
  "resourceURI": "http://gateway.marvel.com/v1/public/series/2119 ↳ (http://gateway.marvel.com/v1/public/series/2119)",
  "name": "Fantastic Five (1999 - 2000)"
},
{
  "resourceURI": "http://gateway.marvel.com/v1/public/series/3281 ↳ (http://gateway.marvel.com/v1/public/series/3281)",
  "name": "Fantastic Five: The Final Doom (2007)"
},
{
  "resourceURI": "http://gateway.marvel.com/v1/public/series/13881 ↳ (http://gateway.marvel.com/v1/public/series/13881)",
  "name": "Fear Itself: The Home Front (2010)"
},
{
  "resourceURI": "http://gateway.marvel.com/v1/public/series/2572 ↳ (http://gateway.marvel.com/v1/public/series/2572)",
  "name": "Iron Man (1998 - 2004)"
},
{
  "resourceURI": "http://gateway.marvel.com/v1/public/series/2031 ↳ (http://gateway.marvel.com/v1/public/series/2031)",
  "name": "J2 (1998 - 1999)"
},
{
  "resourceURI": "http://gateway.marvel.com/v1/public/series/923 ↳ (http://gateway.marvel.com/v1/public/series/923)",
  "name": "Last Hero Standing (2005)"
},
{
  "resourceURI": "http://gateway.marvel.com/v1/public/series/991 ↳ (http://gateway.marvel.com/v1/public/series/991)",
  "name": "Last Planet Standing (2006)"
},
{
  "resourceURI": "http://gateway.marvel.com/v1/public/series/322 ↳ (http://gateway.marvel.com/v1/public/series/322)",
  "name": "MARVEL AGE: SPIDER-GIRL VOL. 1: LEGACY DIGEST (2004)"
},
{
  "resourceURI": "http://gateway.marvel.com/v1/public/series/26878 ↳ (http://gateway.marvel.com/v1/public/series/26878)",
  "name": "Marvel's Greatest Creators: What If? - Spider-Girl (2019)"
},
{
  "resourceURI": "http://gateway.marvel.com/v1/public/series/2059 ↳ (http://gateway.marvel.com/v1/public/series/2059)",
  "name": "Paradise X (2002 - 2003)"
},
{
  "resourceURI": "http://gateway.marvel.com/v1/public/series/2431 ↳ (http://gateway.marvel.com/v1/public/series/2431)",
  "name": "Paradise X Vol. 1 (2007)"
},
{
  "resourceURI": "http://gateway.marvel.com/v1/public/series/2702 ↳ (http://gateway.marvel.com/v1/public/series/2702)",
  "name": "PARADISE X VOL. 2 TPB [NEW PRINTING] (2007)"
}
],
"returned": 20
},
"stories": {
  "available": 313,
  "collectionURI": "http://gateway.marvel.com/v1/public/characters/1009609/stories ↳ (http://gateway.marvel.com/v1/public/characters/1009609/stories)",
  "items": [
    {
      "resourceURI": "http://gateway.marvel.com/v1/public/stories/791 ↳ (http://gateway.marvel.com/v1/public/stories/791)",
      "name": "Cover #791",
      "type": "cover"
    },
    {
      "resourceURI": "http://gateway.marvel.com/v1/public/stories/792 ↳ (http://gateway.marvel.com/v1/public/stories/792)",
      "name": "Interior #792",
      "type": "interiorStory"
    },
    {
      "resourceURI": "http://gateway.marvel.com/v1/public/stories/806 ↳ (http://gateway.marvel.com/v1/public/stories/806)",
      "name": "Cover #806",
      "type": "cover"
    },
    {
      "resourceURI": "http://gateway.marvel.com/v1/public/stories/807 ↳ (http://gateway.marvel.com/v1/public/stories/807)",
      "name": "Interior #807",
      "type": "interiorStory"
    },
    {
      "resourceURI": "http://gateway.marvel.com/v1/public/stories/808 ↳ (http://gateway.marvel.com/v1/public/stories/808)",
      "name": "Cover #808",
      "type": "cover"
    },
    {
      "resourceURI": "http://gateway.marvel.com/v1/public/stories/809 ↳ (http://gateway.marvel.com/v1/public/stories/809)",
      "name": "Interior #809",
      "type": "interiorStory"
    },
    {
      "resourceURI": "http://gateway.marvel.com/v1/public/stories/810 ↳ (http://gateway.marvel.com/v1/public/stories/810)",
      "name": "Cover #810",
      "type": "cover"
    },
    {
      "resourceURI": "http://gateway.marvel.com/v1/public/stories/811 ↳ (http://gateway.marvel.com/v1/public/stories/811)",
      "name": "Interior #811",
      "type": "interiorStory"
    }
  ]
}

```

```

    },
    {
      "resourceURI": "http://gateway.marvel.com/v1/public/stories/812 ▶_(http://gateway.marvel.com/v1/public/stories/812)_",
      "name": "Cover #812",
      "type": "cover"
    },
    {
      "resourceURI": "http://gateway.marvel.com/v1/public/stories/813 ▶_(http://gateway.marvel.com/v1/public/stories/813)_",
      "name": "Interior #813",
      "type": "interiorStory"
    },
    {
      "resourceURI": "http://gateway.marvel.com/v1/public/stories/814 ▶_(http://gateway.marvel.com/v1/public/stories/814)_",
      "name": "Cover #814",
      "type": "cover"
    },
    {
      "resourceURI": "http://gateway.marvel.com/v1/public/stories/815 ▶_(http://gateway.marvel.com/v1/public/stories/815)_",
      "name": "Interior #815",
      "type": "interiorStory"
    },
    {
      "resourceURI": "http://gateway.marvel.com/v1/public/stories/816 ▶_(http://gateway.marvel.com/v1/public/stories/816)_",
      "name": "Cover #816",
      "type": "cover"
    },
    {
      "resourceURI": "http://gateway.marvel.com/v1/public/stories/817 ▶_(http://gateway.marvel.com/v1/public/stories/817)_",
      "name": "Interior #817",
      "type": "interiorStory"
    },
    {
      "resourceURI": "http://gateway.marvel.com/v1/public/stories/818 ▶_(http://gateway.marvel.com/v1/public/stories/818)_",
      "name": "Cover #818",
      "type": "cover"
    },
    {
      "resourceURI": "http://gateway.marvel.com/v1/public/stories/819 ▶_(http://gateway.marvel.com/v1/public/stories/819)_",
      "name": "Interior #819",
      "type": "interiorStory"
    },
    {
      "resourceURI": "http://gateway.marvel.com/v1/public/stories/820 ▶_(http://gateway.marvel.com/v1/public/stories/820)_",
      "name": ""THE PEOPLE PLAYED BY GAMES!" Torn between her loyalties to Kaine and the Black Tarantula, Spider-Girl finally confronts La
dy Oc",
      "type": "cover"
    },
    {
      "resourceURI": "http://gateway.marvel.com/v1/public/stories/821 ▶_(http://gateway.marvel.com/v1/public/stories/821)_",
      "name": ""THE PEOPLE PLAYED BY GAMES!" Torn between her loyalties to Kaine and the Black Tarantula, Spider-Girl finally confronts La
dy Oc",
      "type": "interiorStory"
    },
    {
      "resourceURI": "http://gateway.marvel.com/v1/public/stories/822 ▶_(http://gateway.marvel.com/v1/public/stories/822)_",
      "name": ""IF THIS BE MY DESTINY!" Spider-Girl learns the fate Norman Osborn planned for his grandson when she finally uncovers the
secre",
      "type": "cover"
    },
    {
      "resourceURI": "http://gateway.marvel.com/v1/public/stories/823 ▶_(http://gateway.marvel.com/v1/public/stories/823)_",
      "name": ""IF THIS BE MY DESTINY!" Spider-Girl learns the fate Norman Osborn planned for his grandson when she finally uncovers the
secre",
      "type": "interiorStory"
    }
  ],
  "returned": 20
},
"events": {
  "available": 1,
  "collectionURI": "http://gateway.marvel.com/v1/public/characters/1009609/events ▶_(http://gateway.marvel.com/v1/public/characters/1009609/even
ts)_",
  "items": [
    {
      "resourceURI": "http://gateway.marvel.com/v1/public/events/302 ▶_(http://gateway.marvel.com/v1/public/events/302)_",
      "name": "Fear Itself"
    }
  ],
  "returned": 1
},
"urls": [
  {
    "type": "detail",
    "url": "http://marvel.com/characters/2171/spider-girl?utm_campaign=apiRef&utm_source=a8f9ccf932bf29fd379ef00e11668673 ▶_(http://marvel.c
om/characters/2171/spider-girl?utm_campaign=apiRef&utm_source=a8f9ccf932bf29fd379ef00e11668673)"
  },
  {
    "type": "wiki",
    "url": "http://marvel.com/universe/Spider-Girl_(MC2)?utm_campaign=apiRef&utm_source=a8f9ccf932bf29fd379ef00e11668673 ▶_(http://marvel.co
m/universe/Spider-Girl_(MC2)?utm_campaign=apiRef&utm_source=a8f9ccf932bf29fd379ef00e11668673)"
  },
  {
    "type": "comiclink",
    "url": "http://marvel.com/comics/characters/1009609/spider-girl_may_parker?utm_campaign=apiRef&utm_source=a8f9ccf932bf29fd379ef00e1166
8673 ▶_(http://marvel.com/comics/characters/1009609/spider-girl_may_parker?utm_campaign=apiRef&utm_source=a8f9ccf932bf29fd379ef00e1166
8673)"
  }
]
}
}
}

```

HTML Format Printed for the character. This will go into your `main` element:

```

<article>
  <h1>Spider-Girl (May Parker)</h1>
  May "Mayday" Parker is the daughter of Spider-Man and Mary Jane Watson-Parker. Born with all her father's powers-and the same silly sense of humor-she's grown up to become one of Earth's most trusted heroes and a fitting tribute to her proud papa.</a></td>
<td>Comics</td>
<ul>
<li>Amazing Spider-Girl (2006) #1</li>
<li>Amazing Spider-Girl (2006) #2</li>
<li>Amazing Spider-Girl (2006) #3</li>
<li>Amazing Spider-Girl (2006) #4</li>
<li>Amazing Spider-Girl (2006) #5</li>
<li>Amazing Spider-Girl (2006) #6</li>
<li>Amazing Spider-Girl (2006) #7</li>
<li>Amazing Spider-Girl (2006) #8</li>
<li>Amazing Spider-Girl (2006) #9</li>
<li>Amazing Spider-Girl (2006) #10</li>
<li>Amazing Spider-Girl (2006) #11</li>
<li>Amazing Spider-Girl (2006) #12</li>
<li>Amazing Spider-Girl (2006) #13</li>
<li>Amazing Spider-Girl (2006) #14</li>
<li>Amazing Spider-Girl (2006) #15</li>
<li>Amazing Spider-Girl (2006) #16</li>
<li>Amazing Spider-Girl (2006) #17</li>
<li>Amazing Spider-Girl (2006) #18</li>
<li>Amazing Spider-Girl (2006) #19</li>
</ul>
</article>

```

You must also provide an `a` tag that links back to your `/` route with the text "[Search for Another Character](#)".

If there is no character found for the given ID, make sure to give a response status code of 404 on the page, and render an HTML page with a paragraph class called `error`; this paragraph should describe the error. This route MUST work for every valid character ID in the API, NOT in just the list of characters as that url only returns the first page of characters!

<http://localhost:3000/public/site.css>

This file should have 10 total rulesets that apply to your various pages. Each page must use at least 5 rulesets of the 10. You can include more than 10 if you so desire.

References and Packages

Basic CSS info can easily be referenced in the [MDN CSS tutorial](https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Getting_started) (https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Getting_started).

Hints

You can use variables in your handlebars layout, that you pass to `res.render`. For example, in your layout you could have:

```
<meta name="keywords" content="{{keywords}} />
```

And in your route:

```
res.render("someView", {keywords: "dogs coffee keto"});
```

Which will render as:

```
<meta name="keywords" content="dogs coffee keto" />
```

Or, perhaps, the title tag.

Requirements

1. You **must not submit** your `node_modules` folder
2. You **must remember** to save your dependencies to your `package.json` folder
3. You must do basic error checking in each function
4. Check for arguments existing and of proper type.
5. Throw if anything is out of bounds (ie, trying to perform an incalculable math operation or accessing data that does not exist)
6. You **MUST** use `async/await` for all asynchronous operations.
7. You **must remember** to update your `package.json` file to set `app.js` as your starting script!
8. **Your HTML must be valid** (https://validator.w3.org/#validate_by_input) or you will lose points on the assignment.
9. Your HTML must make semantical sense; usage of tags for the purpose of simply changing the style of elements (such as `i`, `b`, `font`, `center`, etc) will result in points being deducted; think in terms of content first, then style with your CSS.
10. You can be as creative as you'd like to fulfill front-end requirements; if an implementation is not explicitly stated, however you go about it is fine (provided the HTML is valid and semantical). Design is not a factor in this course.
11. All inputs must be properly labeled!
12. All previous requirements about the `package.json` author, start task, dependenices, etc. still apply