Here’s a good way to get started when using OMDB API :

***Going into the OMDB website:***

1. Read through the documentation on <http://www.omdbapi.com/>
2. Determine what URL you are needing in order to render the data you would need to extract from OMDB’s database. (we found it to be: <http://www.omdbapi.com/?s=something> (according to the documentation, this allows us to search their datatabase using the title of the movie.)
3. So, from the above URL, we did a search for the movies starting with “terminator” <http://www.omdbapi.com/s=terminator> was successful URL, that rendered our data results in JSON.
4. Based on the JSON that came back, you want to determine what information you want to extract and place on your HTML page. (we want to extract Title and Year)

{

**Title**: "Terminator 2: Judgment Day",

**Year**: "1991",

**imdbID**: "tt0103064",

**Type**: "movie",

**Poster**: "<http://ia.media-imdb.com/images/M/MV5BMTI4MDAwMDY3N15BMl5BanBnXkFtZTcwODIwMzMzMQ@@._V1._CR46,1,342,473_SY132_CR3,0,89,132_AL_.jpg_V1_SX300.jpg>"

}

***Form Submission and Event Handling:***

1.) From my webpage that I created, I know I have a form tags and input fields.

<form id="movie-search-form">

<input type="text" id="query" placeholder="Movie title">

<button type="submit">Search</button>

</form>

2.) This tells me that I want to “grab” the <form> tag and assign it to a variable.

3.) Add an event listener to the submission of this form, and run the necessary function once the form is submitted. (formSubmitted function)

***Once the form is submitted:***

1. Create a function called: formSubmitted, takes the event as a parameter
2. Prevnt the default action of the form submission
3. Grab the input.value of what was entered into the input field as a search criteria.
4. Add that to the URL to complete the API call
5. Use $.get to extract data from that URL. So, upon success, run a function that would drill down on the results of the JSON object. (resultsReceived function)

***Once the results are received (JSON):***

1. Create a function called resultsReceived, takes the JSON results as a parameter
2. Drill down on those results to get to the array.
3. Loop through each item in the array (using forEach) and run a function that would place display each movie on the page. (displayMovie function)

***Displaying individual movies onto the DOM:***

1. Creating a **function** “**displayMovie**” - that takes in one movie as my parameter
2. Create all the new elements and add appropriate attributes to these newly create elements. (which new elements would I need? See below for the setup.
3. Append these new elements to the parent on the html page.

***Set up to displaying one movie on the webpage.***

1. A webpage was created to show how I want my data to appear in the final results. This set up allowed me to think through which ELEMENTS need to be created. (see below.) I used hardcoded data so that I could see how I wanted my HTML page to look.
2. Creating the CSS classes so that the HTML page renders what the styles that I want to see also helps me “see” additional attributes that I would need to add onto those newly created ELEMENTS.

<main>

<form id="movie-search-form">

<input type="text" id="query" placeholder="Movie title">

<button type="submit">Search</button>

</form>

<ul id="movies">

<li class="movie">

<div class="movie-title">

<a href="http://www.imdb.com/title/tt0106611" target="\_blank">

Cool Runnings

</a>

</div>

<div class="movie-release-date">1993</div>

</li>

</ul>

</main>

1. From here, I can tell that when my data is extracted, I would need to create:

**li, div, a, div** tags to accommodate what I want to see as an end result.

4.) I can also tell that I would need to **class, href, and target** attributes if I want my final rendered results to mirror my hardcoded webpage.

1. **So my pseudo code would be:**
   1. Creating a **function** “**displayMovie**” - that takes in one movie as my parameter
   2. Create all the new elements and add appropriate attributes to these newly create elements.
   3. Notice the <a> has two attributes:

<a href="http://www.imdb.com/title/tt0106611" target="\_blank">

d. The href=”https://…..” used in my <a> had to be concatenated with:

var url = "http://www.imdb.com/title/" + movie["imdbID"]; (movie[“imdbID]” came from the JSON object returned from the API call.)

* 1. Append these new elements to the parent(s) on the html page.