

Web, Mobile and Security

Lab: Asynchronous Content

0. Preparing the work environment

- Launch your virtual machine, ensuring that all parameters are setup correctly (cfr. to the previous lab).
- Create a repo per exercise in your own group at <https://op-gitlab.howest.be/TI/2019-2020/s2-web-mobile-and-security/students>.
- Note: only in case you get an error message navigating to the link above, make sure you first visit <https://op-gitlab.howest.be/TI/2019-2020/s2-web-mobile-and-security/students/student-firstname-lastname> (replacing **firstname-lastname** by your own login name) and try creating a repo then. If this does not work, contact one of the lecturers.
- Make sure to commit and push your code on a regular basis.

1. Exercise 1: Open Movie Database

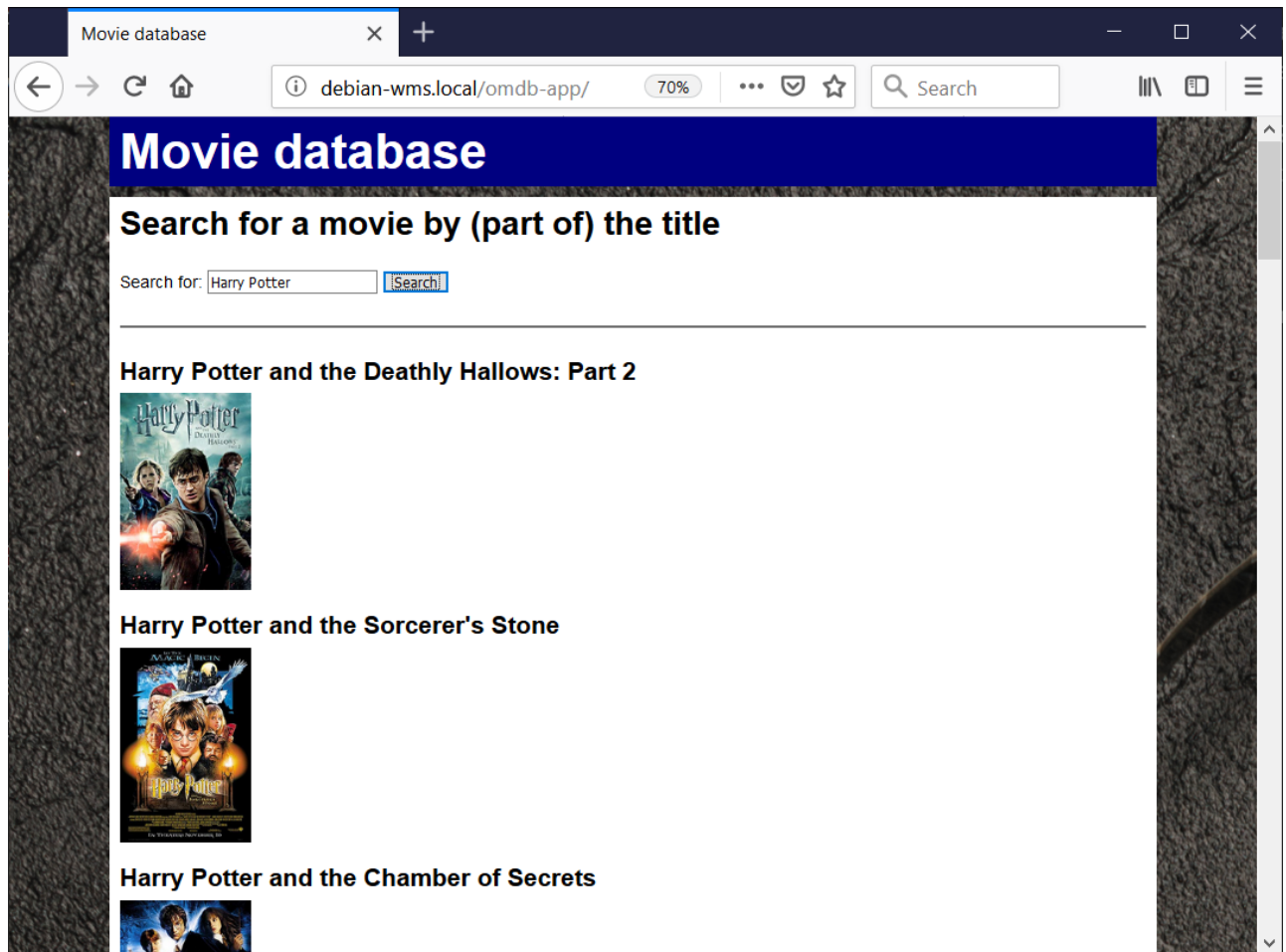
- The Open Movie Database API allows you to retrieve information about movies, comparable to the IMDB.
- To use the API, you need a personal and private API key. Request a new key via this address: <http://www.omdbapi.com/apikey.aspx>

!! Make sure to choose the option **FREE**, which does not cost you any money, but is limited to 1000 requests per day.

- Using PuTTY, connect to your VM as **user**.
- Execute the following command to create a new project:

```
./build-plain-project omdb-app
```

- Try surfing to your new app by navigating to the URL <http://debian-wms.local/omdb-app/>.
- Next, using the **Fetch API**, build a web application that prompts the user for some text and will then list all matching movies.
- Don't focus too much on the layout/CSS of your web page: the main focus is the HTML and JavaScript code, required to obtain the desired functionality.
- Below is a sample screenshot of what the application might look like after the user has filled in the search string *Harry Potter* and clicked **Search**:



- Some tips during development:
 - Study the API documentation thoroughly to determine the correct URL to use.
 - Log the JSON output of the API calls to the console during development, to better understand the response transmitted by the API.

2. Exercise 2: Open Weather Map

- The Open Weather Map API allows you to retrieve information about the weather.
- To use the API, you need a personal and private API key. Request a new key via this address: <https://openweathermap.org/appid>.

!! Make sure to choose the option **FREE**, which does not cost you any money, but is limited to 60 requests per minute.

- Using PuTTY, connect to your VM as **user**.
- Execute the following command to create a new project:

```
./build-plain-project weather-app
```

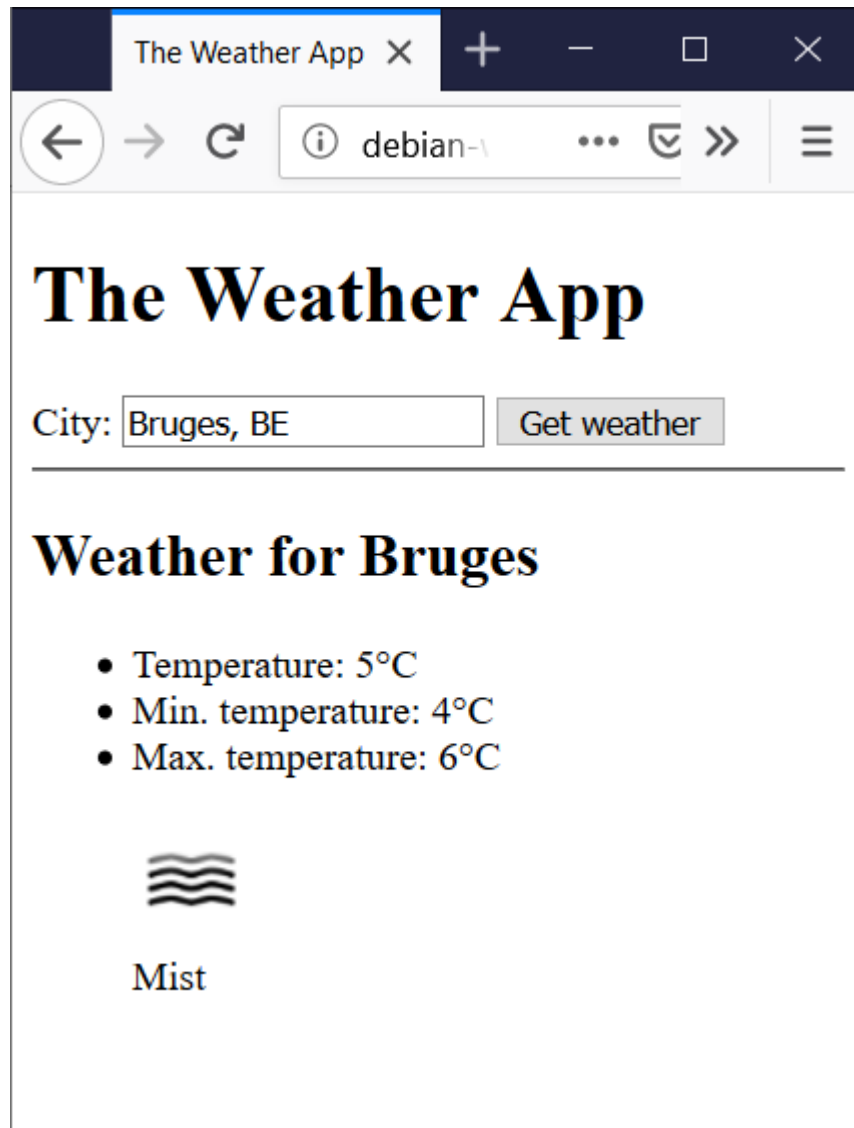
- Try surfing to your new app by navigating to the URL <http://debian-wms.local/weather-app/>.

- Next, using **JSON-P**, build a web application that prompts the user for a location and will then provide weather information in **degrees Celcius** instead of Kelvin or Fahrenheit.
- To work with JSON-P, you need to specify a callback function.

The base URL pattern is:

`https://api.openweathermap.org/data/2.5/weather?q=CITY,COUNTRY&appid=YOUR_API_KEY&units=metric&callback=CALLBACK_FUNCTION_NAME`

- Replace **CITY** by your city (e.g. Bruges).
 - Replace **COUNTRY** by the two character country code (e.g. BE for Belgium).
 - Replace **YOUR_API_KEY** by the API key you received in the beginning of this exercise.
 - Replace **CALLBACK_FUNCTION_NAME** by the name of the callback function in your JavaScript code (cfr. JSON-P theory slides).
- Don't focus too much on the layout/CSS of your web page: the main focus is the HTML and JavaScript code, required to obtain the desired functionality.
 - Below is a sample screenshot of what the application might look like after the user has filled in the location string *Bruges, be* and clicked **Search**:



- Tips:
 - Log any JSON output to the console during development, to better understand the response transmitted by the API.
 - For the weather icon, take a look at <https://openweathermap.org/weather-conditions>.