



## Assignment 7 (NF, minor subject)

---

*Due: Friday 16.12.2016; 14:00h (1 Week)*

### Goals

After doing the exercises,

- You know what API keys are used for
- You can perform Ajax requests using jQuery

### Task 1: API Keys

**Difficulty: Easy**

There are a couple of services that offer public Web APIs. Spotify does not require us to sign up to use the API, but many other services only allow using their API if you sign up and get an API key (for example: flickr).

Answer the following questions:

- a) Why are API keys necessary?
- b) Which benefits do they provide?
- c) Which problems do they generate?

Write down the answers in a file “task01.txt” and include it in your submission. Do not use a Text Editor that uses Rich Text (e.g. do **not** use TextEdit).

### Task 2: Lazy Loading

**Difficulty: Easy**

Research online what “lazy loading” means. What does it have to do with AJAX? When is it necessary? Write 300 words maximum.

Write down your answer in a file “task02.txt” and include it in your submission.



### Task 3: AJAX Movie Search with an API

Difficulty: Medium

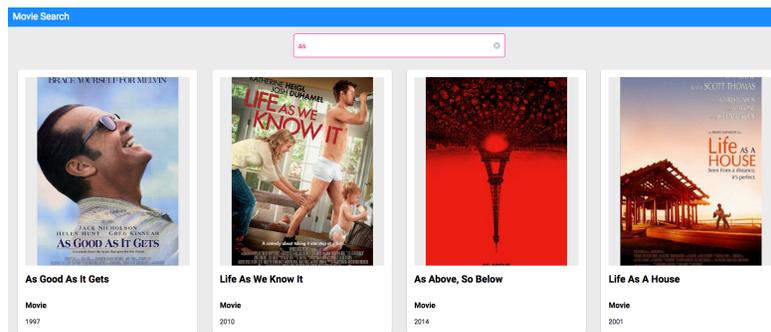
The webpage <http://www.omdbapi.com/> lets you search for movies and TV shows with their API.

The API works simply by fetching information over their URL, for example:

<http://www.omdbapi.com/?s=ghostbusters>

As you can see, the GET query parameter-key is “s” (the value is “ghostbusters” in this case)

Create a web page that allows a user to search for movies. The user enters keywords (for example the movie title) into an <input> field and the webpage automatically fetches the results for that keyword from the API. We use jQuery for this task.



Watch a demo here: <https://youtu.be/3wQ2wjn6dXE>

There is a code skeleton on the web-site, that already includes most of the required functionality. Only two TODOs are left in the code:

1. Inside the search(title) function:
  - a. The Ajax request should be fired here.
  - b. Use the correct jQuery method to create a GET request
  - c. Pass the correct parameters to this method, as shown in the tutorial.
2. Inside the handleResponse(data) function:
  - a. The passed parameter “data” is the response from the server.
  - b. The list of movies is accessible with data.Search
  - c. Loop through the array in data.Search. It contains Objects like this:

```
{ "Title": "Ghostbusters",  
  "Year": "1984",  
  "imdbID": "tt0087332",  
  "Type": "movie",  
  "Poster": "<poster_image_url>" }
```
  - d. Use the generateMovieDiv function to quickly generate an element that contains all necessary things.
  - e. Append the generated element to the resultDiv.

OPTIONAL: Try to do this without the code skeleton.

Include the file movie-search.html in your submission.



## Submission

Please turn in your solution via UniWorX as a ZIP archive. You can form groups of up to four people.

We encourage you to sign up for our Slack Team! All you need is a CIP account and an email address that ends in “@cip.ifi.lmu.de”. Ask us if you don’t know how to get them.

If you have questions or comments before the submission, please contact one of the tutors. They are on Slack: [@tobi.seitz](#), [@peterjuras](#) and [@thomas-weber](#). Remember, that they also want to enjoy their weekends 😊

It also makes a lot of sense to ask questions directly in our [#mmn-ws1617](#) channel. Maybe fellow students can offer help or benefit from the answers, too!

## CodeLabs

As always, everyone is invited to participate in the CodeLabs on Wednesday between 6 and 8 pm. We encourage you to go there if you struggle with installation or nasty bugs that just don’t seem to go away. The times are flexible, so you can go there sometime during the CodeLab.