



Assignment 4

Due: Thu 21.11.2019; 18:00h (1 Week)

Goals

After doing these exercises, you know how to use Javascript to...

- Create AJAX requests
- Work with custom elements

Task 1: MMN Assignment Statistics

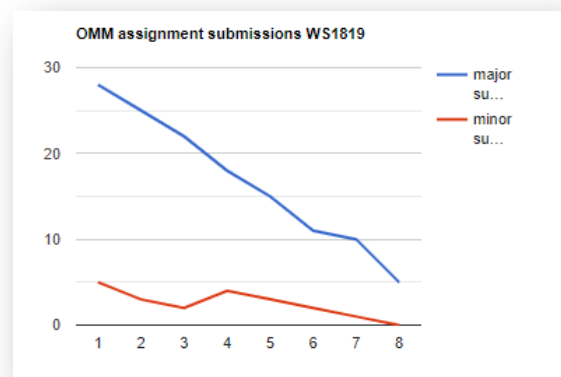
Difficulty: Easy

In this assignment's skeleton folder you will find two files:

- *google-chart-demo.html*: A webpage that demonstrates how the web component google-chart can be used to create charts.
- *x-ajax-demo.html*: Web components cannot only be used for UI. One can also encapsulate background functionality into a custom element, e.g. Ajax requests. This html file implements and demonstrates a web component that performs an Ajax request and prints the response onto the webpage.

Now we want to combine both, to visualize OMM assignment submission statistics that are fetched from the Uni2Work API.

- Make yourself familiar with the google-chart component and the x-ajax component.
- Integrate the google-chart into the x-ajax demo.
- Make the x-ajax demo load assignment statistics of the winter semester 2018/19 from the Uni2Work API, and plot the responded data as a line chart.



(To be honest, there is no Uni2Work API. We thus created a "fake API" by uploading the data to GitHub. So the URL of the API is:

[https://github.com/mimuc/omm-ws1920/blob/master/assignments/04-](https://github.com/mimuc/omm-ws1920/blob/master/assignments/04-webcomponents/assignment-statistics-api/ws1819.json)

[webcomponents/assignment-statistics-api/ws1819.json](https://github.com/mimuc/omm-ws1920/blob/master/assignments/04-webcomponents/assignment-statistics-api/ws1819.json) (Please do not simply download the json file into the project folder and import it locally 😊)



Task 2: Compare with previous terms

Difficulty: Hard

Our fake Uni2Work API can also show data from the previous and the current term, by replacing *ws1819* with *ws1617*, *ws1718* or *ws1920*. Thus it would be cool if the user could select one of the 4 terms via a dropdown (of course still using the *x-ajax* and *google-chart* components).

A possible procedure could be:

- Add a dropdown to the webpage (you could also look for a beautiful web component on <https://www.webcomponents.org/>)
- When the user selects an option in the dropdown, the new url should be constructed and the url attribute of the x-ajax component should be updated
- Currently the x-ajax component triggers only one time (on load), with the url that was set initially. You have to enhance its implementation in order to be reacting automatically on a change of the url attribute. Implement the `attributeChangedCallback` of x-ajax. You can take a look into this example web component to see how that can work: <https://github.com/webcomponents/hello-world-element>

Submission

Please turn in your solution as ZIP file via Uni2Work. You can form groups of up to three people.

We encourage you to sign up for Slack! 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 [@Aleksa](#) and [@Andre](#), remember that they also want to enjoy their weekends 😊

It also makes sense to ask the question in our #omm-ws1920 channel. Maybe fellow students can help or benefit from the answers, too!