Assignment 10 (HF, major subject)

Due: Wed 13.01.2015; 12:00h (3 weeks)

Goals
After doing these exercises,

- You know how to combine multiple data sources to a mash-up

Task 1: Mash-Ups

Many services on the web can be considered “mash-ups”. Every site that uses content from more than one remote service and displays that content on a single page is basically a mashup.

Find three sites that you think are mashups and briefly describe
a) the content sources
b) the topic of the site
c) how many XMLHttpRequests are made when the site loads for the first time.
Put all queries / statements in a .txt file and put it into the folder ‘task1’.

Task 2a: Integrate 3rd party service into our notes app. Difficulty: Intermediate

For assignment 09, we produced a nice asynchronous notes app with NodeJS, Express, and MongoDB. We can also make a mashup based on this app by using a couple of public APIs to enrich the note-taking experience.

Here are some ideas to get you started:

- **Screenshots**: Try to detect, if the note includes a URL. If it does, you can either use our Screen-Shot API from assignment 08 or a remote service (e.g. [https://screenshotlayer.com/](https://screenshotlayer.com/)) to generate a small thumbnail of the site and display it underneath the note.

- **Hash-Tags or Keywords**: Parse the content of the Note and display related content. One way to do it is to pass the content of the note to a text analysis API, like Aylien ([http://aylien.com/text-api-demo](http://aylien.com/text-api-demo)). You can use this service to automatically label the notes or to extract keywords that you can use a search-term for the twitter or flickr API. Be creative!

- **YouTube Videos**: If there is a URL in the note content that matches a youtube-URL, embed the video.

Put all your source code into the folder ‘task2’ (except for the node_modules folder).
Task 2b: Song Streaming Meta Search. Difficulty: Intermediate

If you want to start something new, you do not need to do Task 2a. You can build a “mash-up” from scratch.

If you choose this task, you are asked to create a search engine that allows users to find out where they can legally stream a certain song – maybe to inform their decision to which service they want to subscribe.

Use as many search APIs of streaming services as possible (Spotify, Deezer, Rhapsody/Napster, Microsoft Groove, etc) and create a cool, image-heavy mashup.

Also, you can utilize other music APIs to display information about the artists in a dedicated area of the page (e.g. LastFM, Songkick). It also makes sense to cache search results by inserting them into a database. Why not use MongoDB for that?

Put all your source code into the folder ‘task3’ (except for the node_modules folder).

Submission

Put the solution to Task1 – Task2 into a ZIP folder. However, to keep the submission small, please do not include the node_modules directory. The correctors will do an npm install and do not need the node_modules. Please turn in your solution via UniWorX. You can form groups of up to three people. After the submission deadline, push your solution to our GitHub repository.

The MIMUC team wishes you happy holidays and a good start into the year 2016!