Caroline Carlsson
Department of Media Technology
Linnaeus University
cc222js@student.lnu.se

Rebecka Green
Department of Media Technology
Linnaeus University
rg222dz@student.lnu.se

Torbjörn Holgersson
Department of Media Technology
Linnaeus University
th222mw@student.lnu.se

Enhancement, design and implementation of the VXO@Hand mashup- service

Introduction

For this assignment we have used the API[1] from Foursquare[2], Wikipedia[3] and Youtube[4]. The mashup is continued developed with Node.js[5] version 8.10.0 as backend and Angular[6] as frontend. This mashup is intended for people currently located in Växjö and also in need of several different information sources. If we manage to host the site it will be accessed at www.carolinecarlsson.se/mashup/

Approach to solving the given problem and tasks

The new components in the mashup are generated for each API that we have used for the assignment. There are also some new services implemented to the project. The three new components with included services was divided between us in a way that Caroline Carlsson worked on Foursquare, Rebecka Green on Youtube and Torbjörn Holgersson on Wikipedia.

Foursquare

We are retrieving data from coffee shops that are currently open and

[1] *Application programming interface*
[2] https://foursquare.com/
[3] https://www.mediawiki.org/wiki/API:Main_page
[4] https://www.youtube.com
[5] https://nodejs.org/en/
[6] https://angular.io/

based on recommendations from the Foursquare API. To retrieve this data a call is being made in the backend, see the comments in the code for more explanation. These markers are displayed on the map in form of a coffee cup. Depending on what time you visit the mashup you will see a list of coffee shops in Växjö. If there are no open coffee shops a text will be displayed where the shops otherwise would be listed, see figure 1 below.

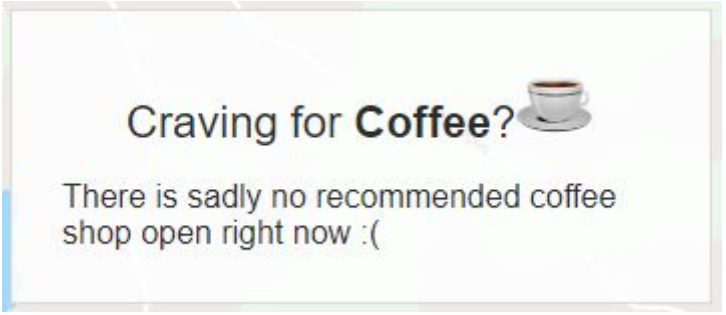

Figure 1, Div displayed when there's no recommended coffee shop open.

The users can also interact with the map by clicking on the coffee marker displayed on the map along with the flickr markers. However it can be hard to identify the markers unless you zoom in to the map since most of the markers seems to be located downtown. When you click on a marker a window will appear on top of this marker with the name of the Coffee shop as well as the street name where it is located.

Wikipedia

Based on the given keyword that, for the time being, is implemented statically in the code the API gets the data from Wikipedia and present it for the user with the title of the page and a snippet of the information that is available on Wikipedia's web application and as well in their database. The Wikipedia component (Carlsson, Green, Holgersson, 2018/2018) is based on Janosch Zbick's tutorial ("Tutorial videos about Mash-Up development with Angular JS.", n.d.) but are extended further and will be presented later in the further development section.

If the user finds the information presented in the snippet to be interesting and do want to know more about the topic the user can choose to go to Wikipedia and read more. The user can do this by pressing on the title of that particular snippet. As can be seen in the code the title itself is generated thanks to the loop function on line ten in wiki,component.html and later presented as an href[7] attribute (Carlsson, Green, Holgersson, 2018/2018). All the data from Wikipedia that is displayed on the web application are as mentioned before gathered using Wikipedia's API. The endpoint URL is as seen below

[7] Hypertext REFerence

```
https://en.wikipedia.org/w/api.php?action=query&origin=*&format=json&list=search&utf8=1&srsearch=Växjö
```

and was obtained from mediawiki (”API:Main page - MediaWiki”, u.å.) and the word for the search in Wikipedia's database is ”Växjö”. One thing to consider in this URL is that `origin=*` was missing or rather not included and was needed in order to avoid a CORS[8] conflict. After the data is obtained it is put into an array as JSON[9] and can then be looped through. A function is then used to get into the correct scope where the wanted data is located.

Youtube

The user have the opportunity to watch Youtube videos located in Växjö. The videos are implemented on the site as an list where the user can see the name of the videos (Fig 2). The module uses information from youtube rest API version 3. The communication between the client and server is established through Angular's http client module.

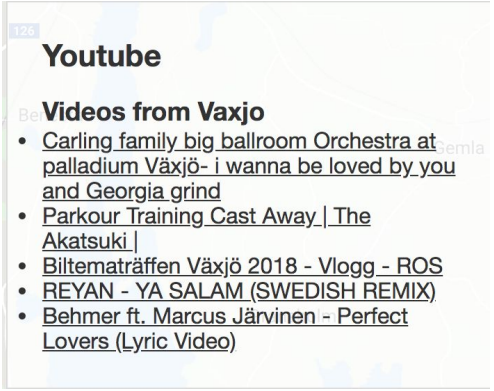

Fig 2: list of available videos from Växjö.

This http client module sends a GET request with necessary parameters such as API-key and Växjö citys GPS coordinates to youtube's api, which corresponds with a JSON formatted response.

The first idea was to embed the videos on the site to directly make them available for the users, but due to security restrictions related to CORS, there was no room to fix this within the current timeframe. It would be advantageously to represent each youtube movie with a marker on the map. However, this is not possible as the answer from Googles API does not contain GPS coordinates for the current videos. A more detailed discussion about further development will be will be presented later in this report under the further development section.

[8] Cross-origin resource sharing
[9] JavaScript object notation

Mashup pattern

The pattern that we have implemented is the assemble pattern, more specific the filter. We retrieve data and chose to remove the parts that are not necessary for our mashup. Additionally we are using location mapping where we locate data on the map with the longitude and latitude.

On the mashup you can view current coffee shops that are open as well as recommended by the foursquare community. While the Wikipedia implementation presents data of the top ten pages from Wikipedia's database containing the keyword Växjö. The users also have the opportunity to watch videos from Växjö which is implemented with Youtube.

Conclusions and Reflections

Developing a mashup that is both functional and fully thought out is not an easy task. Even though the implemented code works in one workstation it would not necessary work on another. A way to solve this problem would be to use a virtual environment with a already set of rules on what versions the different softwares would have, this would also make it easier to use source code management such as Git. A design method was not used in this project, this however was never considered to be needed due to the lack of time and a concrete explanation on what the system should have. In a way traces of kanban would be seen in the chat in how the work was divided between us and while no kanban board was displayed or used we knew how the work was going thanks to a ongoing chat conversation.

Further development

To further develop this mashup some login functionalities would be preferable as it would enhance the possibilities for more personal tasks. For example usage of the foursquare API could include some account based actions such as add coffee shops to your personal list, or show coffee shops that already exists in your list or a list by someone that you are following. Furthermore it would be helpful for the user to add some mouse events for this API. When a user is hovering over the shops in the list, they should get highlighted on the map as well as getting the opening hours displayed on top of the marker. The text could for example be “Closes in 3 hours” and based on the time left the background of the info window, or marker on the other hand, could change its color. And of course, design that affords mouse hovering must also be implemented in this scenario.

For further work on Wikipedia's API it was started but not finished to implement that the user would be able to input keywords in order to search the

Wikipedia database. This would mean that the search word in apiUrl would not be static anymore and the apiUrl would be dynamic. This feature was however not possible to implement correctly due to the trouble with the execution of the code. Even though it could display an undefined object the search word was not able to be changed before the execution hence no word was at al implemented in the code. A suggestion to solve this problem would maybe be eventemitter (ref) but further research is needed.

As mentioned before it would be an advantage for the users if the video was indicated with an marker. This would give the user a better overview of where the video was recorded and be more consistent for the user since both Foursquare and Flickr use this method to visualize their data. Further development with Youtube's API could also be to implement more “intelligence” in term of selections of videos, for example, sort by popularity or generated activity. This would help the user to get a better picture of what is happening in Växjö at the moment.

REFERENCES

API:Main page - MediaWiki. (u.å.). Accessed 11 December 2018, from https://www.mediawiki.org/wiki/API:Main_page

Carlsson, C., Green, R., Holgersson, T. (2018). *vxo@hand mashup done in Angular. Contribute to T-bear/ng_mashup development by creating an account on GitHub*. TypeScript. Accessed from https://github.com/T-bear/ng_mashup (Original work published 2018)

Tutorial videos about Mash-Up development with Angular JS. (n.d.). Accessed 11 December 2018, from https://mymoodle.lnu.se/mod/forum/view.php?id=2055816