CS8803 – Special Topics | Mobile Apps & Services

Peer programming

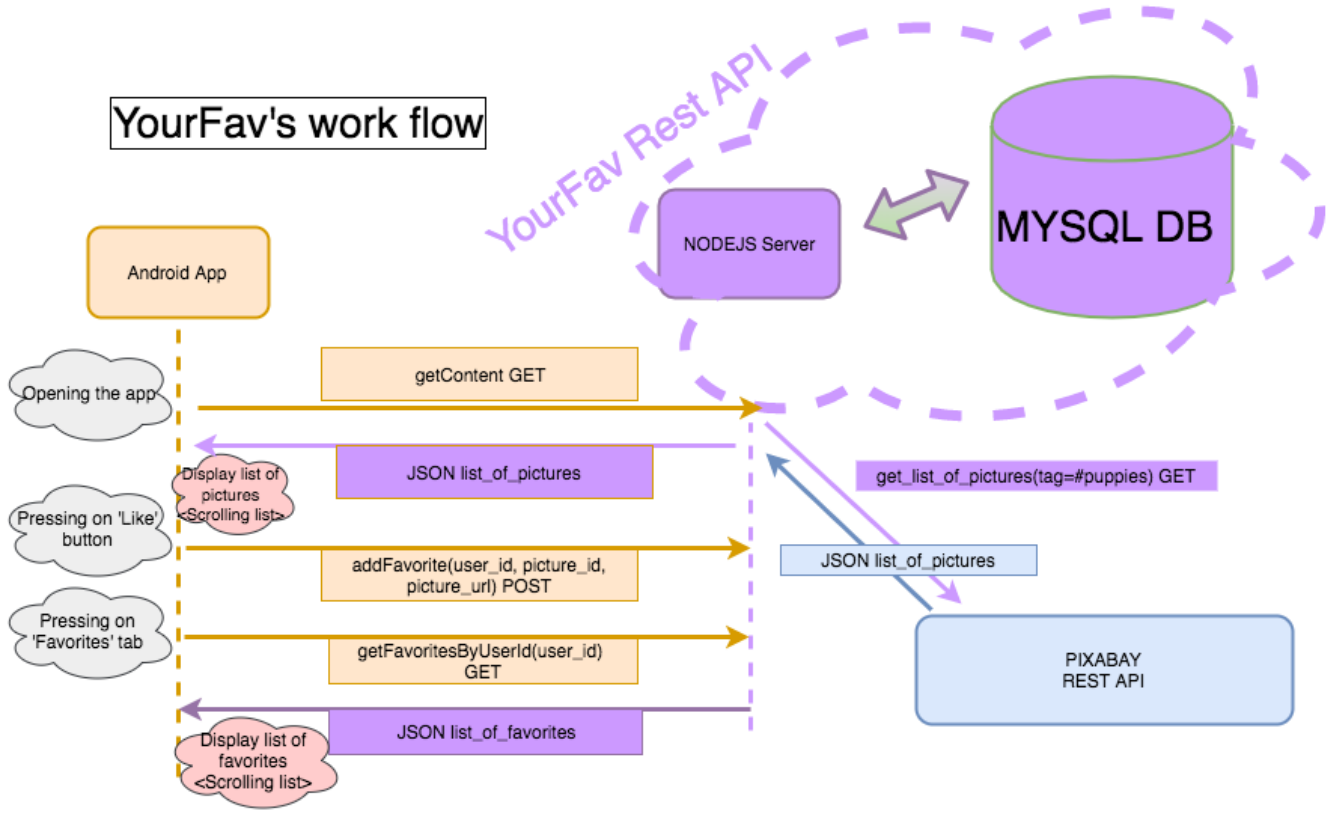
Alexis Durocher – MSCS Georgia Tech & Vanessa Servais – MSECE Georgia Tech

In this third assignment, the goal is to build, from scratch, a complete mobile APP. This project includes a front end part (Android in our case) and a backend part (NodeJS & MySQL embedded in a REST API).

---

1. General Overview - YourFav

YourFav is an Android mobile application that allows any users to connect and see a list of pictures from an external API (pixabay) with a tag predefined (here, we displayed pictures of puppies). The user can scroll among the list of pictures and choose some favorites which will be saved in his profile (our app's database).
A **pseudo-code** version of our app's main workflow :



There is an interaction between our NodeJS server and the Pixabay API before storing information from this external API in our own DataBase.
This app is not deployed yet, it works locally on a Localhost:3000.

Requirement & set up:

- MySQL DataBase | *the credentials to connect to the database are set in the file src/dbconnection.js*
- NodeJS with npm | ***npm install*** *&* ***npm start*** *in the root folder to launch the server locally.*
- An Android emulator to run the front-end app.

2. Backend

Our Backend runs locally and includes a webserver base on NodeJS and a MYSQL database.

a. Webserver

We chose NodeJS to build our backend because NodeJS provides many library to make an easy set up (express-generator for instance) of a complete running server. From this, we could scale up our server and fit it to our needs.

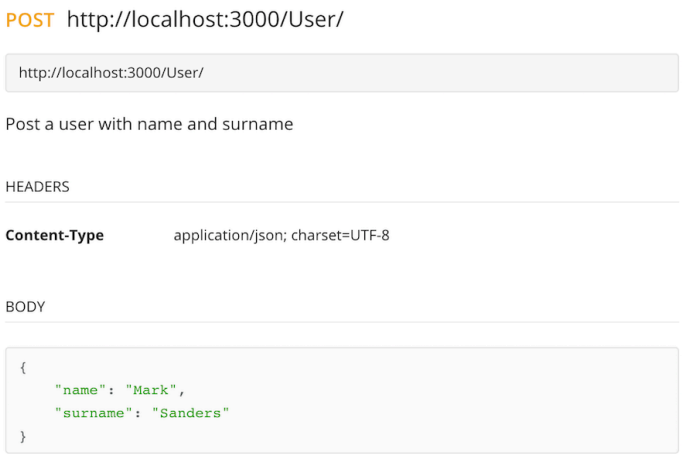
- **./public** : Every files that can be publically accessed. Here, only the report.pdf and our postman collection (API docs) are available. But this folder only makes sense during the deployment phase and can, then, be ignored here.
- **./node_modules** : All NodeJS dependencies linked with npm and managed via the package.json file**.**
- **./bin** : All executable files.
- **./sources:** Our different routers (where we define the actions resulting from calls to certain endpoints of our API).
- **./src** : Contains the model of our data. The SQL command to be executed from the NodeJS server to the database. The DataBase connection's parameters. Some .sql files to build and keep our database consistent on our different computers**.**
- **./views** : Can be ignored.

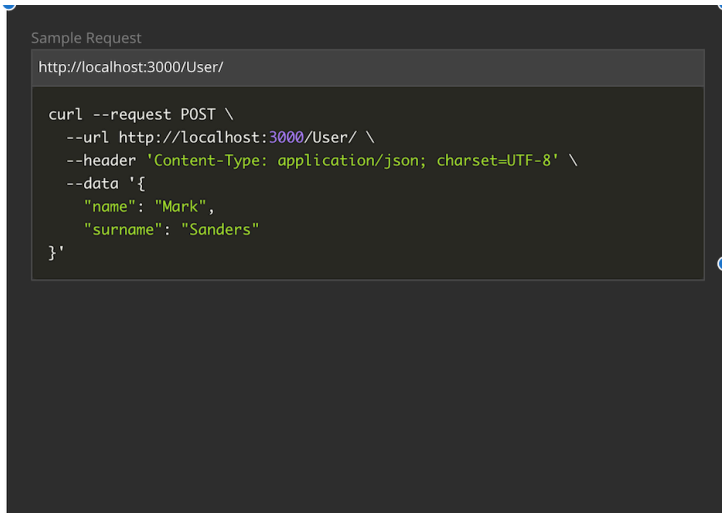
The main app is defined on **app.js**.

b. REST API

Our REST API doc can be automatically generated with our Postman collection. All our features are not used in the app.

Feature 1
POST a new User | *when a new user sign up in the app*





Feature 2

POST a new Favorite picture | *When the user (here id = 2) click on 'like' button, the picture is then stored as a new user's favorite in the Database.*

POST http://localhost:3000/Content

http://localhost:3000/Content

Post a new favorite to the current user

HEADERS

Content-Type	application/json; charset=UTF-8

BODY

```
{
    "id" : "1903313",
    "url": "https://cdn.pixabay.com/photo/2016/12/13/05/15/puppy-1903313_15
    "user": 2
}
```

Sample Request

http://localhost:3000/Content

```
curl --request POST \
  --url http://localhost:3000/Content \
  --header 'Content-Type: application/json; charset=UTF-8' \
  --data '{
    "id" : "1903313",
    "url": "https://cdn.pixabay.com/photo/2016/12/13/05/15/puppy-1903313_1
    "user": 2
}
    '
```

Feature 3 & 4

GET all contents if no id is provided OR user's list of favorites if its id is provided | *Depending on which tab the user click, he will see either the complete list from pixabay API, or his own favorites.*

GET http://localhost:3000/Content

http://localhost:3000/Content

GET all contents from pixabay API

HEADERS

Content-Type	application/x-www-form-urlencoded

Sample Request

http://localhost:3000/Content

```
curl --request GET \
  --url http://localhost:3000/Content \
  --header 'Content-Type: application/x-www-form-urlencoded'
```

GET http://localhost:3000/Content/1

http://localhost:3000/Content/1

GET all favorites of user 1

Sample Request

http://localhost:3000/Content/1

```
curl --request GET \
  --url http://localhost:3000/Content/1
```

3. Front-end: Android application & Design.

4. References

Here are the tutorials and references we used to build our app.

REST API tutorial for NodeJS and Mysql :

https://medium.com/@avanthikameenakshi/building-restful-api-with-nodejs-and-mysql-in-10-min-ff740043d4be

https://jinalshahblog.wordpress.com/2016/10/06/rest-api-using-node-js-and-mysql/