HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch

TWEB
Olivier Liechti
Paul Ntawuruhunga
Miguel Santamaria

Project 1 - GitHub Analytics

Goal

For this first project you'll work mostly on the front-end to create an app that uses the github REST API to analyze repos and present informations to the user via interactive charts and/or sticky content.
The backend is optional since you can send request directly from the client app, but we strongly recommend to have one: you will indeed be limited with 60 requests per hours if you are not identified, whereas you will have 5'000 requests per hours if you are, and you really don't want to store your token on the client part (since anybody can access and use it), trust us... You can also store data in a database with a backend, which will give you more freedom and control.

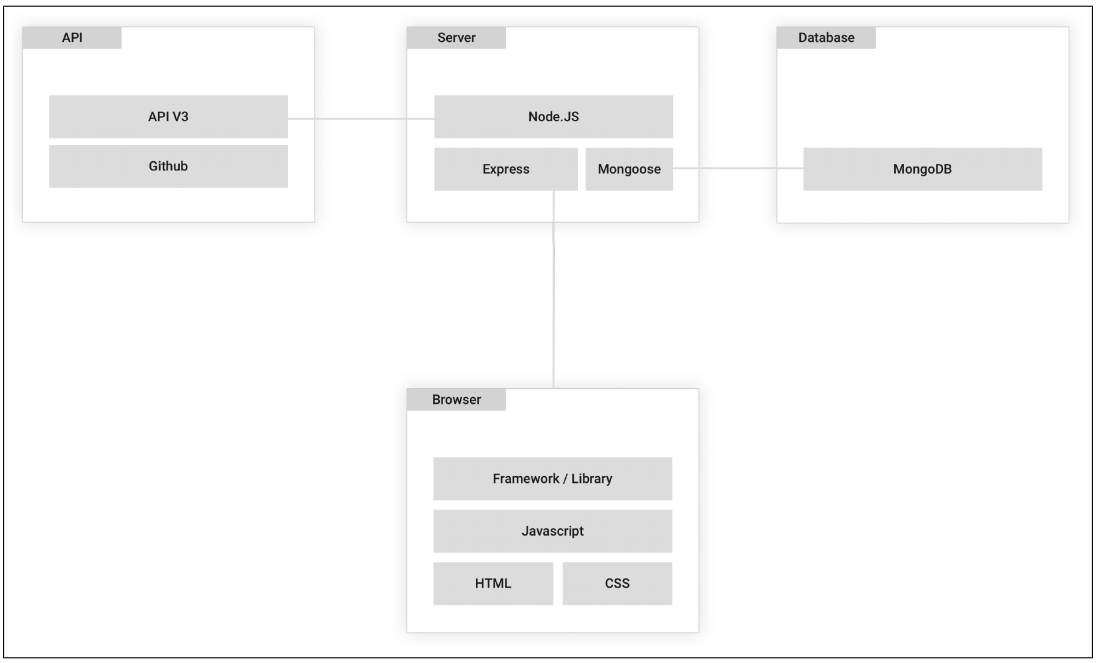
Rules

You are going to build the whole application by yourself, so pack your dancing shoes and put your warmest gloves!

- Your are going to do this project by **groups of 2 people**.
- Your code will be hosted on GitHub.
- You will write a succinct MarkDown file located at the root of your repository. A developer must be able to understand what you did, how to deploy the project, and how to use it.
- The project is separated in **3 phases**.
- Deadline: **28.10.2018, 23:59**.
- This project's grade has a 30% weight on the final labs grade.
- You have to use the **GitHub REST API** by your own (no use of existing libraries, we want you to manually do the requests, because otherwise you will not learn).
- You are going to code at least the frontend, then the optional backend if you want to fetch and store data from GitHub.
- As always, we love creativity, trendiness and fun-but-useful features, so feel free to think outside the box, this can grant you some bonus.

Technologies

- Frontend
  - Templates, Bootstrap, etc.
  - JavaScript (JQuery, Vue.js, React, Angular, etc.)
  - Charts.js, Chartist, Plotly.js, Highcharts, D3 (PGM level), etc.
- Backend
  - Express (you can use the Express Generator of Yeoman to have a strong skeleton).
  - MongoDB, Mongoose
- Hosting
  - Frontend: GitHub Pages
  - Backend: Heroku

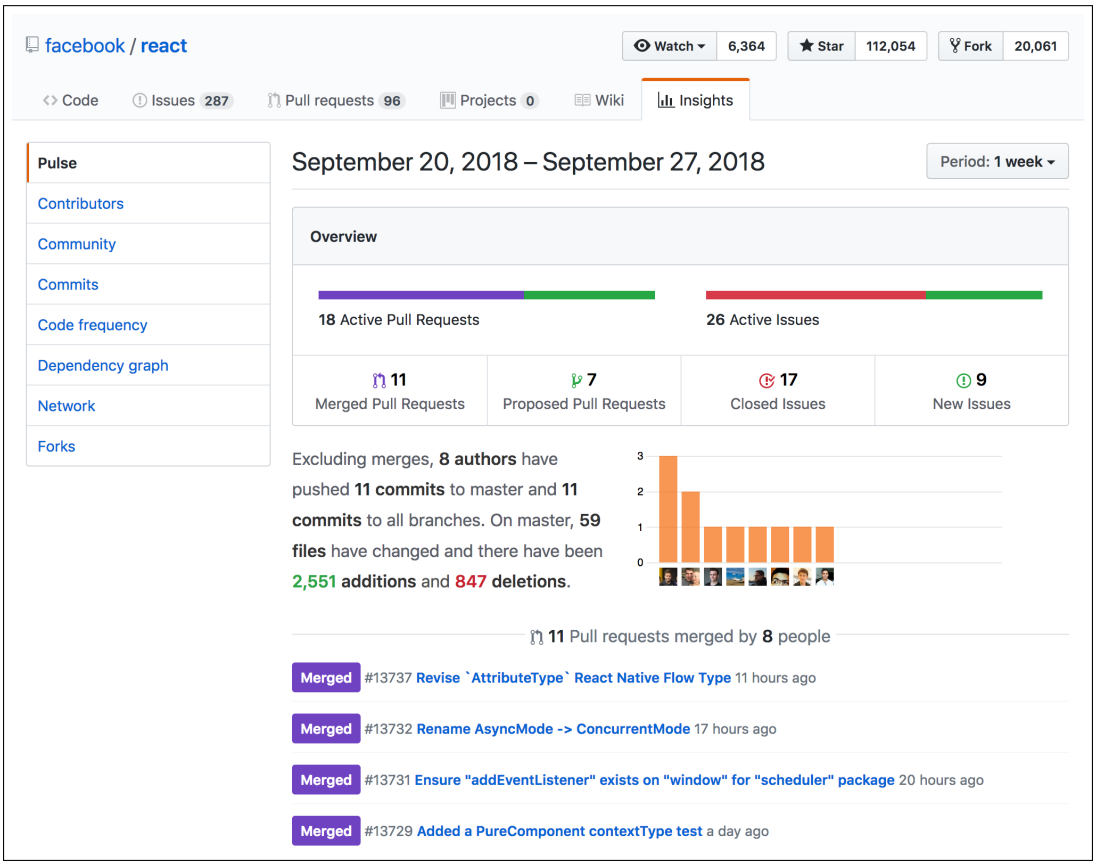


Sketch

- Before coding anything, **take the time to define what you want to build**. Use paper or graphical editors.
- What is the **question** that you will answer in your application? In other words, what is the information that your users will obtain and **why does it matter**?
- **How will users interact with your app**? What will they see when they arrive on the main page? Will the interaction take place in a single page or will the navigate between multiple pages? What will the see on these pages?
- How will you deal with **tasks that take a long time**? For instance, if it takes minutes to run a new analysis, what will the user see?

Examples

- GitHub's insights



- Last year's projects:
  - GitHub Analytics Issues
  - GitHub Aliens
  - GitHub LightHouse
  - Compository
  - Gitanalytics
  - world fork visualization
  - Photo Mosaic
  - Commit Messages Analytics, GitHub Habit Analytics

Phases

Phase 1 - 27.09.2018 (week 2)

- Form groups of 2.
- Agree on the data to be extracted from GitHub.
- Create a mockup of your app.
- Setup client-side project.
  - *(Optional) start with a template of choice.*
  - *(Optional) add a charts library.*
  - *(Optional) use a framework of your choice such as Vue.js, React, Angular.*
- *(Optional) Setup server-side project.*
  - *Add a linter.*
  - *Add a testing framework.*

Phase 2 - 04.10.2018 (week 3)

- Deploy client app to GitHub Pages.
- Start unit tests with sample data.
- Use GitHub REST API.

Phase 3 - from 11.10.2018 to 28.10.2018 (weeks 4-6)

- Implementation
- Project's delivery (we will give you a Google Docs form, or whatever).

HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch

TWEB
Olivier Liechti
Paul Ntawuruhunga
Miguel Santamaria

iICT

Evaluation

Basis

Criteria	Mandatory for 4.0
The app is online and publicly available (e.g. on GitHub Pages and Heroku)	yes
It is possible to test the app locally (with live-server, or node, or any other tech)	yes
Someone arriving on the app, without prior knowledge of the project, understands what it is about and can use it.	yes
The app uses a nice visual template.	yes
There is a repo with a README.md file that explains what the project is about, how to run it locally and how to build it.	yes
The build process invokes a linter. The linter is happy with the quality of your code (no error).	yes
The app fetches data from GitHub and presents it in the UI.	yes
The app works (no crash, no obvious problem in the interactivity, etc.)	yes
You wrote some useful tests for your app.	yes

Bonus

Criteria	Weight
Extra effort has been put in the UI/UX.	+0.5
Extra effort has been put in the depth of the analysis ("you are not just sorting developers per commit count").	+0.5
You use a database.	+0.5
You coded a backend that is useful.	up to +1.0
Extra effort has been put to make the app "sticky" and/or "viral". There is something about it that makes it particularly original, fun. Or there is something that will make people come back to it.	up to +1.0
There is something else that you have done and that you think deserves a bonus.	+0.5

This is how you can earn extra points to go beyond the 4.5 grade.
If you believe that you have done something particularly interesting/challenging/valuable, **sell us the idea**. You have to convince us not only by showing it to us, but also by writing about it (and make the contribution very clear). We don't expect a 15 pages report, but something that could be published on your portfolio. Every "bonus" opportunity that you take should have its own "pitch" (in markdown).