BigLab1

Thursday, April 8, 2021 11:22 AM

12/04->4 weeks->10/05 final submission only that day
GitHub ClassRoom: composition of the group+private repository
At the end use repository to submit code.

1. Github classroom instruction
2. General BigLab1 assignment (different toDo list), with rules for development and submission
3. Assignments for each week(1a,1b,1c,1d)

1 BigLab submission: 0/0.5/1 point to each members of the group

Solution only at the end, ask questions!

Web Applications I / Applicazioni Web I – 2020/2021

Instructions for management and submission of BigLabs through GitHub Classroom

1 Goal

The laboratories submissions will be done by pushing your group project to a specific repository, that will be created by *GitHub Classroom*.

Note: this is the same mechanism that will be used for the final exams.

2 Basic Requirements

The basic requirement is having a personal GitHub account and being familiar with Git commands.
If you are new to Git or GitHub, you can have a look to the GitHub starter course.

3 Accessing BigLabs

In the text of the BigLabs, you will find a link to access the GitHub Classroom's assignment. Following that link, you will have to login in your GitHub account.

4 First login

At the first login on GitHub Classroom, you must associate your GitHub user to your student's ID.
This is needed only once to join the 'Classroom'.

	
1. Authorize your GitHub user to access the Classroom platform.	2. Associate your account with your student identity, by selecting your name from the full list of students. If you are not on the list, please contact one of the teachers on Slack.

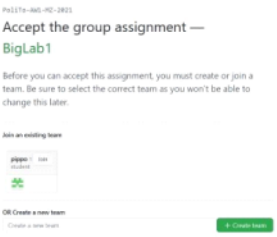
5 Accept the group assignment.

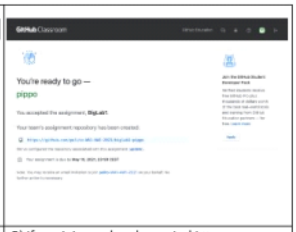
5.1 First BigLab: join (or create) your group.

Reach the assignment with the link provided to you in the text of the BigLab.

At your first login, you must check if one of your colleagues already created your team on GitHub Classroom. If yes, you must join your team clicking on the "join" button. In the example below, only a team called "*pippo*" was already created.

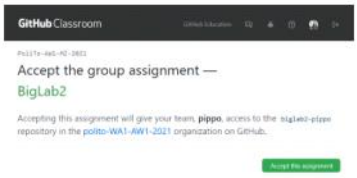
Otherwise, you must create your team using the form on the bottom of the page. It is mandatory to use the **same group name** specified in the Google Form filled at the beginning of the course. Please, **do not join** a wrong team.



	
A) If you are the one that creates the team, you will see a message that your repository is under creation.	B) If you join an already created team, you are ready to go.

5.2 Second BigLab: accept the assignment.

For the second BigLab, you just have to "accept the assignment" to get a copy of the skeleton repository.



6 Your repositories

After creating or joining a team for each BigLab, you will get a link to the repository in which you have to work. The link contains a common part (the BigLab info), concatenated with your Team name.

Now you have your repository and may freely work on it. The repository is private, and only your team and the teachers may access it.

7 Developing the project

All the development of the project **must** take place inside this repository.

We suggest you create a **new branch** for each part of the BigLab, week by week, to keep track of your progress.

For instance, when you start working, you can create a branch called "*week1*" and commit your work on that branch. When you finish working on the first week, you merge everything into the main branch and move to a newly created branch "*week2*".

Please, notice that this is not mandatory; it is just a suggested developing methodology.

8 Submitting the project

- The final project must lie in the **master/main branch** (the content of other branches will not be considered or evaluated).
- Ensure that **all** the needed packages will be retrieved with `npm install`.
- The last commit (i.e., when you want to submit the whole BigLab), **must be tagged with 'final'** (see below for instructions).

We assume that, if your **master/main branch is empty**, or you did not push a commit tagged with **final**, you decided **not to submit** the BigLab for evaluation.

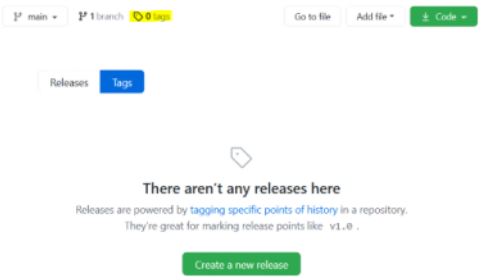
No further commits are considered after the BigLab deadline.

Note: for tagging a commit, you may use (from the terminal) the following commands:

```
# ensure the latest version is committed
git commit -m "...comment..."
git push

# add the 'final' tag and push it
git tag final
git push origin --tags
```

Alternatively, you may insert a tag from the GitHub web interface (click on "tags" and follow the link "Create a new release").



01UDFOV/01TXYOV – WEB APPLICATIONS I

BIGLAB 1: TODO LIST IN REACT

During the four weeks of the first BigLab, you will develop the front-end for a web-based task manager using React. To create your repository, you must login to GitHub Classroom and create or join your group. For more details, please have a look at the GitHub Classroom instructions. Here you can find the links for the BigLab repository on GitHub Classroom:

- Web Applications I: https://classroom.github.com/g/Pxim-2Bu
- Applicazioni Web I [A-L]: https://classroom.github.com/g/hpJQI5GQ
- Applicazioni Web I [M-Z]: https://classroom.github.com/g/bEZrEEhs

To better keep track of your progress, we suggest you work incrementally "week-by-week", e.g., by creating, inside your repository, a branch for each week of the BigLab.

WHAT ARE WE BUILDING IN THESE WEEKS?

a) During the first week, you will start setting up the front-end for the task manager. To do so, you will use the React-enabled version of Bootstrap. The page has to be static; you do not have to implement any user actions. For the visual appearance of the web application, you can get inspiration from the screenshot in the next page.
b) In the second week, you will re-structure your web-based task manager to exploit the component-based approach of the React framework. Specifically, you will divide your application into different components, and you will identify the state and props required to store and visualize tasks, starting from the tasks data structure.
c) During the third week, you will implement the filters (i.e., important, today, next 7 days, private), and you will add the possibility to create and add new tasks.
d) In the last week, you will learn how to restructure your application to use routes for supporting multiple "pages", e.g., by developing a different route for each filter. Furthermore, you will allow users to delete and edit tasks.

EVALUATION CRITERIA & DEADLINES

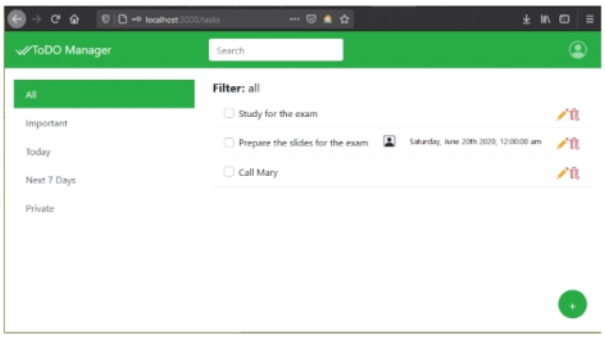
The points received for your work are added to the final exam score to each member of the team.
We will follow these evaluation criteria for evaluating your submission:

- The team members will receive 1 point if the submitted React application is *complete*, i.e., it successfully implements *all functionalities of the 4 weeks*, i.e., points a), b), c), and d).
- The team members will receive 0.5 points if the submitted React application is *partially complete*, i.e., it successfully implements the functionalities of *at least 2 weeks* (e.g., the team implemented only points a) and b)).
- The team members will receive 0 points otherwise.

The assignment must be submitted in the master/main branch before **Sunday, May 9 at 23:59 CEST** (see the the GitHub Classroom instructions for the details on the submission procedure).

Example Screenshot

Note: the actual layout and appearance of your application may be different from the screenshot. Only the functionality mentioned in the 4 points is mandatory, while you may freely choose the graphical presentation.



01UDFOV/01TXYOV – Web Applications I

BigLab 1a: Getting started with React

What are we building this week?

We will start setting up a React-based front-end for the web-based task manager. The appearance of the front-end will be built according to the general specification of the first big-lab and the code you developed in the previous labs (see the screenshot in the general specifications of the first BigLab).

Step-by-step instructions:

- Create a new React app with the command `npx create-react-app` (https://github.com/facebook/create-react-app)
- Use the React-enabled version of Bootstrap and its components to structure the front-end of your web-based task manager. You can find the documentation of React Bootstrap at https://react-bootstrap.github.io.
- For the moment, the whole application can be organized as a single component, e.g., an "App" component stored into App.js file. In other words, you are not requested to split your application into different components.
- As in Lab 2, populate the webpage with a few dummy tasks (4-5).
- **Beware**: you do not have to implement any user actions; filters are not expected to work, neither the button for adding new tasks, etc.

Hints:

1. *Carefully read the general specification of the first BigLab before starting to design the front-end: https://github.com/polito-WA1-AW1-2021/course-materials/blob/main/labs/BigLab1/BigLab1.pdf*
2. *For handling dates and times, you can exploit day.js.*
3. *You can use the code developed during the lectures or during the previous labs as examples or to get inspiration.*

01UDFOV/01TXYOV – Web Applications I

BigLab 1b: React Components' State

What are we building this week?

You will continue to re-structure your web-based task manager to exploit the React framework and its component-based approach. Specifically, you will divide your application into different components, and you will identify the state and props required to store and visualize tasks.

Step-by-step instructions:

- Organize the page using React functional components stored in different files. Your application, for example, might have different components to manage the navigation bar, the sidebar, and the list of tasks with their props.
- Instead of having hard-coded tasks, plan and implement the management of the state and the props of the different components to store a list of tasks. To store tasks, use a proper JavaScript data structure, e.g., the one adopted in Lab 1. Remember that data is stored in the client, since we are still working without a server. Decide which component will hold the state, and how information propagates using props.

Hints:

1. *The general specification of the first BigLab can be found at: https://github.com/polito-WA1-AW1-2021/course-materials/blob/main/labs/BigLab1/BigLab1.pdf*

01UDFOV/01TXYOV – Web Applications I

BigLab 1c: Filters & Forms

What are we building this week?

You will add interactive functionality to your web-based task manager. Specifically, you will implement filters, and you will enable users to create and add tasks.

Step-by-step instructions:

- Consider the full life cycle of the application, and create all the suitable callbacks to implement the required filters (i.e., **all**, **important**, **today**, **next 7 days**, **private**). When a filter is clicked, only the tasks respecting the filter must be visualized. The filter mechanism should not affect the original data source. In other words, filters should affect the visualization of tasks, only.
- Update the web application developed so far to allow users to *dynamically* and *interactively* insert new tasks. To do this, use the Modal component of React Bootstrap: when the user clicks on the '+' button, open a modal and ask for a new task (with all its properties) by filling a form. By submitting the form, add the newly inserted task into the JavaScript data structure.
- The form should be validated before its submission, and you should use proper error messages when inconsistencies are found, e.g., when some mandatory fields are missing.
  Beware: form fields should be validated, to reinforce mandatory field and to avoid having not admitted values.

Hints:

1. *The general specification of the first BigLab can be found at https://github.com/polito-WA1-AW1-2021/course-materials/blob/main/labs/BigLab1/BigLab1.pdf*
2. *The documentation of the Modal component in React Bootstrap can be found at: https://react-bootstrap.netlify.app/components/modal/#modals*

01UDFOV/01TXYOV – Web Applications I

BigLab 1d: React Routes

What are we building this week?

You will update your React-enabled task manager to support multiple "pages" (through routes), and you will allow users to delete and edit tasks.

Step-by-step instructions:

- Enable users to delete specific tasks: add a trash icon (🗑) near each task and set up its corresponding callback to remove the task from the JavaScript data structure that is used to store tasks. When a task is removed, the list of tasks should update automatically.
- Enable users to edit an existing task: add an edit icon (🖉) near each task and re-use the modal for creating a new task, pre-filled with the available information. When a task is updated, the list of tasks should update automatically.
- Update the task manager to support multiple "pages". In particular, *restructure* the React application to implement each filter as a separate route. When the user reloads the webpage from inside a given filter route, the same filter should be applied when the page is ready.

Hints:

1. *The general specification of the first BigLab can be found at*
   https://github.com/polito-WA1-AW1-2021/course-materials/blob/main/labs/BigLab1/BigLab1.pdf