MissBug

CRUDL end 2 end

Node.js

Phase 2 – REST API + Sorting, Paging, Filtering

Model

The bug should now have the following properties:

```
{
    _id : "abc123",
    title : "Cannot save a Car",
    description : "problem when clicking Save",
    severity : 3,
    createdAt : 1542107359454,
    labels : ['critical', 'need-CR', 'dev-branch'],
}
```

Backend

1. Convert your backend to provide a RESTful API on the entity bug, with an API folder that holds the routes, controller and service of the BUG Entity.

   Support server side filtering, sorting and paging:

   1. Sorting examples:
      a. ?sortBy=title
      b. ?sortBy=severity
      c. ?sortBy=createdAt&sortDir=-1
   2. Paging: ?pageIdx=3
   3. Filtering:
      a. by txt
      b. by min severity
      c. by labels (check if any of the labels is included)

   Use postman to test your API

2. Now add a user entity to your project (user.json) and populate it with a few users.
   Here is your MODEL:

```
{

    _id : "u101"
    fullname : "Muki Ja",
    username : "muki",
    password : "muki1234",
    score : 100

}
```

Implement full CRUDL for the user Entity using routing and RESTAPI. Should be pretty easy if you copy the “bug” folder and refactor it to our needs.

Use postman to test your API

*Note – This is not for authentication (no login/signup functionality just yet). Only a full working CRUDL.

Frontend

- Update your frontend to use the REST API
- Use nicer urls for your frontend routing
- Add <UserIndex /> page that allows handling full CRUDL on users. (Showing a list, removing, adding and updating a user)
  (later we will allow only the admin to do these actions)

Backend - Add a cookie for usage limit

- Let's limit the user for viewing no more than 3 bugs during some time
  - Later on, we might want to encourage the user to signup and remove this limit, but this is out-of-scope for now
- So we need to keep track of the bugs the user visited
- From the backend we will send a cookie: *visitedBugs*
  in which we will store an array of visited bug ids
  (use JSON.stringify and JSON.parse where needed)
- Make sure the array is sent as cookie and saved by the browser (check the dev tools)
- Make sure it works by printing a message to the backend console:
  User visited at the following bugs: […]
- When user visits more than 3 different bugs we will respond with an error:

```
return res.status(401).send('Wait for a bit')
```

- Make that cookie last for 7 seconds
- Note that you can clear the cookies at any time using the dev-tools

Bonus – Get a PDF

Allow the user to download a PDF file of the bugs