Angular Assignment

PURPOSE

This assignment is a chance for you to showcase your skills and will hopefully form a good base for discussion further down the process. Here in European Dynamics, we believe that it's for the mutual benefit of both the team and the candidate to get a good taste of one another's way of work and thinking. This assignment is designed to evaluate the candidate skills regarding:

- Angular Components
- Angular Routing & Navigation
- Angular Services
- Angular HttpClient
- Use of RxJS operators
- Use of third-party Angular Components (Angular Material etc.)
- HTML & CSS

NOTES

- The screenshots included in this assignment have been taken from an implementation that uses Angular Material components (https://material.angular.io/). Moreover, the hints/links given in the instructions are also related to Angular Material components. However, please, feel free to use any Angular Components library you may be familiar with.
- Angular version to be used: 7+
- Even if you don’t manage to implement the whole assignment, please do send your work as described in the Deliverable section.
- Don't hesitate to contact us for any clarifications.
- You have **7 days** to complete the assignment.

DELIVERABLE

The deliverable should be the angular project (folder) *without* the *node_modules* folder. You can send it as:

- a zip file ***or***
- a GitHub (or any other online code repository) link

EVALUATION

The assignment’s evaluation will be based on:

- Implementation of core functionality
- Distributing functionality in components
- Code organisation
- Implementation of the ***optional*** requirements

OVERVIEW

A customer organisation has requested the creation of a discussion forum for its employees.

The basic backend functionality has been implemented and there is a need for a basic front-end that will display some mock db records in order to test the REST API.

INSTRUCTIONS

Create an angular application with the following three routes:

1. users

This page should consist of a data table that will display the users' data.
The table should have the following columns:

- Username
- Full Name
- Email
- Action

The *Action* column should not have a header text. It should contain a link (or button) *View Posts* that will navigate the user to the users/:userId route that will be explained later.

You must also implement *pagination* functionality. Pagination can be client-side or server-side. Client-side pagination means that all the data will be fetched from the backend and the data table component will perform the pagination upon these data. In the server-side pagination, the data table component requests only a portion of the data by including the page index and the page size as URL parameters to its request to the REST API (provided that the REST API supports this functionality).

The implementation of **server-side** pagination is ***optional***.

In order to fetch all the users' you can use the following REST endpoint:
http://5da8543fe44c790014cd4b86.mockapi.io/users

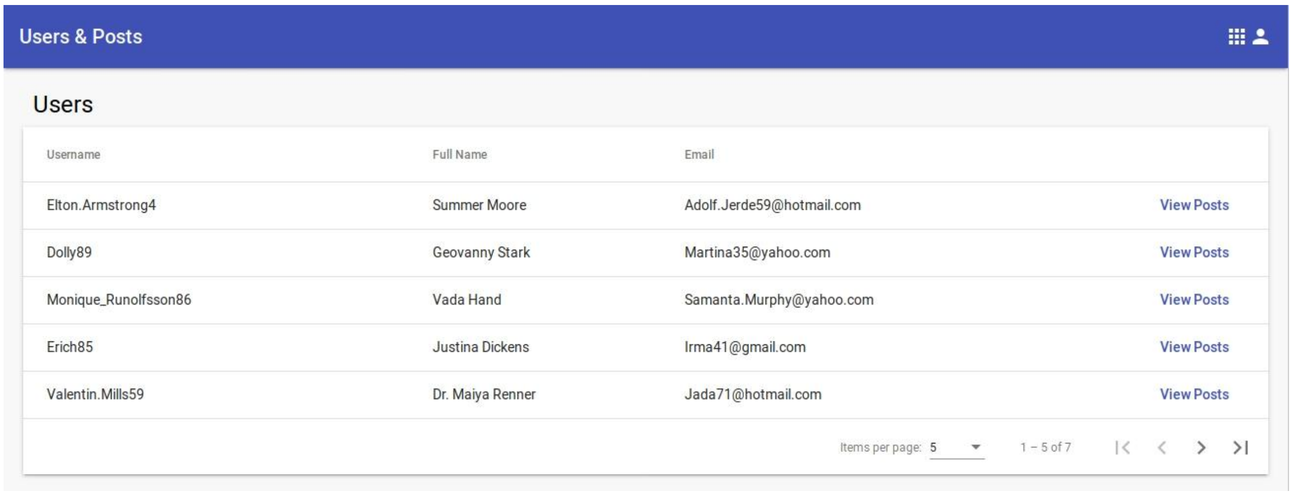
In case you wish to implement server-side pagination you can pass the following URL parameters to the previous endpoint:

- page: The page index
- limit: The page size

E.g. http://5da8543fe44c790014cd4b86.mockapi.io/users?page=2&limit=5

The data received (payload) from this endpoint will be in the following format:

```
{
  items: [
    {
      id: number,
      name: string,
      email: string,
      avatar: string,  // Link to an image
      username: string
    }
  ],
  total: number
};
```


Users & Posts

Users

Username	Full Name	Email	
Elton.Armstrong4	Summer Moore	Adolf.Jerde59@hotmail.com	View Posts
Dolly89	Geovanny Stark	Martina35@yahoo.com	View Posts
Monique_Runolfsson86	Vada Hand	Samanta.Murphy@yahoo.com	View Posts
Erich85	Justina Dickens	Irma41@gmail.com	View Posts
Valentin.Mills59	Dr. Maiya Renner	Jada71@hotmail.com	View Posts

Items per page: 5 1 – 5 of 7

<ins>Links</ins>

Table
https://material.angular.io/components/table/overview

Paginator
https://material.angular.io/components/paginator/overview

Button
https://material.angular.io/components/button/examples

2. users/:userId

In this page we want to display:

- user details (Name, email, avatar)
- all the posts that he/she has created

The posts can be displayed as a list of items. For each post you should display its title and its creation date.

The post's title should be a link that will navigate the user to the users/:userId/posts/:postId route that is explained later.

To fetch the user data you can use the following endpoint:
http://5da8543fe44c790014cd4b86.mockapi.io/users/:userId

e.g. http://5da8543fe44c790014cd4b86.mockapi.io/users/4

You shall get the *userId* from the <ins>route path parameters</ins>.

To the fetch the user's posts you can use the following endpoint:
http://5da8543fe44c790014cd4b86.mockapi.io/users/:userId/posts

e.g. http://5da8543fe44c790014cd4b86.mockapi.io/users/4/posts

The response of this endpoint will be an array of post objects. The post object format is:

```
{
  id: number,
  userId: number,
  createdAt: Date,
  title: string,
  body: string,
}
```

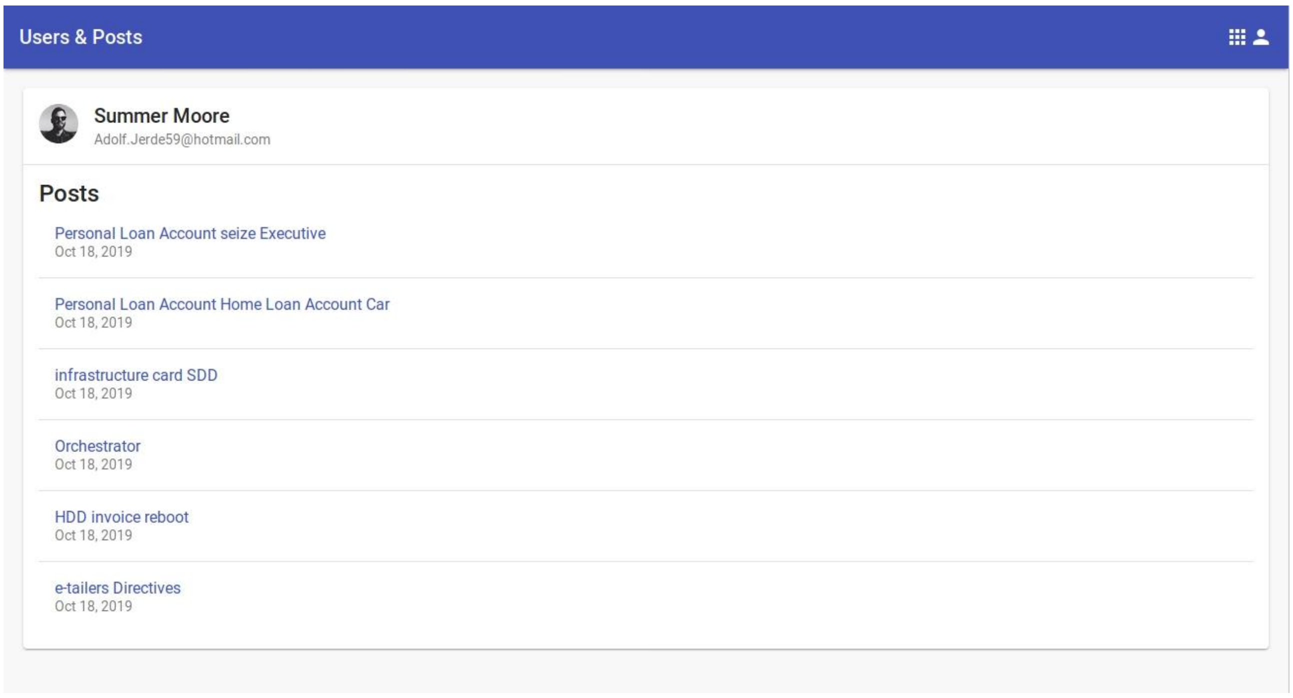
Links

Card
https://material.angular.io/components/card/overview

List
https://material.angular.io/components/list/overview

Hints

You can format the date display by using the Angular **date** *pipe*.



3. users/:userId/posts/:postId

In this page we want to display:

- the post (Title, creation date, body)
- all the comments related to this post

The comments can be displayed as a list of items. For each comment you should display the author's name and avatar, the comment's creation date and the comment's body.

To the fetch the post data you can use the following endpoint:
http://5da8543fe44c790014cd4b86.mockapi.io/users/:userId/posts/:postId

You shall get the *userId* and postId from the route path parameters.

e.g. http://5da8543fe44c790014cd4b86.mockapi.io/users/:userId/posts/:postId

To the fetch the post's comments you can use the following endpoint:
http://5da8543fe44c790014cd4b86.mockapi.io/users/:userId/posts/:postId/comments

e.g. http://5da8543fe44c790014cd4b86.mockapi.io/users/1/posts/1/comments

The response of this endpoint will be an array of comment objects. The comment object format is:

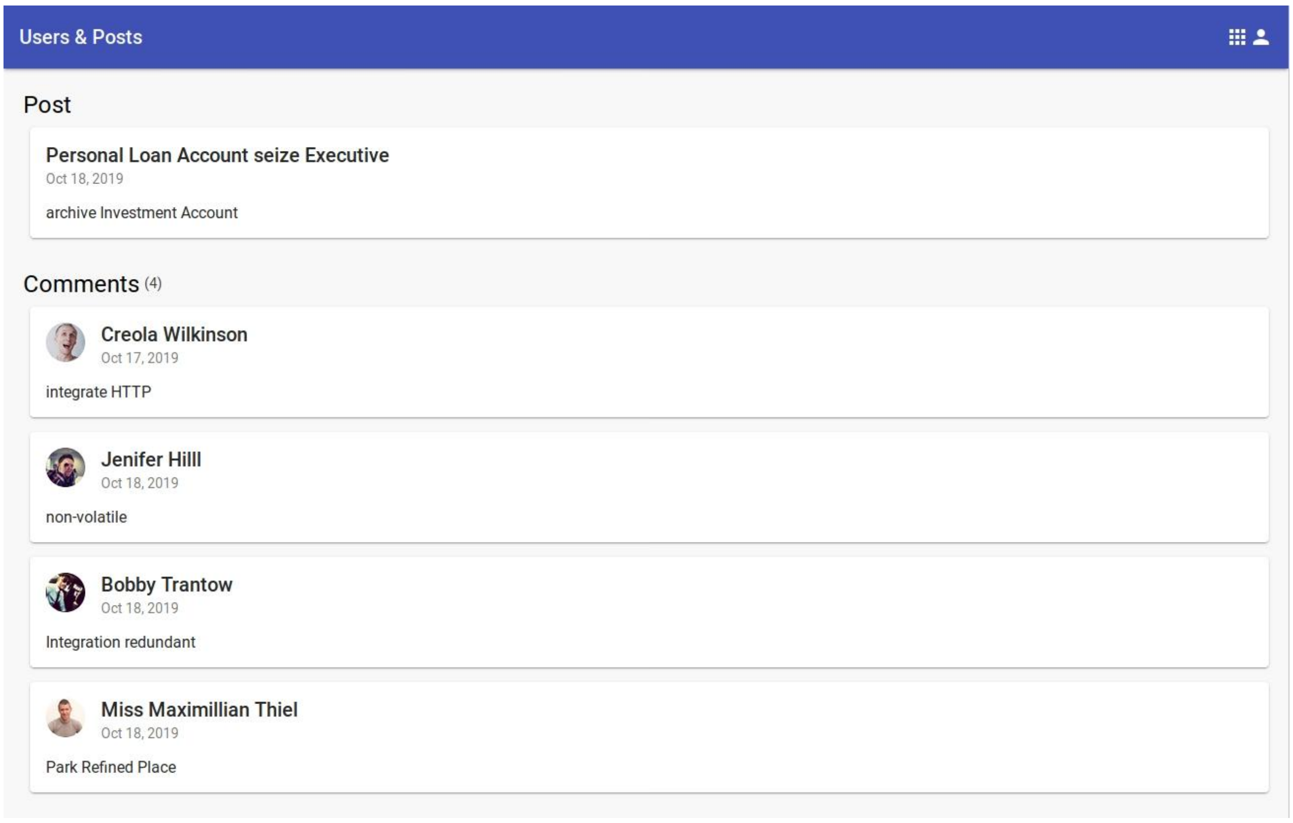
```
{
  id: number,
  postId: number,
  createdAt: Date,
  email: string,
  avatar: string,    // Link to an image
  name: string,
  body: string,
}
```

Links

Card
https://material.angular.io/components/card/overview

List
https://material.angular.io/components/list/overview

Material css classes for fonts
https://material.angular.io/components/button/examples



Post

Personal Loan Account seize Executive
Oct 18, 2019
archive Investment Account

Comments (4)

Creola Wilkinson
Oct 17, 2019
integrate HTTP

Jenifer Hilll
Oct 18, 2019
non-volatile

Bobby Trantow
Oct 18, 2019
Integration redundant

Miss Maximillian Thiel
Oct 18, 2019
Park Refined Place