

Assignment Details

API endpoint for users data

All 10 users profile data is to be downloaded from the following API endpoint:

Method: GET

URL: <https://jsonplaceholder.typicode.com/users>

The schema of the data received in the response is:

```
// Array of 10 users

[

  {

    id,          // The user's id

    username,

    name,

    email,

    phone,

    website,

    address: {

      street, // Address line 1

      suite,  // Address line 2

      city,

      zipcode

    },

    company: {
```

```
    name, // The name of company where the user works
  }
}
]
```

API endpoint for users' avatar pictures

You will be using [Avatars by DiceBear](#). They provide a free HTTP API to create unique avatar images based on any string we send as query parameter. Each string generates a unique image. You will use the `username` to generate a unique avatar for each user.

The URL for the `GET` endpoint is:

```
https://avatars.dicebear.com/v2/avataaars/{{username}}.svg?options[mood][]=happy
```

The `{{username}}` in the URL is the placeholder for the user's username. It should be dynamically replaced by the username received from the users API endpoint. For example, if the username for one of the users is `psamd` then the URL for the avatar for this user will be:

```
https://avatars.dicebear.com/v2/avataaars/psamd.svg?options[mood][]=happy
```

Loading Indicator

Upon opening the app a loading indicator is displayed until the data is fetched from the API and is ready to be displayed. The source code for the loading indicator can be obtained from: <http://tobiasahlin.com/spinkit/>.

Deadline for submission: 2 days

DEMO - <https://react-basic-assignment.psamd.now.sh/>

This assignment is designed for applicants who have learned ReactJS and have minimal hands-on experience in developing react application.

What are we looking for?

With this assignment we would evaluate the following:

- Ability to create new react projects using [Create React App](#)
- Understanding of JSX
- Passing props to components
- Understanding of stateful and stateless Components
- Basic understanding of state management and component lifecycle methods
- Fetching data from an API endpoint
- Conditional rendering
- Working with lists

Instructions

This app is only designed for desktop/laptop and will be tested on Chrome browser.

Bootstrap was used to create the UI design for the demo app. But, applicants are encouraged to use any other CSS library/framework that they are comfortable with.