

Third React JS Challenge

Introduction:

Your task is to build a responsive website that showcases healthy restaurant delivery information. The challenge will consist of four primary pages:

1. *Login Page:*

- Users input their username and password.
- Authenticate users using GraphQL API and Axios.
- Redirect to the homepage on successful login or display an error message on failure.

2. *Register Page:*

- New users input details like username, password, and email.
- Register new users using GraphQL API and Axios.
- Redirect to the login page with a success message on successful registration or display an error message on failure.
- Ensure user input validation (e.g., password length, valid email format).

3. *Homepage:*

- Fetch a list of restaurants from the GraphQL API with Axios.
- Display information for each restaurant.

4. *Restaurant Detail Page:*

- Fetch detailed information for a restaurant based on its ID through Axios.
- Display the restaurant's details, including its top three dishes.

Requirements and Recommendations:

- Use React, HTML5, CSS3, and Typescript.
- Install and utilize the Axios library for API requests.
- Write unit tests.
- Ensure the site is responsive across various screen sizes.
- Prioritize user-friendly interface.

Specifications

FitMe

Name	Type	Example
name	String	"A string"
rating	Number	1
deliveryTime	String	"A string"
isExpensive	Boolean	true
location	String	"A string"
image	String	"A string"
topDishes	Array	[1, "a string"]

Reading Restaurants:

Request

URL: <https://parseapi.back4app.com/classes/FitMe>

Method: GET

Headers:

X-Parse-Application-Id:

IrAPveloMI57TTby5U0S4rFPBrANkAhLUll8jFOh

X-Parse-REST-API-Key:

8aqUBWOjOpIfA6lstntyYsYVkt3RzpVtb8qU5x08

content-type: application/json;

Parameters

A where URL parameter constraining the value for keys. It should be encoded JSON.

Success Response

Status: 200 OK

Body

a JSON object that contains a results field with a JSON array that lists the objects.

Dish

Name	Type	Example
price	Number	1
name	String	"A string"
description	String	"A string"
image	String	"A string"

Reading top dishes:

Request

URL: <https://parseapi.back4app.com/classes/Dish>

Method: GET

Headers

X-Parse-Application-Id:

lrAPveloMI57TTby5U0S4rFPBrANkAhLUll8jFOh

X-Parse-REST-API-Key:

8aqUBWOjOplfA6lstntyYsYVkt3RzpVtb8qU5x08

content-type: application/json;

Parameters

A where URL parameter constraining the value for keys. It should be encoded JSON.

Success Response

Status: 200 OK

Body

a JSON object that contains a results field with a JSON array that lists the objects.

User

User is a subclass of Object, what meanings that has the same properties and methods as a generic object. The difference is that **Parse.User** has some special additions specific to user accounts.

As others objects, users have a flexible schema, the differences are that the username and password fields are required and username and email must be unique per user.

<https://parseapi.back4app.com/user/MyCurrentUserObjectId>

Creating users:

Request

URL: <https://parseapi.back4app.com/users>

Method: POST

Headers

X-Parse-Application-Id:

lrAPveloMI57TTby5U0S4rFPBrANkAhLUll8jFOh

X-Parse-REST-API-Key:

8aqUBWOjOpIfA6lstntyYsYVkt3RzpVtb8qU5x08

X-Parse-Revocable-Session: 1

Content-Type: application/json

Body

A JSON document with the key-value pairs that represent your object's data. It must contain a valid username and password.

Success Response

Status: 201 Created

Headers

Location: <https://parseapi.back4app.com/users/MyNewUserId>

The Location header will contain the endpoint of the newly-created user.

Body

A JSON document with the objectId, createdAt and sessionToken fields of the newly-created user.

Log in users:

Request

URL: <https://parseapi.back4app.com/login>

Method: GET

Headers

X-Parse-Application-Id: lrAPveloMI57TTby5U0S4rFPBrANkAhLUll8jFOh

X-Parse-REST-API-Key:

8aqUBWOjOpIfA6lstntyYsYVkt3RzpVtb8qU5x08

X-Parse-Revocable-Session: 1

Parameters

Pass the username and password in URL parameters. It should be encoded JSON.

Success Response

Status: 200 OK

Body

A JSON document with the user's fields.

➔ You need to treat the errors from the api, you can use modals, alerts and others features to alert the user of the error.

➔ You need to publish you code to github and send the link of the repository to the SM's and the instructors below.

Scrums

Rafaela - rafaela.janeczko@compasso.com.br

Marina - marina.ferreira@compasso.com.br

Natalia - natalia.dartora@compasso.com.br

React.js instructors

Jonatan - jonatan.machado@compasso.com.br

Luis - luis.silva@compasso.com.br

Marcelo - marcelo.orlovski@compasso.com.br

➔ No copies are allowed. The projects that share the same code will not be evaluated.

➔ Create a Readme explaining how to run your project

Prototype:

[FitMe - Online Food Delivery Website](#)