

Study First:

01 - React Playlist: <https://www.youtube.com/playlist?list=PL4cUxeGkcC9gZD-Tvwfod2galSzfRiP9d>

02 - React Hooks: <https://www.youtube.com/watch?v=f687hBjwFcM>

03 - Follow up tutorial <https://react-tutorial.meteor.com/> . Stop tutorial after Step 10

References

- <https://react-tutorial.meteor.com/>
- <https://github.com/ics-software-engineering/meteor-application-template-react>

01 - Clone Template Project and Boilerplate

Generate new project (React Meteor Typescript)

```
meteor create --typescript name-of-my-new-typescript-app
```

Install missing dev packages

```
npm i -D @babel/eslint-parser @babel/preset-react @types/jest @types/meteor
@types/meteor-roles babel-plugin-module-resolver eslint eslint-config-airbnb
eslint-plugin-import eslint-plugin-jsx eslint-plugin-jsx-ally eslint-plugin-
meteor eslint-plugin-react eslint-plugin-react-hooks eslint-utils
```

File Structure

Use the file structure similar to this reference project

<https://github.com/ics-software-engineering/meteor-application-template-react>

02 - Install preferred UI Kit / Design System

Choose and install a design system UI Kit to build the system

- Bootstrap
- Material UI
- Mantine UI
- Chakra UI

03 - Prepare Routes

Routes for the app

Route	Permission	Description
/	Guest	Home - Articles List
/articles/:id	Guest	Article Details
/articles/add	Logged In	Article Add
articles/mine	Logged In	List my articles
/articles/:id/edit	Logged In	Article Edit
/login	Guest	Login
/register	Guest	Register
/not-found	Guest	Page Not found

Notes:

- If user navigates to a wrong route, navigate to not found page
- If a guest user navigates to a loggedIn Protected Route, navigate to login
- On logout, navigate to login
- After successful registration login automatically

04 – Authentication

Register Route

- Collect the data from the user, (Name, Email, Password)
- Add the name data to the `user.profile`
- On successful registration navigate to home
- On failure registration display error message (e.g. email already exists)
- Display loading indicator, and disable form fields while loading

Use **Formik** and **Yup** for form validations

Login Route

- Form (email, password)
- Use formik and yup for form validations
- On success login, navigate to home
- On failure, display error message

Notes

- If user already logged in, navigate to home.
- If user logged in, display navbar on home with [user.profile.name](#) and a logout button

05 - Prepare Collections

Prepare mongo collections and models

Articles

- `_id`
- `title`
- `description`
- `createdOn`
- `modifiedOn`
- `createdById`

Comments

- `_id`
- `text`
- `articleId`
- `createdOn`
- `createdById`

06 - Article Add

Frontend Form

Form fields

Field	Type	Required
title	text	true
description	text	true

Notes

- Use Formik and Yup for frontend validations
- When form is submitted, provide loading indicator on button and disable form fields
- On success, display success message
- On fail, display error message

Backend Meteor Method

- Create a meteor method with the correct name
- use `check` to validate meteor method inputs
- Check the user is logged in, if not logged in throw `Meteor.Error`
- Prepare document to be inserted into the database, append missing field (`createdOn`, `createdById`)
- Insert document into database
- Return the `_id` of the new article to the frontend
- On the react project, navigate to the new article page `articles/:id`

07 - Articles List

Frontend

- Create a list that displays the articles on the home page
- Sort the articles newest first
- Display the article title, `createdOn` Date in a readable format

Backend

- Create a Meteor **Method** that returns the required articles

08 - Articles Details

Frontend

- Display article details (title, description, createdBy Name, Date)
- Display comments section
- Display comment input box (Only if logged in user)
- Sort comments, oldest first
- Display comment date and creator name
- Display a delete comment button. User can only delete his comments.

Backend

- Use meteor **method** to get article details
- Use meteor **publication** to get comments related to article
- Use meteor method to insert a new comment
- Use a meteor method to delete a comment

09 - My Articles

List my articles, and display and edit button next to each article on the list
When edit button is clicked, go to edit article form

10 - Article Edit

User can edit an article
Display initial values for article
On update, update the modifiedOn Date for the article

11 - Filter/Pagination

On the articles list, add pagination and search box

Pagination

- Display only 10 items per page
- Display the total count of articles in the system
- Show a button for each page on the bottom of the list, and click on the page button to navigate to the page
- When navigating to a new page, update the url with the correct query params `?page=2`
- When opening a url that has a query param e.g `?page=3` load the page to page 3 directly

Filter

- Create a search input box on the top of the list
- Create a button with label "Search"
- User can enter some text in the text input box and press search
- User should get results of articles that has this text as part of its title or description
- Reset pagination after clicking search (Go to first page)
- Add searchText query to the url query params : (e.g. ?search=tourism)
- If the user pastes a url that has the query param, the initial value of the search box should have the search text and the results should be correct

12 - React Query

Install react query to optimize meteor methods for fetching data
Refactor all meteor methods for fetching data to use react-query

13 - Meteor Grapher

Installation

<https://github.com/cult-of-coders/grapher>

Documentation

<https://cult-of-coders.github.io/grapher/>

Links

- Create links between articles, comments and users
- Refactor meteor methods to use `createQuery` instead of traditional `find` methods
- Return the articles with user name using create Query

Reducers

- Create a reducer on the articles collection that returns the comment count.
- Display the comment count on the articles list on home

14 - Install Simpl Schema

Use simpl schema to define collections

<https://github.com/Meteor-Community-Packages/meteor-collection2>

Attach the correct schemas

Use the default value concept to insert the createdOn, modifiedOn values automatically