Khazana by Srikant Kumar Kalaputapu

MAKE SURE TO FREAKING TAKE DOWN HEROKU SERVER

# Khazana

"CookingForMillenials" by Srikant Kumar Kalaputapu

Redux form saves info between users loggin in and out. maybe not store in redux for create recipe

//or just clear in action

Alternative Name:

Cooking for Millenials

Realized that i should only add ingredients if added with steps. otherwise its hard to link up the

ingredient id to the step\_ingredient

Time length: 2 weeks.

THOUGHTS:

Big mistake:

User-perspective vs Tech-perspective

Original Problem im trying to solve:

users: easy to create recipe

BUT hard to read on mobile when in the kitchen.

The real problem i was trying to solve was the above. But instead i made it harder for users

to create recipes.

Mistakes:

Freaking forgot about normal react and lifecycle methods when trying to make the redux form

React will see fields update on addition of ingredients from the previous page (page2 -> page3)

So

I kept refactoring thinking ahead of time for the future. Add images. How is the recipe going to look like? Now i have to change the way i serialize the data. The skateboard is the most important thing. Thinking ahead is a great thing but with the time limit its useless overthinking.

This is a common theme in learning. Try your hardest and focus on the work at hand not anything else.

This comes with practice.

The point of this project was to learn redux and learn how to manage my time. I learned that things take longer than planned for good work. To just do work its no big deal.

Learning lessons:

I tried to do all of the form front end and then all of the back end.

Build vertically. I got cocky and thought i was on pace but then these issues came up.

For skateboard just do normal form with create and show.

Stop trying to be fancy with wizard form and worry about mobile friendlyness

First get redux to work with the form and add extra fields. etc.

#1 priority easy viewing on mobile!!!

Show page > create page

Question:

Serialize the recipes with sideloading so that everything loads initially

or fetch based on recipe stubs ?

Less fetches the better...

So fetch recipe and sideload recipe-steps

grab step-ingredients and ingredients on show page load ?

Question:

Could filter myRecipes from all recipes in the mapStateToProps instead of maintaining in redux state. BUT:

custom reducer and action is better because its easier to understand/read logically

Confuse people if they saw allRecipes being used and then they would need to figure out why

Im only getting the user's specific recipes -> easier to communicate!

-> Load on login ! not on App? but then error because of check for loaded allrecipes.

-> Maybe have a delay and wait?

After userInfo loaded -> call dispatch on that too.

so chain dispatch off of promise -> BUT WHAT ABOUT REFRESH?

OUTCOME:

Redux is for caching the state. Dont need to get data based off other fetched data

otherwise the dispatch for allRecipes needs to do myRecipes and vice versa

Knots & unnecessarily dependent.

-> custom route fetch myRecipes

-> Tie to profile page/RecipeList probably recipeList

OR: return the users recipes within the original fetch since we have to fetch to user anyway

-> best solution. When we get user we always get recipes

QUESTION:

Did too many db backend tables and worried about sideloading instead of one at a time. Now i realize i do need redux and i have no backend to support the front end.

POSITIVE OUTCOME:

Realized i really do need redux since i dont want to sideload everything all at once due to multiple components having partial information - aka lots of props otherwise.

Rushing is bad but we learn from it!

Tests for code

-> in general, immediate payoff

-> JUNIT or general

-> create-react-app has REST test framework

-> ENZYME -> snapshot testing enzyme makes it for you! a snapshot

Cooking For Millenials by Srikant Kumar Kalaputapu

Focus: React with slideable library (mobile first) and build basics for table

Project end date for features - December 5th, presentations are December 7th, 2018

COMPONENT HIERARCHY:

App

- Navbar

-Search

- MainContainer

- Login

- Profile

-MyRecipes

-RecipeList

-RecipeItem

-RecipeForm

-Ingredients

-Directions

-Image - optional or part of directions

- RecipeCatalog

-RecipeCard

-RecipeShow

SCHEMA

Users

Recipes HAS MANY R-S

Recipe-Steps HAS MANY S-I

-step number

-instructions

-image

Steps-Ingredients (joiner)

-amount

Ingredients HAS MANY S-I

-type

-name

Cooking utensils ???

ROUTES

resources :recipes

resources :ingredients, only: [:create,:show, :index]

resources :recipe\_steps, only: [:create, :show, :index]

resources :step\_ingredients, only: [:create, :show, :index]

resources :users, only: [:create]

post '/login', to: 'auth#create'

get '/profile', to: 'users#profile'

Ingredients need to be sideloaded with step\_ingredients?

Status

ToDo:

Signup page

Components for Recipe

One at a time for the Form

Index for a user recipes:

RecipeList

RecipeItem

SideLoad Serializing for Recipe

Cooking utensils??

Completed:

User

Auth

Login

Recipe

Recipe\_Step

Step\_Ingredient

Ingredient

Seed data

Added data validation

Serializers for tables

Sideload the data for recipe

In-Progress:

Add Redux - earlier the better

Create routes for each of the models

Index to get all

Or show for specifics ?

Routes

Stretch features:

Image upload per step - CarrierWave https://medium.com/@jakeberman\_97785/image-uploading-with-a-react-frontend-64ee2cc9b33d

Email 2 factor auth?

Search feature

Gamify for learning ?

Standardized form language NLP for recipes?

Redux? - multiple redux stuff for state - reused

Expand button for the ingredients

Notes:

Build vertical - db then front end

Build Feature by Feature since we have to deal with people.

Dont really need redux except maybe for search feature

Dont need full CRUD

Social media sharing

Slideable and mobile first

Issues:

No sideloading for recipes since:

Recipe info for just stub/cards

Recipe Show Pages need:

Recipe, Recipe-Steps, Step-Ingredients and Ingredients

Recipe-step has many ingredients through

Recipe-step, step-ingredient and ingredient need to be sideloaded together

Recipe-step is the main db

Can either do:

Recipe step has many S-I and has I many through S-I

Or: chain has manys - not working huh

Heroku - cant use sqlite since sqlite is a file based db. (good for small scale app not for big scale app)