Propsplit Not so Quick Audit

Routine audit propsplit: 21 March 2023

[**Data Model 1**](#_ibfzt26w7ocv)

[1. Id’s should have been strings and we should have duplicated MEDIUM 1](#_oj0rqw9oxdx0)

[2. Useless index.js ? LOW 2](#_vv6yd7shynlh)

[**The backend 2**](#_ek4trf35llh0)

[3. Naming sometimes difficult to understand MEDIUM 2](#_u4uidhgqi2ng)

[4. Too many ENUMS MEDIUM 2](#_38bfy7iw5d2p)

[5. Why do we have a file called utils/errors if we’re using sentry? MEDIUM 2](#_qb0k96g04aku)

[6. Some of the utils are inside of index.js and some inside of the folder. LOW 2](#_dnj838b9tlk5)

[7. Some cleaning to do on props, args LOW 3](#_mdzz3ede3q33)

[8. No post search anywhere HIGH 3](#_9vxqd36buhh5)

[9. Non conventional returns from controllers HIGH 3](#_dhp8qpiaxeo)

[10. Two file uploader controllers, one called /upload and one called /file HIGH 3](#_hn1podwgb0hd)

[**The web app 3**](#_kemhhxje2l8g)

[11. Maybe you want to replace redux with zustand MEDIUM 3](#_rmxos8eir71u)

[12. Messy Assets LOW 4](#_u9hwl45dgkps)

[13. Usage of hooks MEDIUM 4](#_hmpglnlql3ap)

[14. Reduce level of abstraction. HIGH 4](#_mo122f5knmzc)

[15. Too many conditions in the dom HIGH 5](#_1tclkgtr3ga0)

[16. Naming issues on scenes LOW 6](#_eu07e6r8g5u0)

[17. Usage of formdata. HIGH 6](#_8l80ilmdolre)

[18. Dead big chunks of code LOW 6](#_pr1vbmh2nxy)

[**The UX Feedbacks 7**](#_5f2303ocuw79)

[19. Drawer Issues HIGH 7](#_f2jteaftsy71)

# Data Model

### Id’s should have been strings and we should have duplicated MEDIUM

Not enough data duplication and id values are saved as objectId. As per Le Stud convention we save them as strings and we duplicate a little bit of what we might need. Here for example the following

userId: { type: mongoose.Schema.Types.ObjectId, ref: 'user', required: true },

should become.

userId: { type: String, ref: 'user', required: true },

userName: { type: String, default: '' },

userAvatar: { type: String, default: '' },

Twitter had their data like this for their first few years. We’re just trying to keep the company alive and flexible at this point, so this is the way to go. Premature Optimisation.

replies: [

{

replierId: { type: mongoose.Schema.Types.ObjectId, ref: 'user', required: true },

replyMessage: { type: String, required: true },

replyAt: { type: Date, default: Date.now },

},

Same with replies here, wouldn’t it be nice to have the avatar and the name without complicated controllers? I’m sure I’ll write a bit more when I get to the controller.

### Useless index.js ? LOW

I find it weird, it’s working without it in our other projects. To check.

<https://github.com/selego/propsplit/blob/staging/api/src/models/index.js>

# The backend

### Naming sometimes difficult to understand MEDIUM

const offersToPerformAndNot = async (auction, offers) => {

What is offersToPerformAndNot, even if it has a longer name I’m not sure what it does. Maybe it’s because I don’t fully understand the business?

### Too many ENUMS MEDIUM

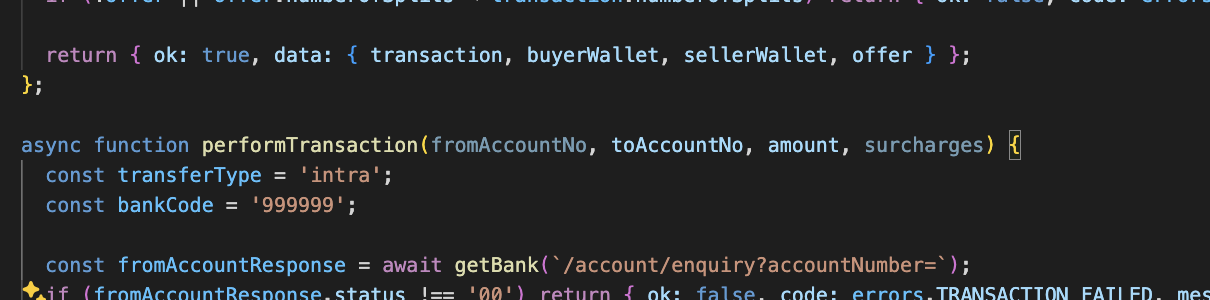
By itself it’s not a red flag but it probably means there’s a lot of things to click before you get here, it’s a bit of hidden information. To see how many times it’s being used but unless used in at least 3 files, I wouldn’t globalize it.

### Why do we have a file called utils/errors if we’re using sentry? MEDIUM

### Some of the utils are inside of index.js and some inside of the folder. LOW

I find this inconsistent.

### Some cleaning to do on props, args LOW



### No post search anywhere HIGH

You’ve used the good old get. We find it cleaner with the post search. See more in the whitepaper. Le Stud Convention.

### Non conventional returns from controllers HIGH

return res.status(200).send({ ok: true, data: { wallet, transfer: transferResponse.data } });

Usually we only return one type of data in the data and do not have in nested like this.

### Two file uploader controllers, one called /upload and one called /file HIGH

And i have the feeling the wrong one is being used the most, please use /file like in our other projects.

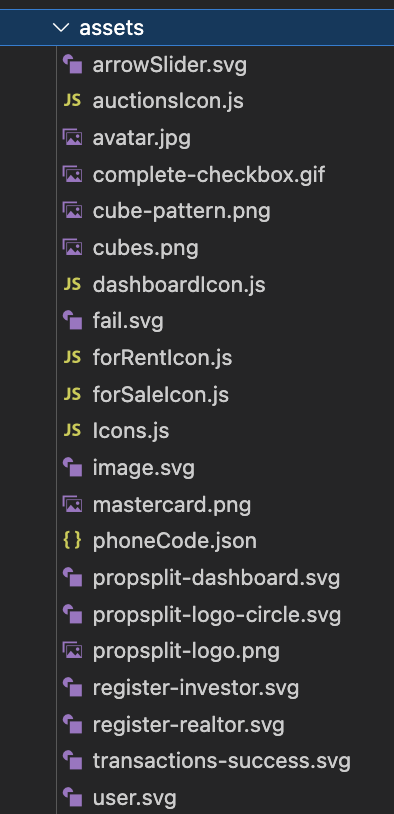
# 

# The web app

### Maybe you want to replace redux with zustand MEDIUM

It will un-complify it a bit and we’ve already decided that we’re all moving to zustand.

### Messy Assets LOW



### Usage of hooks MEDIUM

There’s only three hooks and they’re simple but I think we really want to stay away from them unless necessary so I have to point it out to open a conversation.

### Reduce level of abstraction. HIGH

You could just do something in the lines of   
Step 1 && <Component/>  
Step 2 && <Component/>  
  
Will be easier to read in my opinion and you might be able to save yourself from some conditions in the dom (see next point)

const stepsData = [

{

id: 1,

name: "Select listing type",

component: SelectListingType,

},

{

id: 2,

name: "Listing details",

component: ListingDetails,

},

{

id: 3,

name: "Media upload",

component: MediaUpload,

},

{

id: 4,

name: "Summary",

component: Summary,

},

];

### 

### Too many conditions in the dom HIGH

Code looks complex, whenever there’s too many conditions, there’s usually a place for a component.

<span className="text-sm text-app-cyan-900">{step.name}</span>

</div>

</>

) : step.id === currentStep ? (

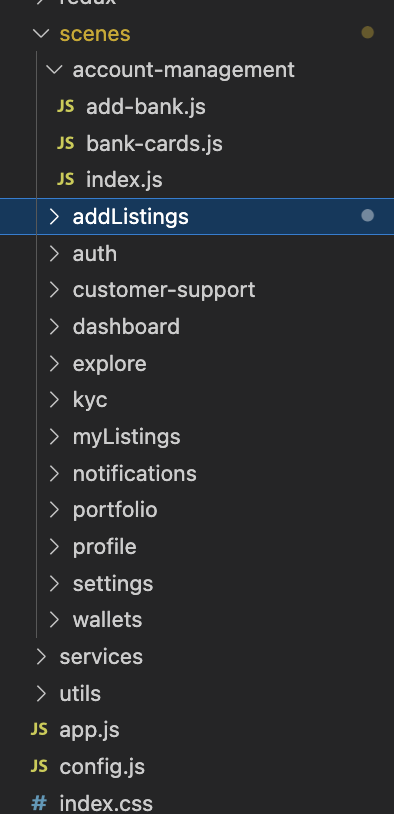
<>

{step.id !== stepsData.length ? <div className="absolute left-3

top-4 -ml-px mt-0.5 h-full w-px bg-gray-300" aria-hidden="true" /> : null}

<div className="flex items-center gap-3" aria-current="step">

### Naming issues on scenes LOW



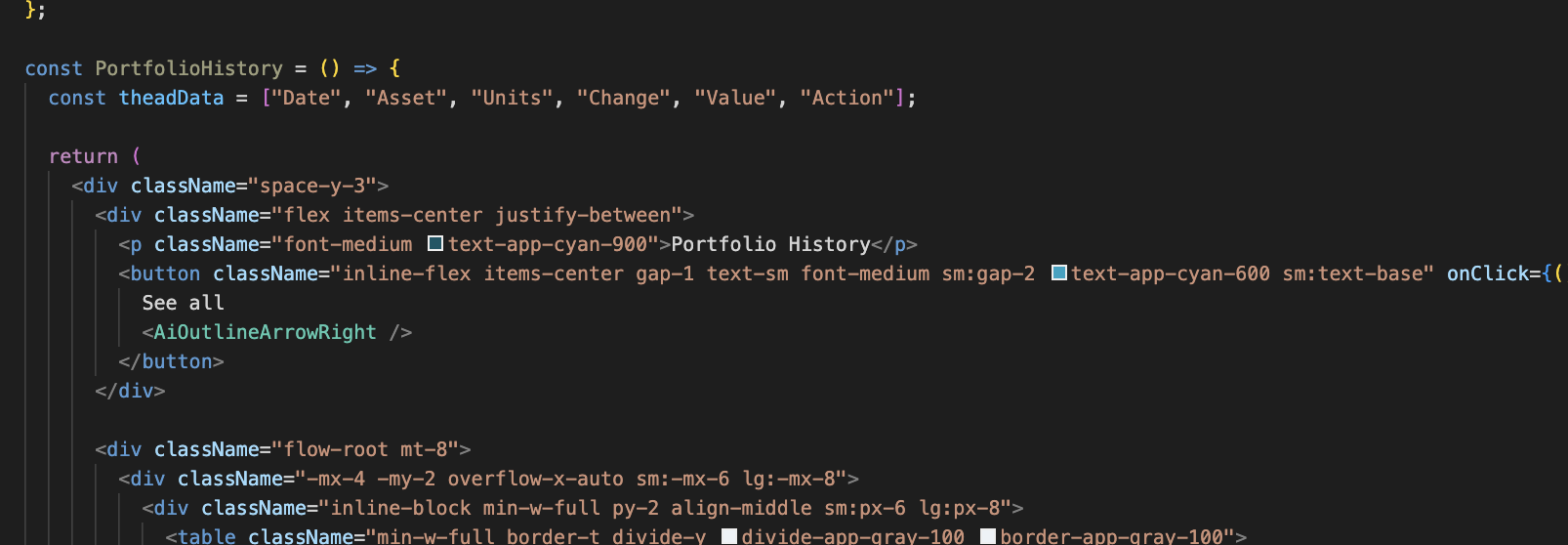
Mix of camelCase snake-case

### Usage of formdata. HIGH

const uploadedImages = await api.postFormData("/upload", formData);

### Dead big chunks of code LOW

For example portolio History



# The UX Feedbacks

### Drawer Issues HIGH



Conventionally in Le Stud we do not put user profile, settings and logging out in the left side menu. Instead we put it on a dropdown on the top right profile picture. Like below:   
  
