# Data Structure

The app will use JSON objects to establish the communication between SPA and API. These object will be, mainly, a single cash flow and a list of cash flows.

## Single cash flow result

A single cash flow will be a cash movement, related to a user and will contain information such as the title, the classification, date and amount. The classification should be an id as it will be selected from a dropdown for the user so the list can be filtered by these classifiers without introducing mistakes.

An example for this kind of object could be the following:

{

“title”: “New keyboard”,

“classification”: 2,

“date”: “2012-04-23T18:25:43.511Z”

“amount”: 36

}

To bear in mind, the date variable should follow the JavaScript’s Date format, as the language specifies.

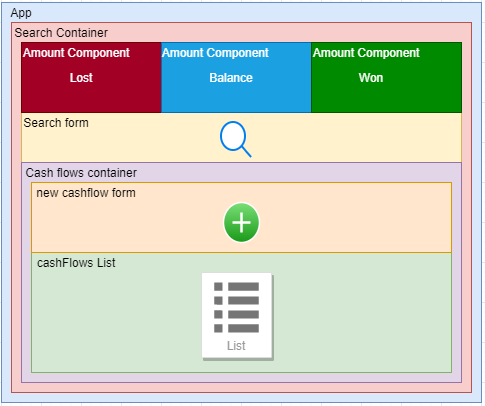
## Cash Flow List

The cash flow list will contain all the cash flows from an user, filtered or not by one or more variables.

# Components

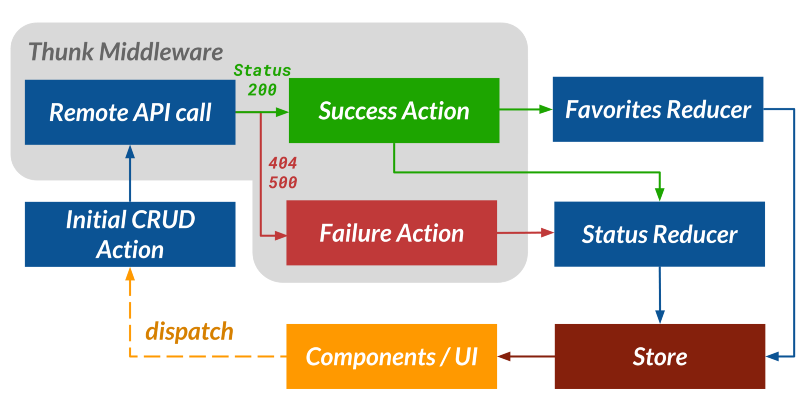
As components are the most important part in React, is necessary to determine which ones are going to interact with Redux and which ones are for presentation. In that way, the reuse of each component and therefore, the structure of the application will be highly improved.

The component hierarchy should look something like this:



The Search Container will be rendered with the cash flows, amounts and the search form. All these components will need to communicate each other as the list of cash flows depends on the search made by the user, that will provide and action to the Redux store, the amounts also depend on the list. Also, a new item in the list should alter the list.

# Reducer Brief Reference



1: https://www.freecodecamp.org/news/how-i-architected-a-single-page-react-application-3ebd90f59087/

When an action is dispatched, it should make a call to the API and wait for the result, wich has two possible results: success or failure.

In case the result is successful, the result is stored, and the status is updated, so the UI can display it.

Otherwise, if a failure is returned, then the status is updated with a failure and the UI will display the correspondent error message.

# References

How I architected a single-page React application, 13 April 2019, by Gooi Ying Chyi <https://www.freecodecamp.org/news/how-i-architected-a-single-page-react-application-3ebd90f59087/>

**react-clean-architecture** project, latest commit on 18 Jul 2019, by eduardomoroni <https://github.com/eduardomoroni/react-clean-architecture>

On doing ‘Clean Architecture’ in React applications (Parts 1 & 2), 6 October 2019, by Janith Leanage <https://medium.com/@janithl/on-doing-clean-architecture-in-react-applications-666d568362e> <https://janithl.github.io/2019/10/react-clean-architecture-part-2/>