Flux

lunes, 31 de octubre de 2022 17:19

Flux

→ Flux is a design patern to keep data flowing in one direction.
→ Flux is an alternative to MVC model

In Flux, application state data is managed outside of React components in stores.
↳ hold and change data: stores are the only thing that update a view.

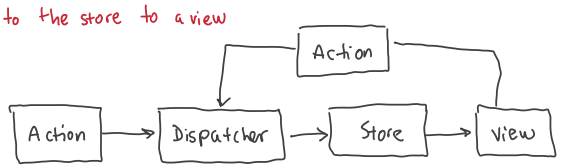
If a user interacts with a web page (click, submit, etc) an action would be created to represent the user's request.
↳ An action provides the instructions and data required to make a change
↳ Actions are dispatched using a component called the dispatcher.

The dispatcher queues actions and dispatch them to the correct store.

Once a store receives an action, it uses it as instructions to modify state and thus update the view.

Data flows in one direction:

action to a dispatcher to the store to a view



→ Actions can be dispatched from a view, or they can come from a web server, for example

→ Every change requires an action.

→ Every action provides the instructions to make the change

→ A store is the only thing that can make a change.

→ Stores update the data.

→ Views update the components whose data changed.

* Actions
→ Actions provide the instructions and data that store uses to modify the state.

→ Action creators are functions that abstract all details to build an action.

→ Actions are objects that at minimum contain a type field.
↳ the action type is an uppercase string that describes the action.
↳ Actions may also contain any data required by the store

action object

```
dispatcher.handleAction({ type: 'TICK' })
```

↳ the dispatcher's handleAction method is invoked, which dispatches the action object

* Dispatcher

→ the dispatcher takes the action and sends it to the store.

→ when an action is dispatched it is handled in the order that is was received and sent to the store(s).

* Stores

→ Objects that hold the applications state data.

→ Current state can be obtained using properties

→ Everything a store needs to change state is provided in the action.

→ A store will handle an action by type and change data accordingly.

→ Once data is change, the store notifies the views that subscribed to it to know their data changed.