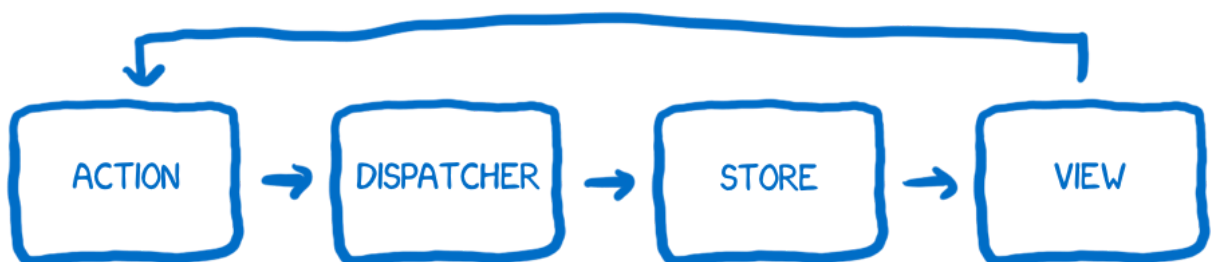


Use of Flux Architecture in SUSI AI

What is Flux ?

Flux is micro-architecture for software application development. Flux is the application architecture that Facebook uses for building client-side web applications. It complements React's composable view components by utilizing a unidirectional data flow.

Flux Overview



As can be seen from the above diagram, Flux has four main components:

Component	Description
Actions	Helper methods that pass data to the Dispatcher.
Dispatcher	Receives these Actions and broadcast payloads to registered callbacks.
Stores	The Stores that are registered to the Dispatcher will update the Views accordingly.
Views	Views are actually React Components that grab the state from the stores and then pass it down to the child components.

How these 4 components work together ?

1. The Setup : Application Initialization :-

- a. Stores let the registered Dispatcher know that they want to be notified whenever an action comes in.
- b. Then the controller views ask the stores for the latest state.
- c. When the stores give the state to the controller views, they pass that state along to their child views to render.
- d. The controller views also ask the stores to keep them notified when state changes so that they can re-render accordingly.

2. The Data Flow :-

Once the setup is done, the application is ready to accept user input. So let us trigger an action by having the user make a change.

- a. The view tells the action creator to prepare an action.
- b. The action creator formats the action and sends it off to the Dispatcher.
- c. The Dispatcher dispatches the action off to all of the stores in sequence. Then the store decides whether it cares about this action or not, and changes the state accordingly.
- d. Once it's done changing state, the store lets its subscribed view controllers know.
- e. Those view controllers will then ask the store to give them the updated state.
- f. After the store gives it the state, the view controller will tell its child views to re-render based on the new state.

So this is how Flux works. Now let us see an example of Flux being used in SUSI Web Chat.

Use of Flux Architecture to Switch Between Themes in SUSI Web Chat

Different Flux components used in SUSI Web Chat -

1. Actions - [Settings.actions.js](#)
2. Dispatcher - [ChatAppDispatcher.js](#)

3. Store - [UserPreferencesStore.js](#)

4. View - [MessageSection.react.js](#)

Let us now see in detail how we are able to switch themes :-

In the MessageSection.react.js file, we have the following functions to handle theme change :

```
handleThemeChanger = () => {  
  //code  
  this.applyLightTheme();  
}
```

```
applyLightTheme = () =>{  
  //code  
  Actions.settingsChanged(settingsChanged);  
}
```

Hence, the view tells the action creator to prepare an action.

Now, the settingsChanged() function in Settings.actions.js file is called as follows :

```
export function settingsChanged(settings) {  
  ChatAppDispatcher.dispatch({  
    type: ActionTypes.SETTINGS_CHANGED,  
    settings  
  });  
  //code  
}
```

Hence, the process of invoking the callbacks is initiated through the dispatch() method, which takes a data payload object as its sole argument.

Now, as the `UserPreferencesStore` is registered with the `ChatAppDispatcher`, it receives the actions dispatched by the `Dispatcher`. It checks for the type of action, and with the help of switch cases, executes the code for the corresponding action type accordingly. In this case, the states inside the `Store` are updated with the new state which the User wants to switch to.

```
UserPreferencesStore.dispatchToken = ChatAppDispatcher.register(action => {
  switch (action.type) {
    //code
    case ActionTypes.SETTINGS_CHANGED: {
      let settings = action.settings;
      //code
      UserPreferencesStore.emitChange();
      break;
    }
    //code
  }
});
```

This is how Flux Architecture is facilitating the switching of themes in SUSI Web Chat.

Resources

- GitHub Repository - <https://github.com/facebook/flux>
- Tutorials for Flux -
 - <https://scotch.io/tutorials/getting-to-know-flux-the-react-js-architecture>
 - <https://facebook.github.io/flux/>
- Relevant Blogs -
 - <https://blog.fossasia.org/use-of-flux-architecture-to-switch-between-themes-in-susi-web-chat/>
 - <https://blog.andrewray.me/flux-for-stupid-people/>