Front-End Development for the MPC in the Cloud Sprint-3

QingXing Li, Yicun Hou, Haoyu Xu
Mentors: Joseph Caiani, Máirín Duffy

Project Overview

- The main project is the front end of ChRIS(ChRIS Research Integration System) project
- Goal: Design a front-end interface using React & Redux. Receive the input from the user and send the result from MOC to the users.
- Sprint1 : Set up the ChRIS Store backend in Docker and rewrited ChRIS Store UI with Redux
- Sprint2: Write the declaration file of all the APIs by using Typescript, and installed the backend of the Chris UI

Sprint 3- What we did

- Debug the declaration file of all the APIs
- Test and Debug Actions & Reducer of all the APIs by using Jest
  - Plugin
  - Feed
  - Users
  - UI
  - Message
- Get started the containerize of the Chris Store by using Openshift

JEST

- Jest is a delightful JavaScript Testing Framework with a focus on simplicity.
- JEST can work with projects using: TypeScript, Node, React, Angular, Vue and more!

How Jest works

`describe` breaks your test suite into components. You might have a describe for each function in your class, each module of your plugin, or each user-facing piece of functionality.

`it` is where you perform individual tests. You should be able to describe each test like a little sentence, such as "it calculates the area when the radius is set".

`Expect` gives you access to a number of "matchers" that let you validate different things.

Matchers: let you test values in different ways.

Example

```
describe(“example”, => {
    it('object assignment', () => {
        const data = {one: 1};
        data['two'] = 2;
        expect(data).
            toEqual({one: 1, two: 2});
    });
})
```

Action Test

Async Actions:

It’s a set of actions from request for data to dispatch the state

Example:

Click the ”GetAllFeed” icon on web, and it will send the request first, and there are two possible feedbacks: Success or Failure

```
describe("display message", () => {
  it("return action of type display message", () => {
    const testMessage: IMessage = {
      message: "hello",
      type: "success",
      displayType: "modal",
    }

    const expectedAction = {
      type: messageActionTypes.DISPLAY_CONFIRMATION,
      payload: testMessage
    };
    expect(handleMessage(testMessage)).toEqual(expectedAction);
  });
});
```

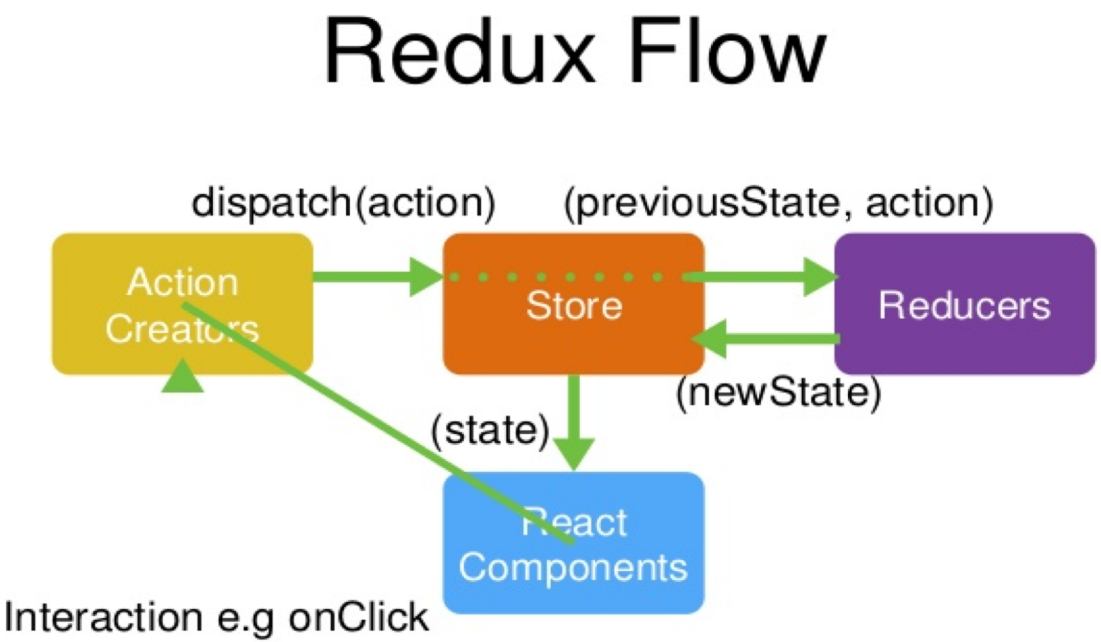
```
// type them properly as well -> For more info: https://github.com/piotrwitek/typesafe-actions
export const getAllFeedsRequest = (name?: string) => action(FeedActionTypes.GET_ALL_FEEDS, name);
export const getAllFeedsSuccess = (feeds: IFeedItem[]) => action(FeedActionTypes.GET_ALL_FEEDS_SUCCESS, feeds);
```

Question form last sprint:
How can Redux updates state?

```
// ***** NOTE: Working *****
const reducer: Reducer<IFeedState> = (state = initialState, action) => {
  switch (action.type) {
    case FeedActionTypes.GET_ALL_FEEDS_SUCCESS: {
     // return { ...state, feeds: action.payload.collection.items }; // Using the chrisApi
      return { ...state, feeds: action.payload.data.results }; // Using the chrisApi
    }
    case FeedActionTypes.GET_FEED_DETAILS_SUCCESS: {
     // return { ...state, details: action.payload.collection.items }; // Using the api
      return { ...state, details: action.payload };
    }
```

Reducer Test

```
it("FetchToken should return ",()=>{
    expect(userReducer(initialState,{
        type:UserActionTypes.FETCH_TOKEN,
        payload:UserState
    })).toEqual(
        {
            username: "string",
            token:  null,
            isRememberMe: false,
            isLoggedIn: false
        }
    )
});
```



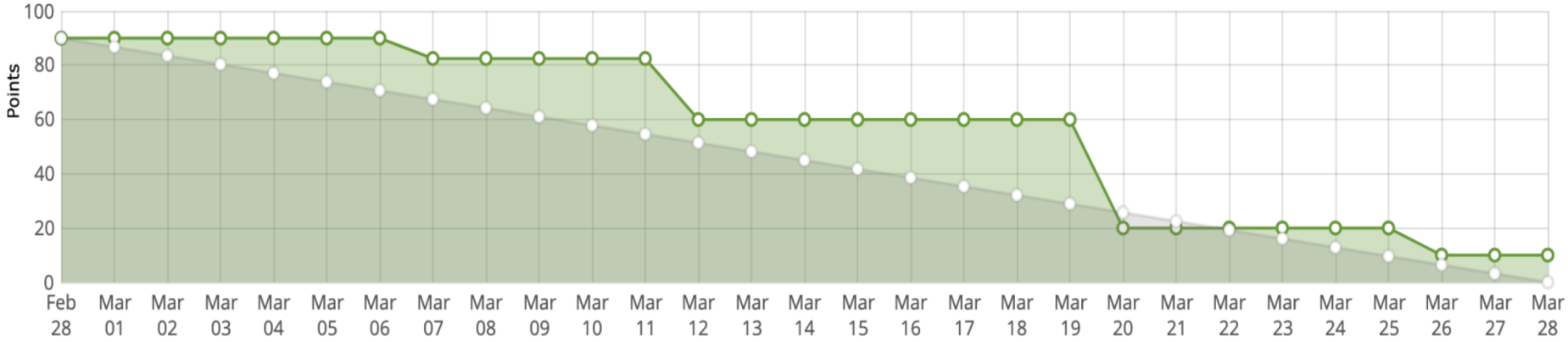
Demo

- TypeScript’s type declaration for the the project
- Action testing
- Reducer testing

Burn Down Chart

FRONT-END-DEVELOPMENT-FOR-MULTI-PAR... SPRINT 3 28 FEB 2019-28 MAR 2019

33% 90 total points 30 completed points 2 open tasks 9 closed tasks 0 iocaine doses



Sprint 4- Next to do

- Research how to containerize the Chris Store
- Deploy the Chris Store to MOC using OpenShift
- Do some scale tests about Chris Store

Questions?