Actions are the only source of information for the store as per Redux official documentation. It carries a payload of information from your application to store.

As discussed earlier, actions are plain JavaScript object that must have a type attribute to indicate the type of action performed. It tells us what had happened. Types should be defined as string constants in your application as given below −

const ITEMS\_REQUEST = 'ITEMS\_REQUEST';

Apart from this type attribute, the structure of an action object is totally up to the developer. It is recommended to keep your action object as light as possible and pass only the necessary information.

To cause any change in the store, you need to dispatch an action first by using store.dispatch() function. The action object is as follows −

{ type: GET\_ORDER\_STATUS , payload: {orderId,userId } }

{ type: GET\_WISHLIST\_ITEMS, payload: userId }

Actions Creators

Action creators are the functions that encapsulate the process of creation of an action object. These functions simply return a plain Js object which is an action. It promotes writing clean code and helps to achieve reusability.

Let us learn about action creator which lets you dispatch an action, **‘ITEMS\_REQUEST’** that requests for the product items list data from the server. Meanwhile, the **isLoading** state is made true in the reducer in ‘ITEMS\_REQUEST’ action type to indicate that items are loading, and data is still not received from the server.

Initially, the isLoading state was false in the **initialState** object assuming nothing is loading. When data is received at browser, isLoading state will be returned as false in ‘ITEMS\_REQUEST\_SUCCESS’ action type in the corresponding reducer. This state can be used as a prop in react components to display loader/message on your page while the request for data is on. The action creator is as follows −

const ITEMS\_REQUEST = ‘ITEMS\_REQUEST’ ;

const ITEMS\_REQUEST\_SUCCESS = ‘ITEMS\_REQUEST\_SUCCESS’ ;

export function itemsRequest(bool,startIndex,endIndex) {

let payload = {

isLoading: bool,

startIndex,

endIndex

}

return {

type: ITEMS\_REQUEST,

payload

}

}

export function itemsRequestSuccess(bool) {

return {

type: ITEMS\_REQUEST\_SUCCESS,

isLoading: bool,

}

}

To invoke a dispatch function, you need to pass action as an argument to dispatch function.

dispatch(itemsRequest(true,1, 20));

dispatch(itemsRequestSuccess(false));

You can dispatch an action by directly using store.dispatch(). However, it is more likely that you access it with react-Redux helper method called **connect()**. You can also use **bindActionCreators()** method to bind many action creators with dispatch function.