Actions are payloads of information that send data from your application to your store. They are the only source of information for the store. You send them to the store using [store.dispatch()](https://redux.js.org/api/store" \l "dispatch).

Ex: const ADD\_TODO = 'ADD\_TODO'

import { ADD\_TODO, REMOVE\_TODO } from '../actionTypes'

## Action Creators

Action creators are exactly that—functions that create actions.

Action creators simply return a function

function addTodo(text) {

return {

type: ADD\_TODO,

text

}

}

# Reducers

Reducers specify how the application's state changes in response to [actions](https://redux.js.org/basics/actions) sent to the store.

**actions** only describe what happened, but don't describe how the application's state changes.

function todoApp(state = initialState, action) {

// For now, don't handle any actions

// and just return the state given to us.

return state

}

ex:

function todoApp(state = initialState, action) {

switch (action.type) {

case SET\_VISIBILITY\_FILTER:

return Object.assign({}, state, {

visibilityFilter: action.filter

})

default:

return state

}

}

Object.assign() method is used to copy the values of all enumerable own properties from one or more source objects to a target object. It will return the target object.

# Store

we defined the [actions](https://redux.js.org/basics/actions) that represent the facts about “what happened” and the [reducers](https://redux.js.org/basics/reducers) that update the state according to those actions.

The Store is the object that brings them together. The store has the following responsibilities:

* Holds application state;
* Allows access to state via [getState()](https://redux.js.org/api/store" \l "getState);
* Allows state to be updated via [dispatch(action)](https://redux.js.org/api/store" \l "dispatch);
* Registers listeners via [subscribe(listener)](https://redux.js.org/api/store" \l "subscribe);
* Handles unregistering of listeners via the function returned by [subscribe(listener)](https://redux.js.org/api/store" \l "subscribe).

MiddleWare:

1. We need it for Logging

2. Crash Reporting