



Redux Questions

▼ What is Redux?

- Redux is a library for managing application state
- Redux is a solution for global state management, allowing you to pass data without the need to pass props through every level (aka props drilling)

▼ What are the main constituents of Redux?

Actions

- **Actions** are plain Javascript objects with `type` property to indicate what type of action is being performed
- Actions are dispatched to the central store using `store.dispatch()`
- Actions are usually created by Action Creator functions, which given some input generates the corresponding action.

Reducers

- **Reducers** are functions that take an Action and a current state and returns the resulting state
- Reducers must be pure functions and have no side effects

Store

- The **Store** is the central object that maintains and updates the application state

▼ How does Redux work?

- Redux revolves around strict unidirectional data flow
- **First:** an *Action* is dispatched to the store via `store.dispatch(action)`
- **Second:** Redux determine the resultant state by calling a *Reducer* function
- **Third:** The root reducer combines the output of multiple reducers into a single state tree
- **Fourth:** The *Store* saves the new state tree and notifies listeners that are registered

▼ What are reducers in redux?

The reducer is a pure function that takes the previous state and an action, and returns the next state.

```
(previousState, action) => newState
```

It's very important that the reducer stays *pure*. Things you should never do inside a reducer:

- Mutate its arguments;
- Perform side effects like API calls and routing transitions;
- Call non-pure functions, e.g. `Date.now()` or `Math.random()`

▼ What is Redux Thunk?

Redux Thunk middleware allows you to write action creators that return a function instead of an action. The thunk can be used to delay the dispatch of an action, or to dispatch only if a certain condition is met. The inner function receives the store methods `dispatch` and `getState()` as parameters.

▼ What is the difference between `mapStateToProps()` and `mapDispatchToProps()`?

`mapStateToProps()` is a utility which helps your component get updated state (which is updated by some other components):

```
const mapStateToProps = (state) => {
  return {
    todos: state.visibilityFilter,
  };
};
```

`mapDispatchToProps()` is a utility which will help your component to fire an action event (dispatching action which may cause change of application state):

```
const mapDispatchToProps = {
  setLinkCounter,
  setDraftDataAnalysis,
};
```