

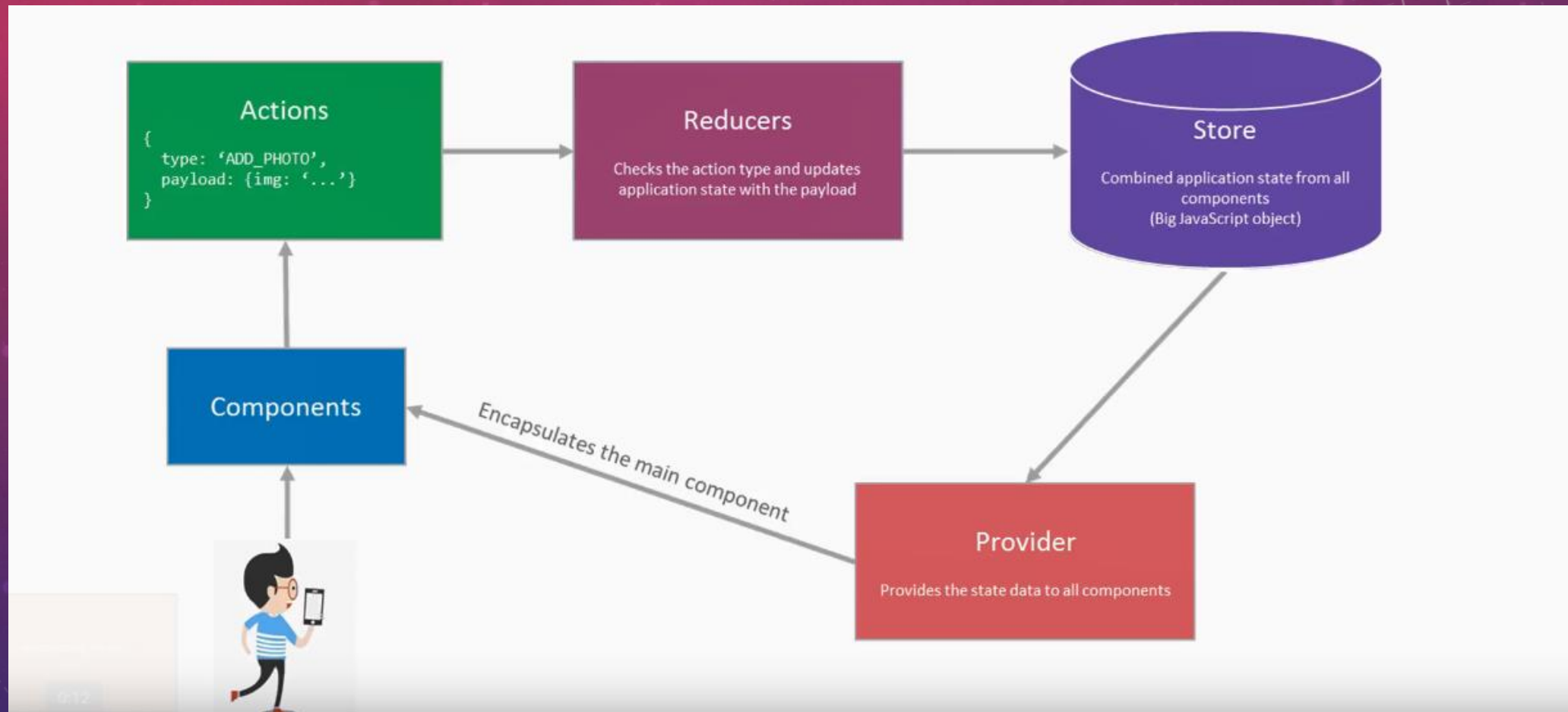


REDUX

AN OPEN-SOURCE JAVASCRIPT LIBRARY FOR MANAGING AND
CENTRALIZING APPLICATION STATE

Slides created by Sourav Saraf

REDUX FLOW



NECESSARY PACKAGES

- react-redux
- @reduxjs/toolkit

STEP 1 : CREATE A REDUCER

- Reducer is responsible for updating the redux state.
- The logic based on which the state needs to be updated is written here.
- Reducer method accepts 2 arguments, **state and action**.
- Reducer always returns the **entire state**.
- Action – An object with “type” property. Type is a name given to the operation you want to perform.
- Action can have property apart from “type”.
- Type is a user defined name, usually written in upper case. Eg.: UPDATE_VALUE
- Create a new file named rootReducer.js (name can be anything)
- Define your state and logic method here.
- Export the method.

STEP 2: CREATE THE STORE

- Open layout.js file
- Import configureStore from @reduxjs/toolkit
- This configureStore accepts an object
- We need to pass reducer in this object

STEP 3: ADDING PROVIDER

- In the layout.js file, add the “use client” annotation.
- Import Provider from ‘react-redux’
- Wrap the body tag inside the `<Provider></Provider>`
- Provider accepts the store as props.

STEP 4: USING THE REDUX STATE

- We are done with the configuration part. Now its time to use the state.
- According to the redux flow, “**An action should be dispatched, which will trigger the reducer and reducer will be updating the state. Then the updated value will be provided to the components using the redux state**”
- So we need to import a hook, useDispatch from ‘react-redux’ package.
- To use the value of the redux state, we need to use the ‘useSelector’ hook from ‘react-redux’ package.

USING THE DISPATCH HOOK

- How to use useDispatch hook ?
- After importing, write the following code:
- `const dispatch = useDispatch()` // dispatch is a variable name, not a keyword.
- Then, upon any interaction, add the code `()=>dispatch(action)`

USING THE SELECTOR HOOK

- How to use the useSelector hook?
- useSelector hook gets a default method with state as its default parameter.
- We can extract the values from the state accordingly.
- Eg: Redux State = {value: 20}
- `const val = useSelector(state => state.value)`



THANK YOU!

SOURAVSARAF3@GMAIL.COM