**Redux**

* A Predictable State Container for JS Apps
* Centralizing application's state and system logic enables powerful capabilities like undo/redo, state persistence, and much more
* Redux works with any UI layer, and has a large ecosystem of addons to fit your needs
* There are some state management tools such as Recoil, Rematch, Hookstate etc

**Before Start**

* State
* Context
* Reducer

**Install & Setup**

* First install redux with react redux
* Install redux dev tools extension
* Install vs code extension for redux dev tools

**Redux Anatomy**

* There are two main features of redux   
  -> reducer   
  -> Store / Context
* **Here is the basic state flow of Redux**



* **Advance flow of redux**



**Redux Structure**

* We can follow any kind of structure for maintaining redux
* **The standard structure of redux is**   
  -> actionTypes   
  -> actions  
  -> initialState   
  -> reducer

**Redux, React Redux kooks**

* For creating a store   
  createStore(‘reducer’, ‘middleware’);
* For Combining reducer  
  combineReducers({ });
* To get state   
  const dispatch = useSelector( state => state );
* Frontend data manage with state   
  const dispatch = useDispatch();

**Redux Middleware**

* Redux middleware run between action and reducer
* Redux middleware is used to manage redux side effect
* We can set any kind of operation before state update
* Most popular redux middleware redux-**thunk**, redux-**logger**, redux-**saga**, redux-**action**, redux-**reselect** . . .
* To use middleware into store   
  createStore(‘reducer’, ‘middleware’);
* We can set any kind of middleware by using **applyMiddleware** hook   
  createStore(‘reducer’, applyMiddleware(‘middleware\_name’, ) );
* Redux middleware structure currying function   
  const middlewareRedux = (store) => (next) => (action) => {   
   next(action);  
  }
* Install redux-thunk for manage async operation   
  npm install redux-thunk