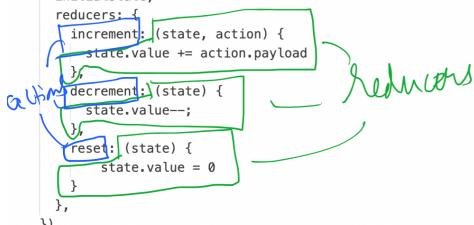


counterSlice.js

```
const initialState = { value: 0 }

const counterSlice = createSlice({
  name: 'counter',
  initialState,
  reducers: {
    increment: (state, action) {
      state.value += action.payload
    },
    decrement: (state) {
      state.value--;
    },
    reset: (state) {
      state.value = 0
    },
  },
})

export const { increment, decrement, reset } = counterSlice.actions
export default counterSlice.reducer
```



createStore

```
import counterReducer from './counterSlice';
configureStore({
  reducer: {
    counter: counterReducer,
    profile: profileReducer
  }
})
```

Root Reducer

↑
By Default

REDUX - DEV TOOLS

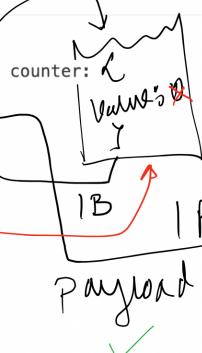
IS
ENABLED

counterSlice.js

```
const initialState = { value: 0 }

const counterSlice = createSlice({
  name: 'counter',
  initialState,
  reducers: {
    increment: (state, action) {
      state.value += action.payload
    },
    decrement: (state) {
      state.value--;
    },
    reset: (state) {
      state.value = 0
    },
  },
})

export const { increment, decrement, reset } = counterSlice.actions
export default counterSlice.reducer
```



createStore

```
import counterReducer from './counterSlice';
configureStore({
  reducer: {
    counter: counterReducer,
    profile: profileReducer
  }
})
```

Root Reducer

```
let dispatch = useDispatch();
<button type="button" onClick={() = dispatch(increment(5))}>
<button type="button" onClick={() = dispatch(decrement())}>
<button type="button" onClick={() = dispatch(reset())}>
```

①

counterSlice.js

```
const initialState = { value: 0 }

const counterSlice = createSlice({
  name: 'counter',
  initialState,
  reducers: {
    increment: (state, action) {
      state.value += action.payload
    },
    decrement: (state) {
      state.value--;
    },
    reset: (state) {
      state.value = 0
    },
  },
})

export const { increment, decrement, reset } = counterSlice.actions
export default counterSlice.reducer
```



createStore

```
import counterReducer from './counterSlice';
configureStore({
  reducer: {
    counter: counterReducer,
    profile: profileReducer
  }
})
```

Root Reducer

```
let dispatch = useDispatch();
<button type="button" onClick={() = dispatch(increment(5))}>
<button type="button" onClick={() = dispatch(decrement())}>
<button type="button" onClick={() = dispatch(reset())}>
```

①

