

# ReactJS

---

React with Redux

# Recap

- **Redux:** flux-like, single store, reducer, action

# Today's Plan

- Smart and Dumb components
- Construction of Reducers
- react-redux: A module
- Provider: Connect store with router
- mapStateToProps
- connect: components with store

# Smart and Dumb Components



smart

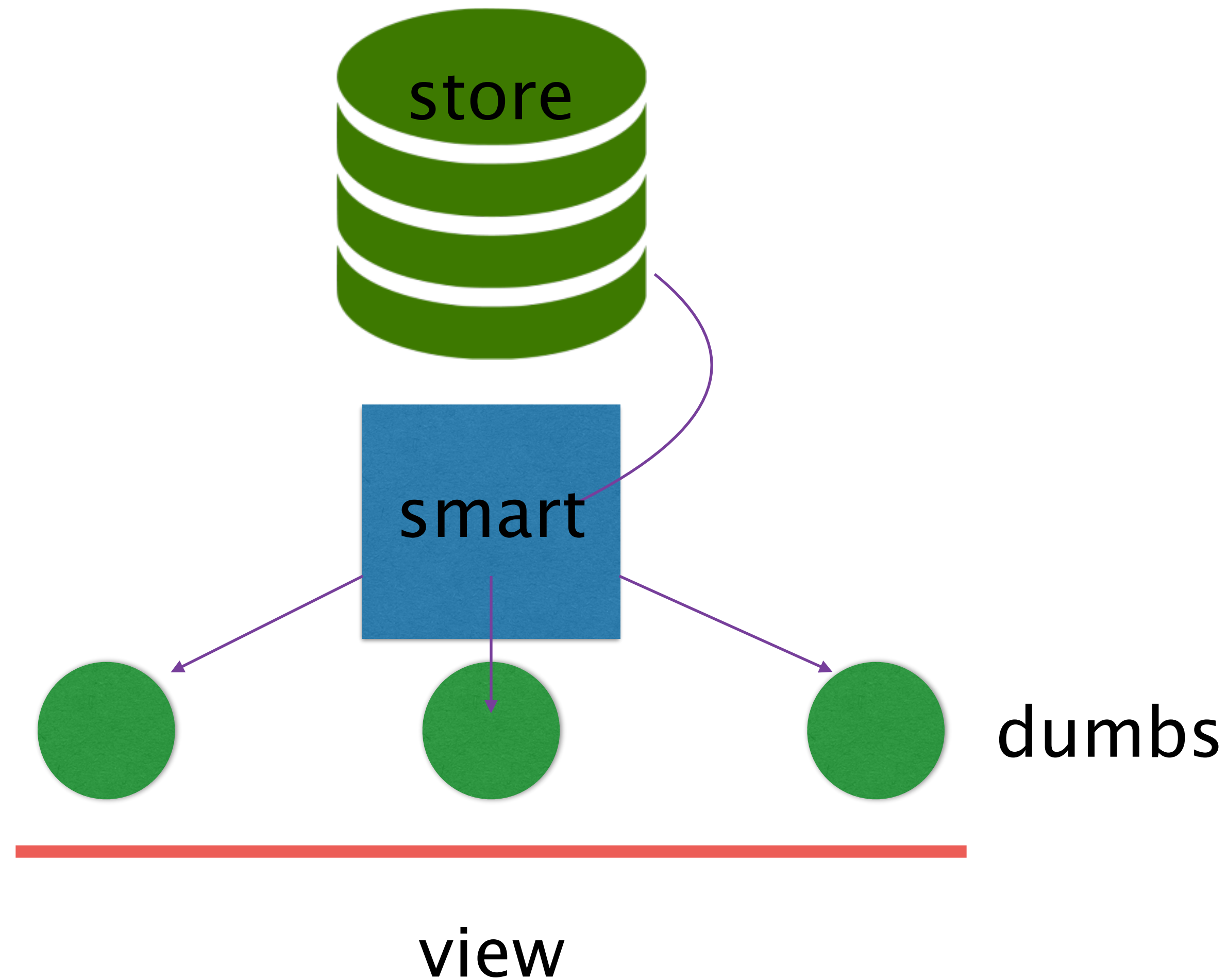
knows how to deal with data!



dumb

only view! no clue – what is data!

# Smart and Dumb Components



# Smart and Dumb Components

## Smart or Container

- Deals with data – connects to API and store via reducers
- Full of state and props
- Home for presentational or dumb components

## Dumb or View

- Does not know – when and how to deal with data
- Receives necessary data from container via props
- Reusable
- Resides inside a smart parent

# Construction of Reducers

- It is a pure function
- Generally have one initial state object
- Switch case

# Construction of Reducers

```
let initialState = {
  fname: '',
  lname: '',
  email: '',
  optIn: false,
  formObj: {}
}

let formReducer = (state = initialState, action) => {
  switch (action.type) {
    case 'ON_SUBMIT':
      return Object.assign({}, state, {
        formObj: action.formObj
      })
    case 'FORM_DATA':
      return Object.assign({}, state, {
        fname: action.fname,
        lname: action.lname,
        email: action.email
      })
    default:
      return state
  }
}

export default formReducer;
```



# react-redux

Official React bindings for Redux

<https://github.com/reactjs/react-redux>

# react-redux

```
import {provider} from 'react-redux';
```

```
import {connect} from 'react-redux';
```

# react-redux: Provider

```
import {provider} from 'react-redux';
```

Provides 'store' to react-router

```
<Provider store={store}>{router}</Provider>
```

# react-redux: mapStateToProps

a props supplier for the container

receives entire redux store as argument

# react-redux: mapStateToProps

```
const mapStateToProps = function(store) {  
  return {  
    users: store.userState.users  
  };  
}
```

# react-redux: connect()

```
connect([mapStateToProps],  
[mapDispatchToProps], [mergeProps], [options])
```

If specified, the component will subscribe to Redux store updates

contains Redux action creator

If specified, it is passed the result of  
mapStateToProps(),  
mapDispatchToProps(), and the parent  
props

pure = true  
or  
withRef = false

# react-redux: connect()

```
export default connect(mapStateToProps) (TodoApp)
```

# Summary

- Smart components will have relation with store
- Dumb components will receive props from its smart parent and display
- Reducers are pure functions
- react-redux is the module to integrate redux with react
- <Provider store> provides store to the react component via router
- mapStateToProps a props supplier to the container
- connect()() connects container with reducer



# Appendix: Material–UI

- Google's Material–UI – a hot choice for mobile first UI developers

[`http://www.material-ui.com/`](http://www.material-ui.com/)

Thanks Guys!