O2

OPEN ORIENTED

凹凸实验室

React 学习笔记（一）

Simba

Hello world

```
<!DOCTYPE html>
<html>
  <head>
    <script src="react.js"></script>
    <script src=“react-dom.js"></script>
    <script src="browser.min.js"></script>
  </head>
  <body>
    <div id="example"></div>
    <script type="text/babel">
      ReactDOM.render(
        <h1>Hello, world!</h1>,
        document.getElementById('example')
      );
    </script>
  </body>
</html>
```

- React核心库和dom扩展包
- 支持JSX语法

JSX语法

- 在js代码中直接写html

```
var element = <h1>Hello,world!</h1>
```

- 在html中插入表达式

```
var style= { fontSize: “14px” };
var element = <h1 style={style}>Hello,world!</h1>
```

- 在html中插入”html”

```
var html= ‘<img onload=“alert(1)’  src=“xxx”/>;
var element = <h1>{html}Hello,world!</h1>
```

JSX的实现

```
const element = (
  <h1 className="greeting">
    Hello, world!
  </h1>
);
```

```
const element = React.createElement(
  'h1',
  {className: 'greeting'},
  'Hello, world!'
);
```

组件

```
function Hi(props){
        return <h2>Hi, {props.name}</h2>
}

var element = <Hi name=“Hanmeimei” />
```

* Props are Read-Only

使用ES6的class定义组件

```
class Hi extends React.Component {
  render() {
    return (
        <h2>Hi, {this.props.name}</h2>
    );
  }
}

var element = <Hi name=“Hanmeimei” />
```

* Props are Read-Only

组件的状态和生命周期

```
class Clock extends React.Component {
    constructor(props) {
      super(props);
      this.state = {date: new Date()};
    }
    componentDidMount() {

    }
    componentWillUnmount() {

    }
    render() {
    }
}
```

- componentDidMount将会在组件渲染到DOM之后执行
- componentWillUnmount将会在组件被移除的时候执行
- The Data Flows Down

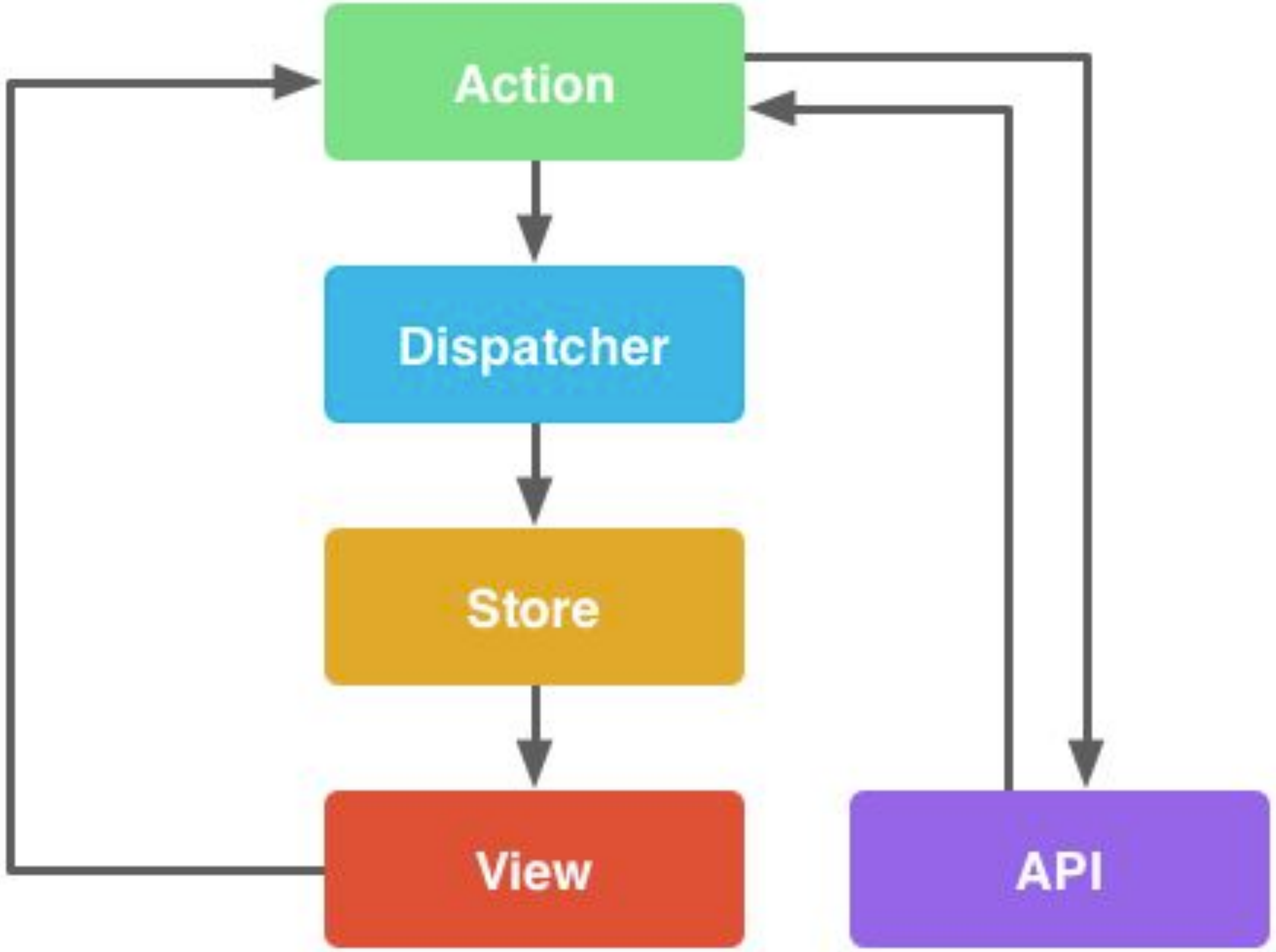
事件

```
function sayHi(e){
        console.log('Hi!');
}

function Hi(props){
        return <h2 onClick={sayHi}>Hi, {props.name}</h2>
}
```

单向数据流

```
class NameForm extends React.Component {
        handleChange(event) {
          this.setState({value: event.target.value});
        }
        render() {
          return (
                <input type="text"
value={this.state.value}
onChange={this.handleChange} />
          );
        }
}
```



THANKS

FOR YOUR WATCHING

OPEN ORIENTED

凹凸实验室