

React学习总结

演讲者：周连毅 部门：应用部

目录

- React的优点
- Flux架构
- React的缺点
- TODO

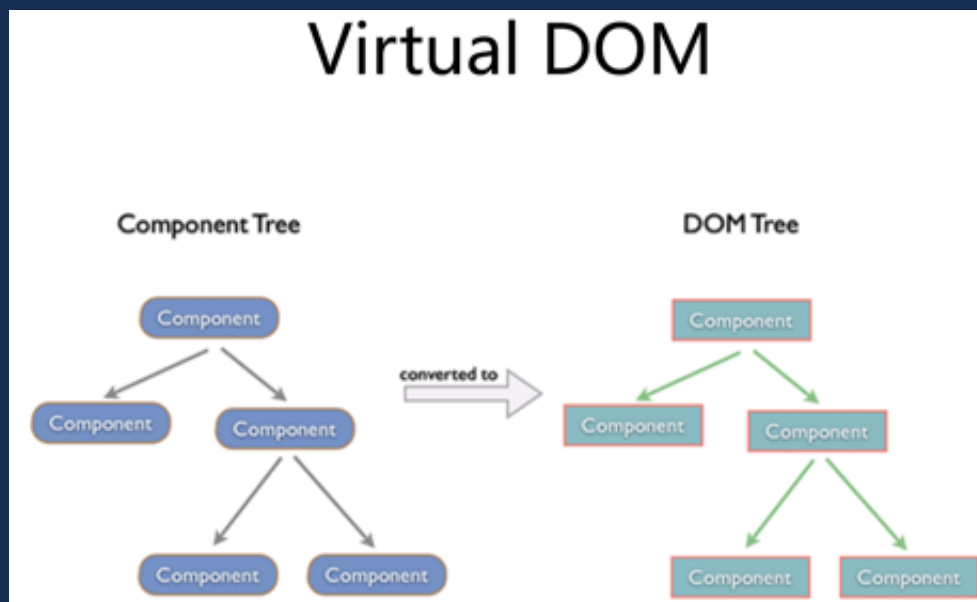
React的优点

页面渲染性能好

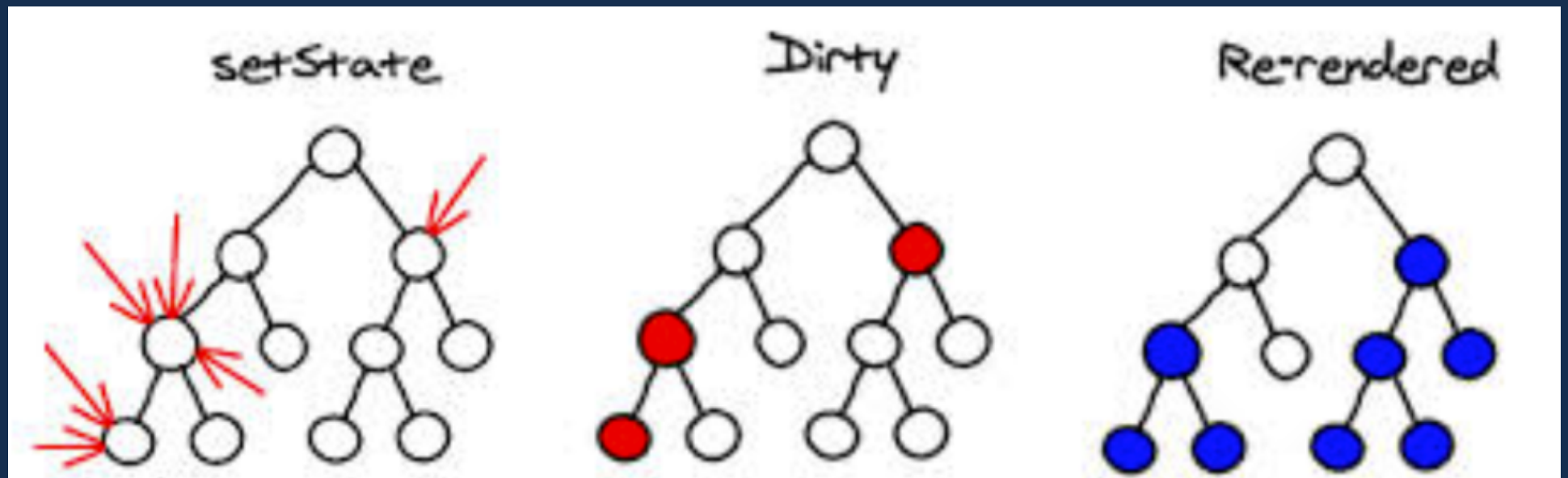
- 使用virtual DOM,避免操纵DOM,保证页面更流畅
- DOM的写操作批量进行,减少layout计算次数

```
ReactElement {  
  type: 'div',  
  ref: null,  
  key: null,  
  props: Object  
}
```

Virtual DOM



渲染过程









<http://jonmiles.github.io/react-performance-tests>

组件化

```
<Calendar locale={CalendarLocale}
  showTime={this.state.showTime}
/>;
```

- 松耦合
- 可复用 <http://react-components.com/>

Branch: master ▾ wp-calypso / shared / components / card /	
 aduth Framework: Use classnames in place of react/lib/joinClasses	
..	
 docs	Initial commit of wp-calypso
 README.md	Update README.md
 compact.jsx	Framework: Use classnames in place of react/lib/joinClasses
 index.jsx	Framework: Use classnames in place of react/lib/joinClasses
 style.scss	Initial commit of wp-calypso

- 维护方便

事件绑定消耗内存小

- 所有的事件都是从body上冒泡的

```
5ekp5gcvsw.1.0.2.0.0">Latest  
ponents</h2>  
  data-reactid=".15ekp5gcvsw.1.0.2.0.1">  
  li data-reactid=  
5ekp5gcvsw.1.0.2.0.1.$lapanoid-react-  
tube">  
  <a class="component-name" href="/  
component/lapanoid-react-youtube" data-  
reactid=  
".15ekp5gcvsw.1.0.2.0.1.$lapanoid-  
react-youtube.0">lapanoid-react-  
youtube</a>  
/li>  
li data-reactid=  
5ekp5gcvsw.1.0.2.0.1.$frh-react-dial-  
">...</li>  
li data-reactid=  
5ekp5gcvsw.1.0.2.0.1.$react-pagers">  
li>
```

Styles	Computed	Event Listeners
--------	----------	-----------------

 <input checked="" type="checkbox"/> Ancestors		
---	--	--

▶ blur		
--------	--	--

▶ capturePageEvt		
------------------	--	--

▶ change		
----------	--	--

▶ checkFSAvailabilityEvt		
--------------------------	--	--

▼ click		
---------	--	--

▶ document		
------------	--	--

▼ focus		
---------	--	--

▶ document		
------------	--	--

▼ input		
---------	--	--

▶ document		
------------	--	--

▼ keydown		
-----------	--	--

▶ document		
------------	--	--

▶ document		
------------	--	--

[Docs](#)[Support](#)[Download](#)[Blog](#)[GitHub](#)[React Native](#)[Supported Tags and Attributes](#)[Event System](#)[DOM Differences](#)[Special Non-DOM Attributes](#)[Reconciliation](#)[Web Components](#)[React \(Virtual\) DOM Terminology](#)

Event delegation: React doesn't actually attach event handlers to the nodes themselves. When React starts up, it starts listening for all events at the top level using a single event listener. When a component is mounted or unmounted, the event handlers are simply added or removed from an internal mapping. When an event occurs, React knows how to dispatch it using this mapping. When there are no event handlers left in the mapping, React's event handlers are simple no-ops. To learn more about why this is fast, see [David Walsh's excellent blog post](#).

<https://facebook.github.io/react/docs/interactivity-and-dynamic-uis.html>

测试有保障

- 组件化测试简单
[https://github.com/react-](https://github.com/react-component/calendar/tree/master/tests)
- [component/calendar/tree/master/tests](https://github.com/react-component/calendar/tree/master/tests)
- 官方提供测试文档
<https://facebook.github.io/react/docs/test-utils.html>
- 官方提供测试框架 <https://facebook.github.io/jest/>

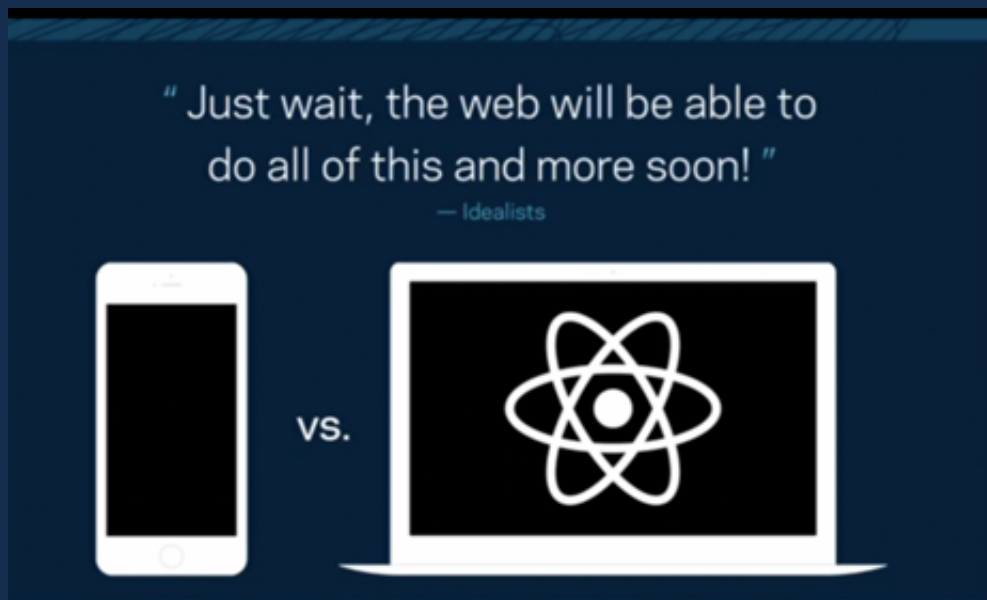
数据驱动页面渲染

- 通过state和props控制数据,渲染自动更新
- 例子: todomvc

<https://github.com/lianyzhou/react-todomvc-basic>

跨平台 / 终端

- 支持前端渲染/node.js端渲染(SEO友好)
- 支持web端/手机端(React Native)



比较流行 比较前沿

- 关注程度高,github上star多
- 国外有网站在用 <http://www.zehnergroup.com/>
- 国内有网站在用 <https://jianliao.com>
- 支持ES6
- 与Broserify,Webpack结合紧密

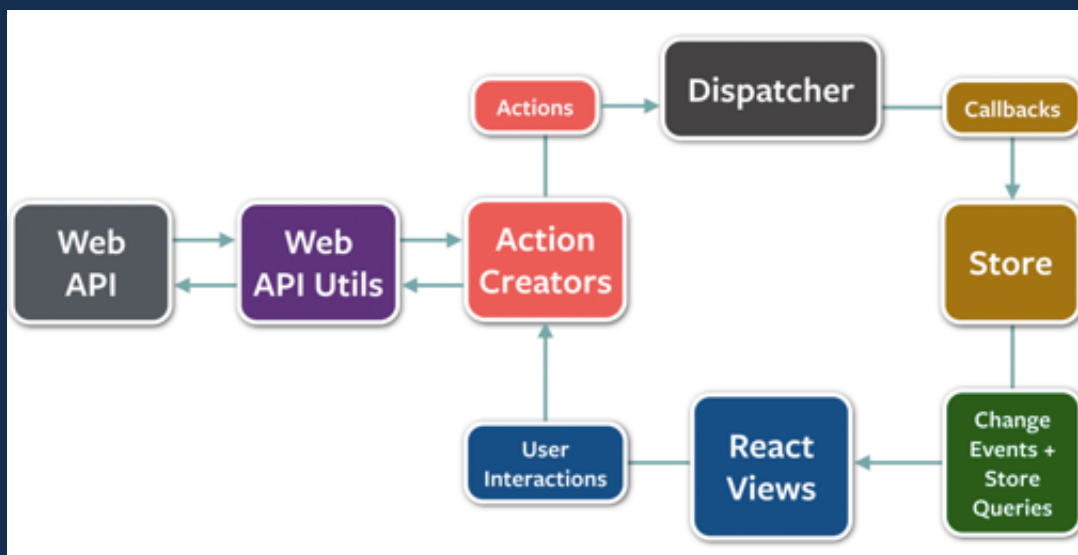
生态圈比较繁荣

- 国内有论坛专门研究 <http://www.react-china.org/>
- 有基于react component的仓库
<http://react-components.com/>
- 与react搭配做单页应用的插件火爆
 - react-router
 - react-bootstrap
- flux相关的实现很多
 - redux
 - reflux
 - fluxxor
 - alt

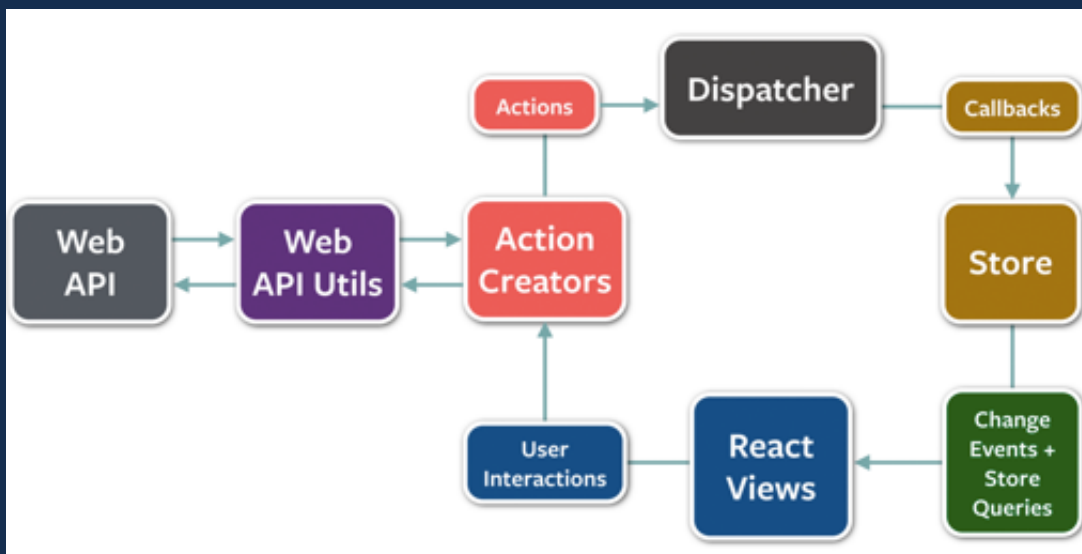
Flux架构

Flux

- 单向数据流, 一种模式



Flux



Action 用户交互、数据交互都是一个Action

Dispatcher 分发动作给 Store

Store 数据存储中心，响应 Action，对数据进行修改

View 监听数据变化的事件，从 Store 获取数据，触发 Action

React的缺点

jsx模板写起来费劲

```
render : function() {  
  var toggleComplete = this.toggleComplete;  
  var removeTodo = this.removeTodo;  
  
  return (  
    <section className="main">  
      <ul className="todo-list" style={{display:this.props.todos.length ? 'block' : 'none'}}>  
        {  
          this.props.todos.map(function(todo) {  
            var checkbox;  
            if(todo.completed) {  
              checkbox = (  
                <input className="toggle" type="checkbox" checked="checked" onClick={toggleComplete.bind(nu  
              );  
            } else {  
              checkbox = (  
                <input className="toggle" type="checkbox" onClick={toggleComplete.bind(null , todo.id)}>  
              );  
            }  
  
            return <li className={classNames({completed : todo.completed})}>  
              <div className="view">  
                {checkbox}  
                <label>{todo.todo}</label>  
                <button className="destroy" style={{cursor:'pointer'}} onClick={removeTodo.bind(null ,  
              </div>  
            </li>  
          })  
        }  
      </ul>  
    </section>  
  );  
}
```

专注于模板渲染,不是完整的MVC框架

- 自己想办法做router react-router ?
- 自己想办法做MVC
 - Backbone,将handlebars替换 ?
 - 用Flux,官方实现,非官方实现 ?

官方没有提供大型项目架构和源码

- 大家讨论,制定一个架构
- 参考github上非官方脚手架的架构
- 参考线上使用了react的网站的架构

不能与requirejs和seajs集成

- 更改源码可以实现与requirejs和seajs集成
- 学习用browserify或webpack进行开发

TODO

TODO

- react-router
- MVC选型
- webpack或browserify
- ES6
- 测试
- react开发环境的搭建
- immutable-js

Q & A