

QCon 全球软件开发大会 【北京站】2016

ReactMix HTML+CSS+JS写ReactNative

薛端阳(addison xue)

ReactNative第一印象



ReactNative第一印象

- 看着像CSS的用法，但是不是CSS
- ReactJs的语法风格+好多自定义控制
- 各种异步的写法，很多在浏览器是同步的api到了rn都要异步
- 对于我们已有项目最大的问题

重构成本

能不能简单点



ReactMix第一印象



```
209 class CXYLHL extends App {  
210     constructor(){  
211         super();  
212         this.state = {  
213             bodyClass: ''  
214         };  
215         if (is_someday()) {  
216             this.state.bodyClass = 'someday'  
217         }  
218         this.state.good = [];  
219         this.state.bad = [];  
220         pickTodaysLuck(this.state.good, this.state.bad);  
221     }  
222     componentDidMount(){  
223         pickTodaysLuck();  
224     }  
225  
226     render() {  
227         return (  
228             <Body>  
229             <Div id="J_body" className={this.state.bodyClass}>  
230                 <Div className="container">  
231                     <Header>  
232                         <Div className="title">  
233                             <Div className="header"><Span>程序员老黄历</Span><Div className="sup">be  
234                             </Div>  
235                         </Header>  
236                         <Div className="date">  
237                             {getTodayString()}  
238                         </Div>  
239                         <Div className="good">  
240                             <Div className="title">  
241                                 <Div>  
242                                     <Div className="b">宜</Div>  
243                                 </Div>  
244                             </Div>  
245                             <Div className="content">  
246                                 {  
247                                     this.state.good.map(function(item) {  
248                                         return (  
249                                             <Div>  
250                                                 <Div>{item}</Div>  
251                                             </Div>  
252                                         </Div>  
253                                     )  
254                                 )  
255                             </Div>  
256                         </Div>  
257                     </Header>  
258                     <Content>  
259                     </Content>  
260                 </Div>  
261             </Body>  
262         )  
263     )  
264 }
```

ReactMix第一印象

- 这个就是CSS吗
- 看着很像HTML，为啥Div是首字母大写
- 支持多className的选择器 好像比weex牛逼？

很像H5

为啥要有ReactMix

- 我们的业务中有90%的历史代码
- 这部分代码是基于传统的HTML+CSS+JS开发的
- 我们希望能使用ReactNative带来的优势
- 又不想花费过多的成本来重新翻译

所以有了ReactMix

- 帮助我们翻译已有的H5代码比较容易的过渡到ReactNative
- 解决一套代码，H5，安卓，iOS通用的问题

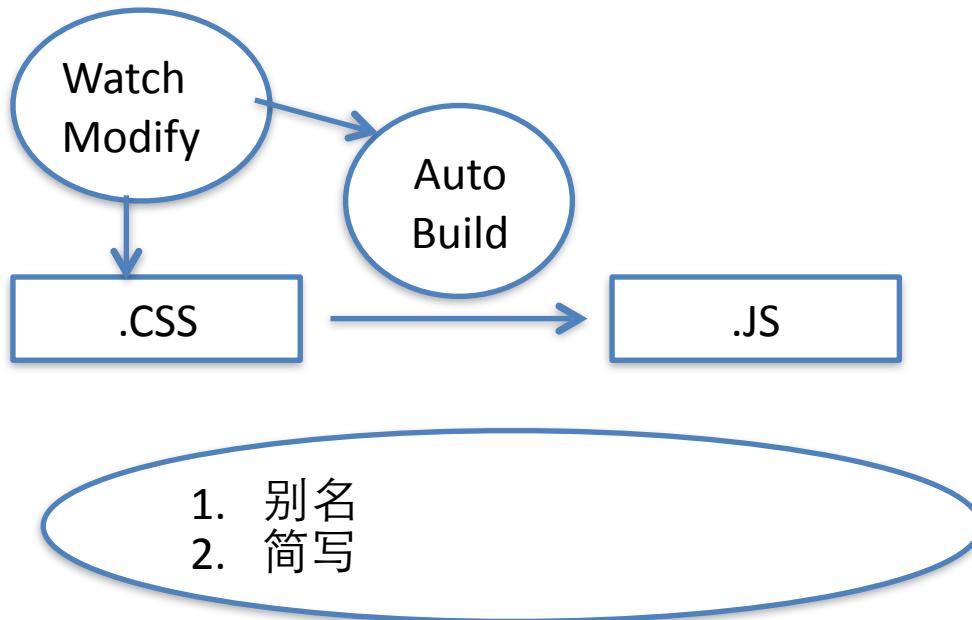


ReactMix如何支持CSS

- 直接使用原始版本的Css
- 巧妙用于ReactNative的require函数是静态编译加载的特性
- ImportCSS("require('path')")

ReactMix如何支持CSS

- 静态语言翻译



```
MicrosoftApp.html MicrosoftApp.css MicrosoftApp.js MicrosoftApp.resx MicrosoftApp.manifest
```

```
1<div> 1<br> 2<div class="container"> 3<h1>Hello World!</h1> 4<h2>This is a Microsoft App!</h2> 5<img alt="A small decorative icon." data-bbox="480 10 520 50"/> 6<div> 7<h3>Header</h3> 8<ul style="list-style-type: none; padding-left: 0; margin: 0; font-size: 0.8em; color: #808080; font-weight: bold; background-color: #f0f0f0; border-bottom: 1px solid #ccc; position: relative; z-index: 1; "> 9<li> 10<a href="#">Home</a> 11<a href="#">About</a> 12<a href="#">Contact</a> 13</li> 14</ul> 15<div style="background-color: #f0f0f0; padding: 10px; margin-top: 10px; border-radius: 5px; position: relative; z-index: 0; "> 16<div style="text-align: center; margin-bottom: 20px; border: 1px solid #ccc; border-radius: 5px; padding: 10px; width: fit-content; margin: auto; "> 17<h3>Container</h3> 18<ul style="list-style-type: none; padding-left: 0; margin: 0; font-size: 0.8em; color: #808080; font-weight: bold; background-color: #f0f0f0; border-bottom: 1px solid #ccc; position: relative; z-index: 1; "> 19<li> 20<a href="#">Home</a> 21<a href="#">About</a> 22<a href="#">Contact</a> 23</li> 24</ul> 25<div style="text-align: center; margin-bottom: 20px; border: 1px solid #ccc; border-radius: 5px; padding: 10px; width: fit-content; margin: auto; "> 26<h3>Footer</h3> 27<ul style="list-style-type: none; padding-left: 0; margin: 0; font-size: 0.8em; color: #808080; font-weight: bold; background-color: #f0f0f0; border-bottom: 1px solid #ccc; position: relative; z-index: 1; "> 28<li> 29<a href="#">Home</a> 30<a href="#">About</a> 31<a href="#">Contact</a> 32</li> 33</div>
```

ReactMix如何支持CSS

- 简写

Margin/Padding/border/...

基于AST静态语法分析 -> 扩展还原成为完整css 定义

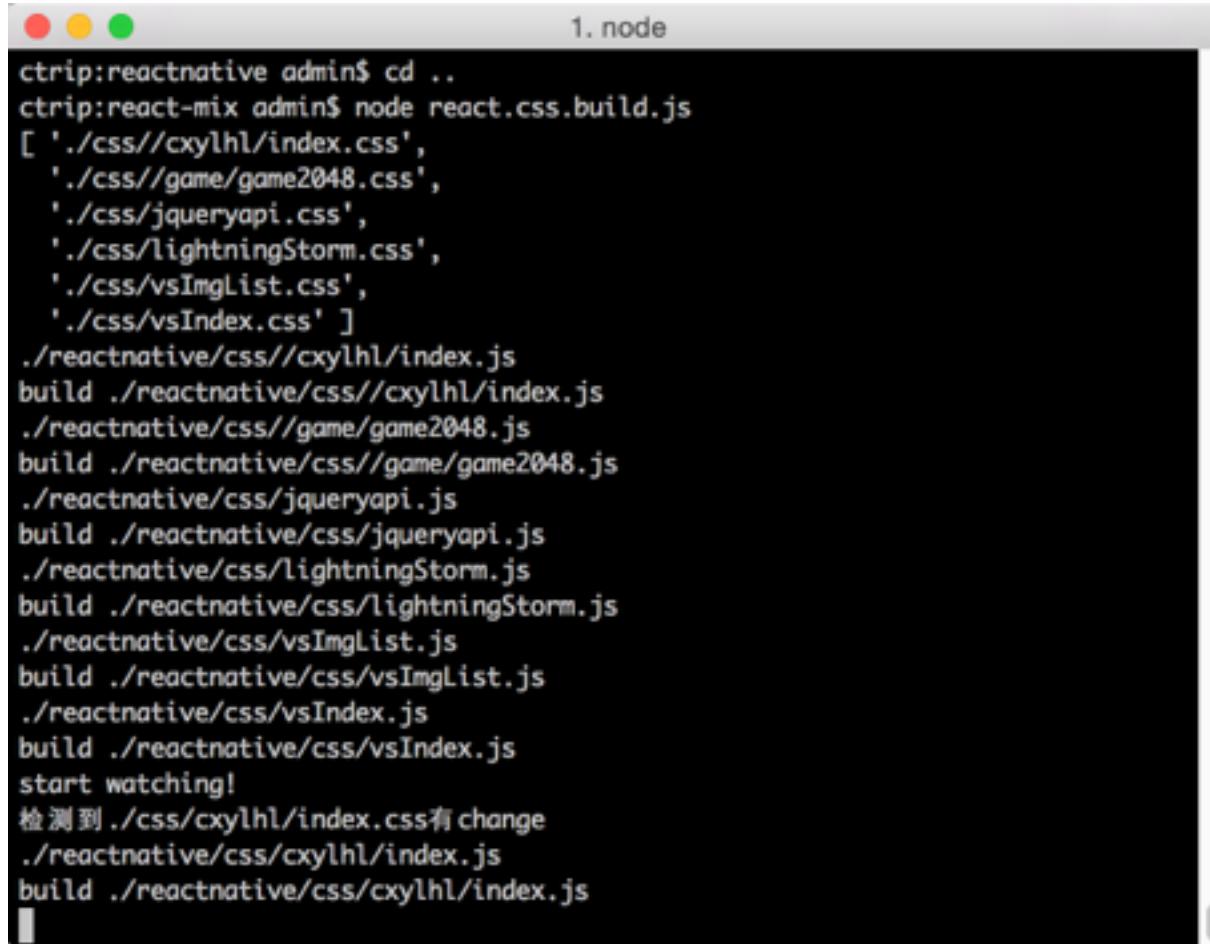
- Media Query

在组件的render方法的时候基于上下文环境变量做解析

还支持比如zIndex,backgroundImage,display等等，不仅仅使用CSS-layout提供的布局模型，还支持更多在H5开发中常用的CSS布局模式，未来会包括对于float浮动的支持

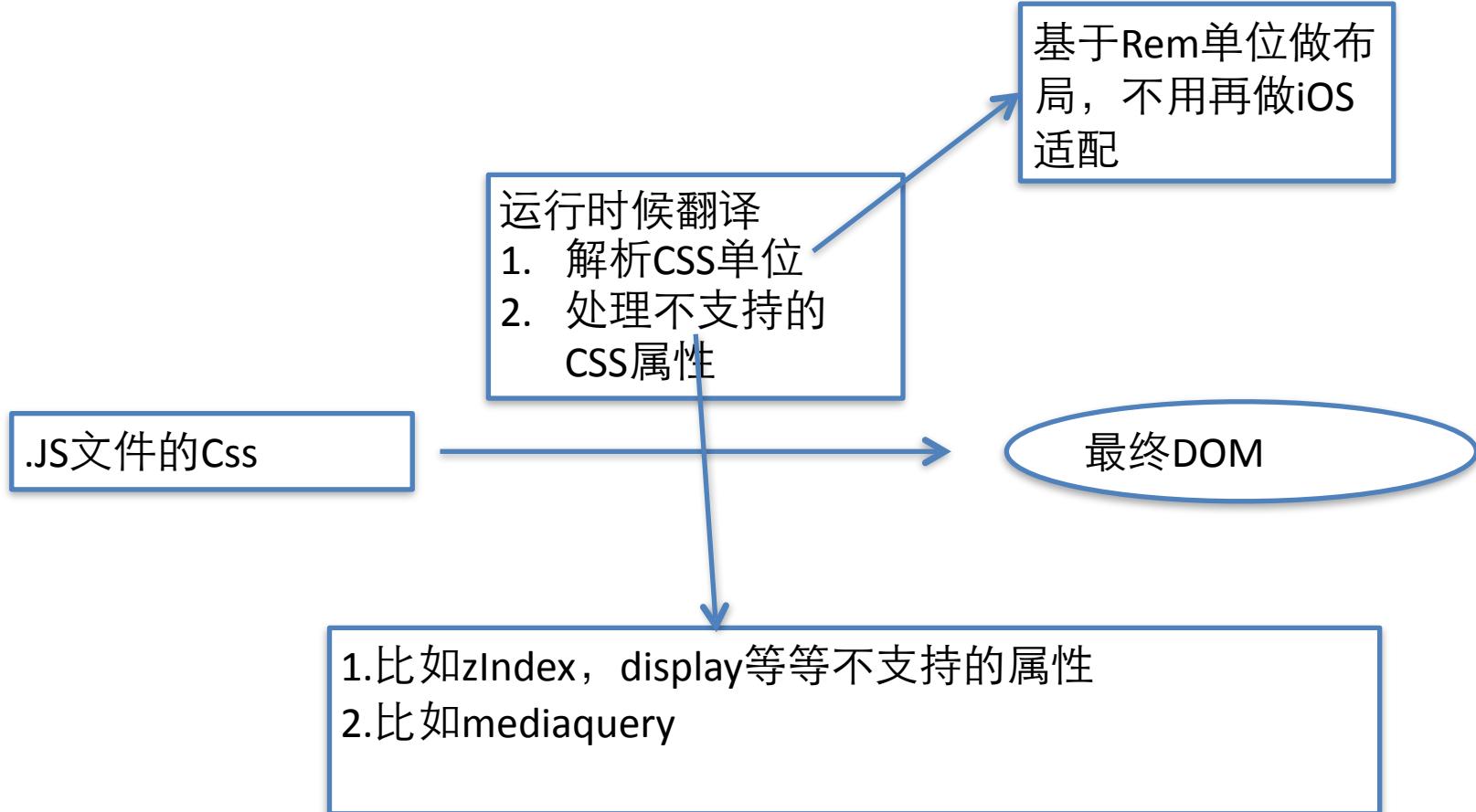
如何复用原有的Css

- 通过自动工具Watch CSS文件，同步生成子集的Css Js格式



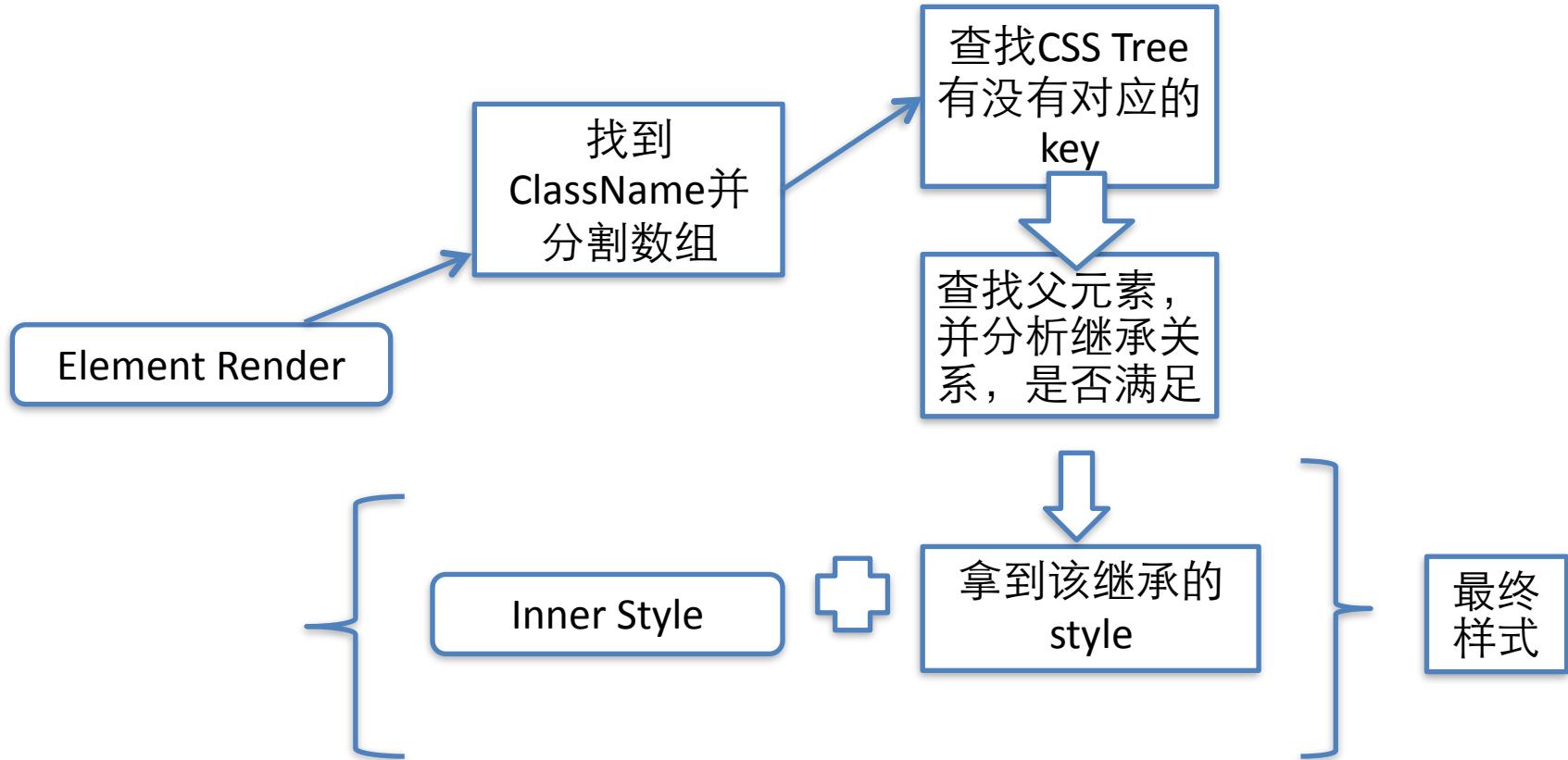
```
1. node
ctrip:reactnative admin$ cd ..
ctrip:react-mix admin$ node react.css.build.js
[ './css//cxylhl/index.css',
  './css//game/game2048.css',
  './css/jqueryapi.css',
  './css/lightningStorm.css',
  './css/vsImgList.css',
  './css/vsIndex.css' ]
./reactnative/css//cxylhl/index.js
build ./reactnative/css//cxylhl/index.js
./reactnative/css//game/game2048.js
build ./reactnative/css//game/game2048.js
./reactnative/css/jqueryapi.js
build ./reactnative/css/jqueryapi.js
./reactnative/css/lightningStorm.js
build ./reactnative/css/lightningStorm.js
./reactnative/css/vsImgList.js
build ./reactnative/css/vsImgList.js
./reactnative/css/vsIndex.js
build ./reactnative/css/vsIndex.js
start watching!
检测到./css/cxylhl/index.css有change
./reactnative/css/cxylhl/index.js
build ./reactnative/css/cxylhl/index.js
```

ReactMix如何支持CSS



如何复用原有的Css

- 通过运行时动态查找，来支持常用className的继承和组合



CSS combine and inherit

- 组合

遍历该节点所有的className,按照字母顺序排序后组合为key

- 继承
- .classB
- .classA .classB

```
{  
  ^classB: [  
    {  
      prefix: [ 'classA', 'class...' ]  
      style: {  
        content  
      }  
    }  
  ]  
}
```

CSS Unit

- 通过运行时动态翻译，来支持rem,em,pt,px等CSS单位



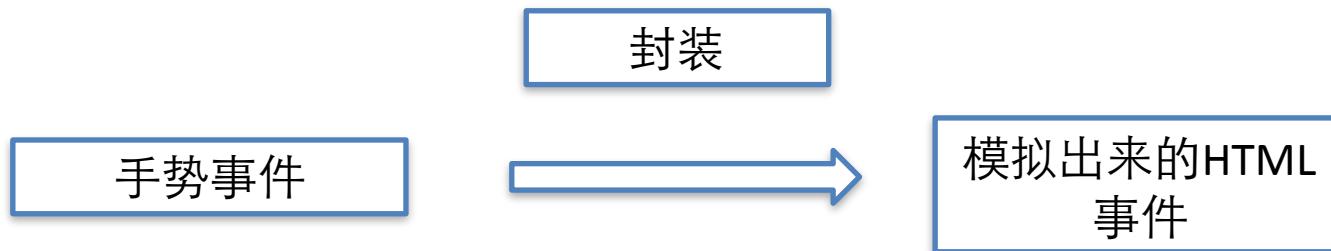
搞定了CSS接下来就是HTML

- 使用Div表示块标记，Span表示内联标记

View	Div
Text	Span
自定义	Input
自定义	A
Image	Img
...	...

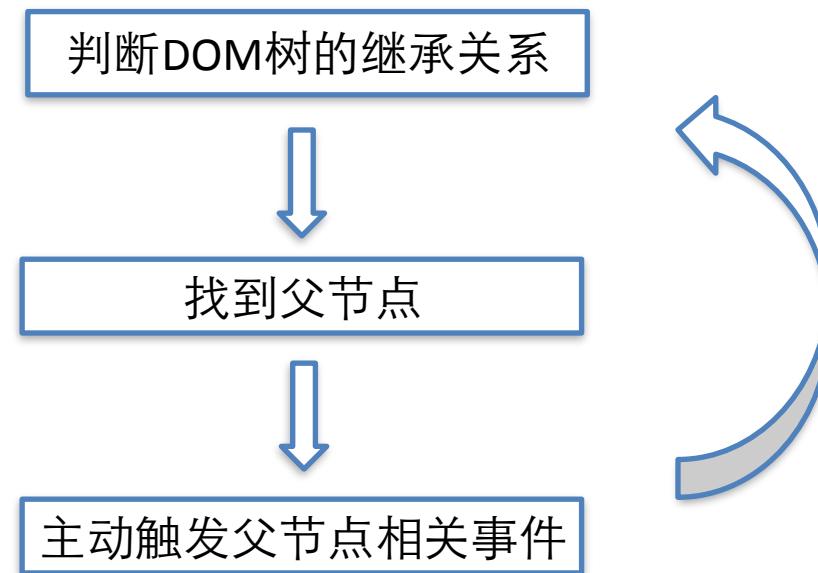
HTML节点的常用API

- ReactNative不支持标准的HTML事件，比如onClick或者onTouch之类
- ReactNative不支持我们常用的选择器直接找到一个元素，然后做一些常用的DOM操作



HTML常用事件和冒泡

- 支持标准的HTML事件以及事件冒泡



事件对象

- ReactNative原生是支持手势事件
- 基于每个通过ReactMix New HTMLComponent都默认创建了所有标准HTML事件的空实现委托
 - 在每个委托内，抽象了对于标准事件的模拟
 - 实现了类似DomEvent对象
 - 手动触发它的冒泡机制

基于事件委托的方式，实现了标准事件对象的
preventDefault,StopImmidately,StopPropagation以及returnValue

封装基本对象

- 所有的ReactMix元素都派生与Element对象
 - Element对象是继承于ReactComponent
 - Element对象自身实现了常用DOM API的封装
 - 所有通过类似Sizzle选择器找到的对象都是一个Element对象实例

DOM API语法糖

- 支持类似JQuery的API

```
oxyhi.js index.js example_onclick example_ontouch example_jquerya
1 import './common/LightningStorm';
2 includeCSS('require("./css/jqueryapi")');
3
4 class TestPage extends App{
5     render(){
6         return (
7             <Div id="J1" onClick={()=>[alert(1)]}>
8                 <Div id="J2" onClick={()=>[alert(2)]}>
9                     // 对于jQuery API的测试，你可以测试这个节点有哪些常用的api可以操作
10                </Div>
11                <Button onClick={this.btn1.bind(this)}>addClass cls2</Button>
12                <Button onClick={this.btn2.bind(this)}>addClass cls1</Button>
13                <Button onClick={this.btn3.bind(this)}>removeClass cls1</Button>
14                <Button onClick={this.btn4.bind(this)}>removeClass cls2</Button>
15            </Div>
16        )
17    }
18    btn1(){
19        $('#J2').addClass('cls2');
20        console.debug($('#J2').classList)
21    }
22    btn2(){
23        $('#J2').addClass('cls1');
24        console.debug($('#J2').classList)
25    }
26    btn3(){
27        $('#J2').removeClass('cls2');
28        console.debug($('#J2').classList)
29    }
30    btn4(){
31        $('#J2').removeClass('cls1');
32        console.debug($('#J2').classList)
33    }
34 }
35 }).run();
```

解决全局查找元素的难点

- 支持基本的#id,.className,tagName方式定位元素

```
Element.js cxyhl.js example_listvie index.css example_addEven 5
1 import './common/LightningStorm';
2
3 class TestPage extends App{
4     componentDidMount(){
5         $('#J2').addEventListener('click', function(){
6             alert(3);
7         })
8     }
9     render(){
10        return (
11            <Div id="J1">
12                <Div id="J2">
13                    这是一个addEventListener的例子，并且支持冒泡，你可以修改index.ios.js，删除index.ios.js，删除index.ios.js
14                </Div>
15            </Div>
16        );
17    }
18 }
19 ).run();
20
21
22
```

解决全局查找元素的难点

- 在元素生成的生命周期函数注册
- 在销毁时候删除

模拟标准的HTML Element

- 通过继承React.Component
- 创建自定义的事件委托，模拟默认的HTML事件
- 修改默认的render方法，来支持对于CSS属性的解析和转换成为标准的ReactNative元素
- 通过扩展自定义的元素，来实现类似Jquery的DOM API

异步API写起来像同步API

- 维护异步API关键词表
- 在ReactNative Start的rn官方的自动编译脚本插入一个reactmix task
- 在关键词前和方法定义前插入async + await 实现codestyle的同步

```
function method(){  
  var a=localStorage.getItem('key')  
  ...  
}
```

转换成

```
async function method(){  
  var a= await localStorage.getItem('key')  
  ...  
}
```

用了这些多ES7，H5咋办？



兼容打包H5的代码

- 因为使用了ES7的语法，需要通过babel等工具做代码降级



A screenshot of a code editor showing a file named 'webpack.config.' with a tab bar above it containing 'Element.js', 'cxyihl.js', 'index.css', 'example_addEven', and 'webpack.config.'. The code editor displays the following code:

```
1 var webpack = require('webpack');
2
3 module.exports = {
4     entry : [ 'babel-polyfill', './reactnative/common/LightningStorm.js',
5               './web/common/ui.js', './reactnative/index.ios.js' ],
6     output : {
7         filename : './web/index.web.build.js'
8     },
9     module : {
10         loaders : [ {
11             loader : 'babel-loader',
12             exclude : /node_modules/,
13             query : {
14                 plugins : [ 'transform-object-assign' ],
15                 presets : [ 'es2015', 'stage-0', 'react' ]
16             }
17         }],
18         noParse : [ /Dimensions/ ],
19     },
20     plugins : [ new webpack.IgnorePlugin(/(^?!babel-polyfill)\w+$/)(\.\.\rn\.\w+)(.*\.\png)/)
21 }
```

对于第三方自定义控件

- 基于全局空间的方式引入
- 不同的平台各自定义

扩展require方法的实现，不仅仅支持CMD打包，还支持H5的AMD加载方式

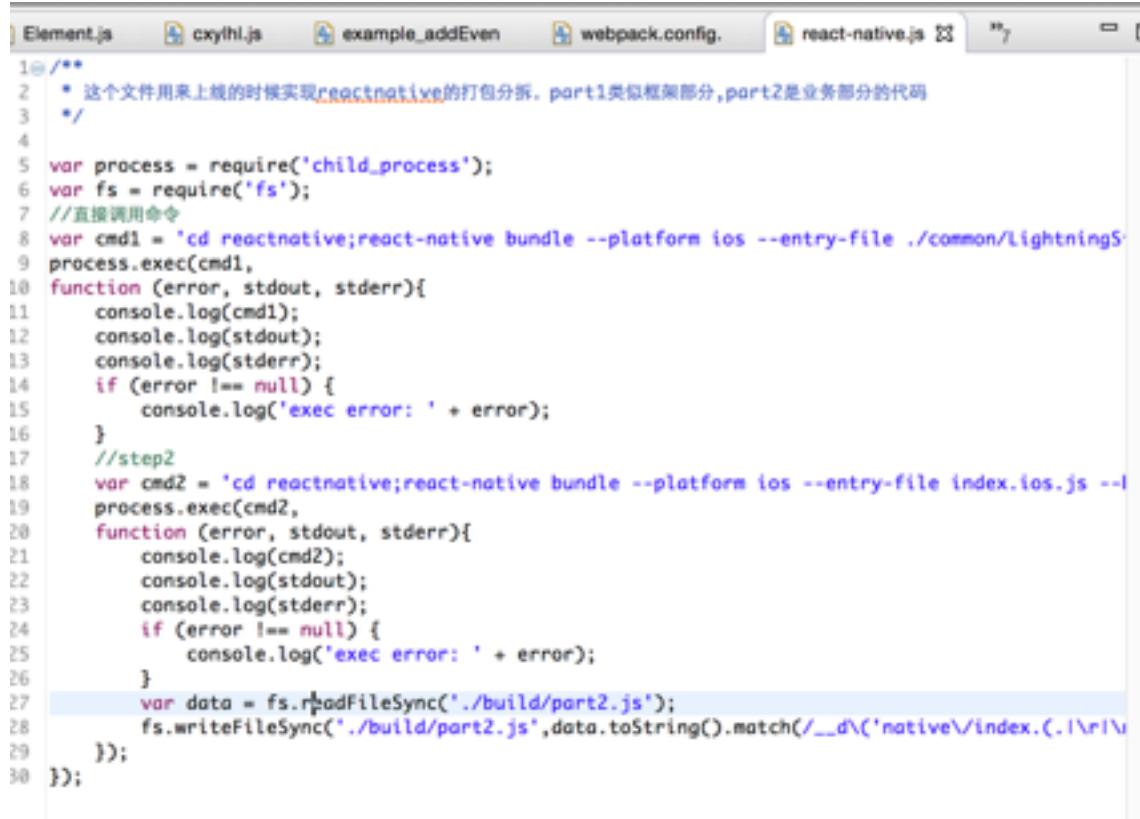


上线代码会不会太大？



打包分割

- Common包，通过框架require大部分使用且不经常变更的组件
- 剩下的业务包即业务代码



The screenshot shows a code editor window with several tabs at the top: Element.js, cxyhi.js, example_addEven, webpack.config., and react-native.js. The react-native.js tab is active, showing line numbers 1 through 38. The code is a JavaScript script designed to split a React Native application into two parts: a common framework part and a business logic part.

```
1 /**
2  * 这个文件用来上线的时候实现reactnative的打包分拆。part1类似框架部分,part2是业务部分的代码
3 */
4
5 var process = require('child_process');
6 var fs = require('fs');
7 //直接调用命令
8 var cmd1 = 'cd reactnative;react-native bundle --platform ios --entry-file ./common/LightningS';
9 process.exec(cmd1,
10 function (error, stdout, stderr){
11     console.log(cmd1);
12     console.log(stdout);
13     console.log(stderr);
14     if (error !== null) {
15         console.log('exec error: ' + error);
16     }
17     //step2
18     var cmd2 = 'cd reactnative;react-native bundle --platform ios --entry-file index.ios.js --l
19 process.exec(cmd2,
20 function (error, stdout, stderr){
21     console.log(cmd2);
22     console.log(stdout);
23     console.log(stderr);
24     if (error !== null) {
25         console.log('exec error: ' + error);
26     }
27     var data = fs.readFileSync('./build/part2.js');
28     fs.writeFileSync('./build/part2.js',data.toString().match(/_d\('native'\)/g).join('')
29 });
30});
```

最终效果

- RN JS框架自身大小在200k左右，其中因为需要支持动态解析所以\引入reactmix部分大约在20k不到，其他都是静态语言编译工具
- 页面的平均渲染时间在iOS 200ms，安卓400ms左右，平均比Hybrid加载速度快30%

项目github地址, <https://github.com/xueduany/react-mix>,
或者github直接搜索react-mix



2000年
业务始于

2015中国旅游集团第1名
国内最大机票电商平台

7000+
人员规模

6大洲
业务覆盖

5000+
城市覆盖

700万+
月预定量

2.5亿
会员数量

10亿
APP
累计下载

无线端日交易额峰值破5亿

很缺 jp-recruit@ctrip.com

