

# React 同构实践

# About me

2012 - 2015      百度

2015 - now      腾讯



<http://imweb.io>



# 同构

- React & Redux
- why
- How
- Challenge

# Thinking in React

# Thinking in React



web component

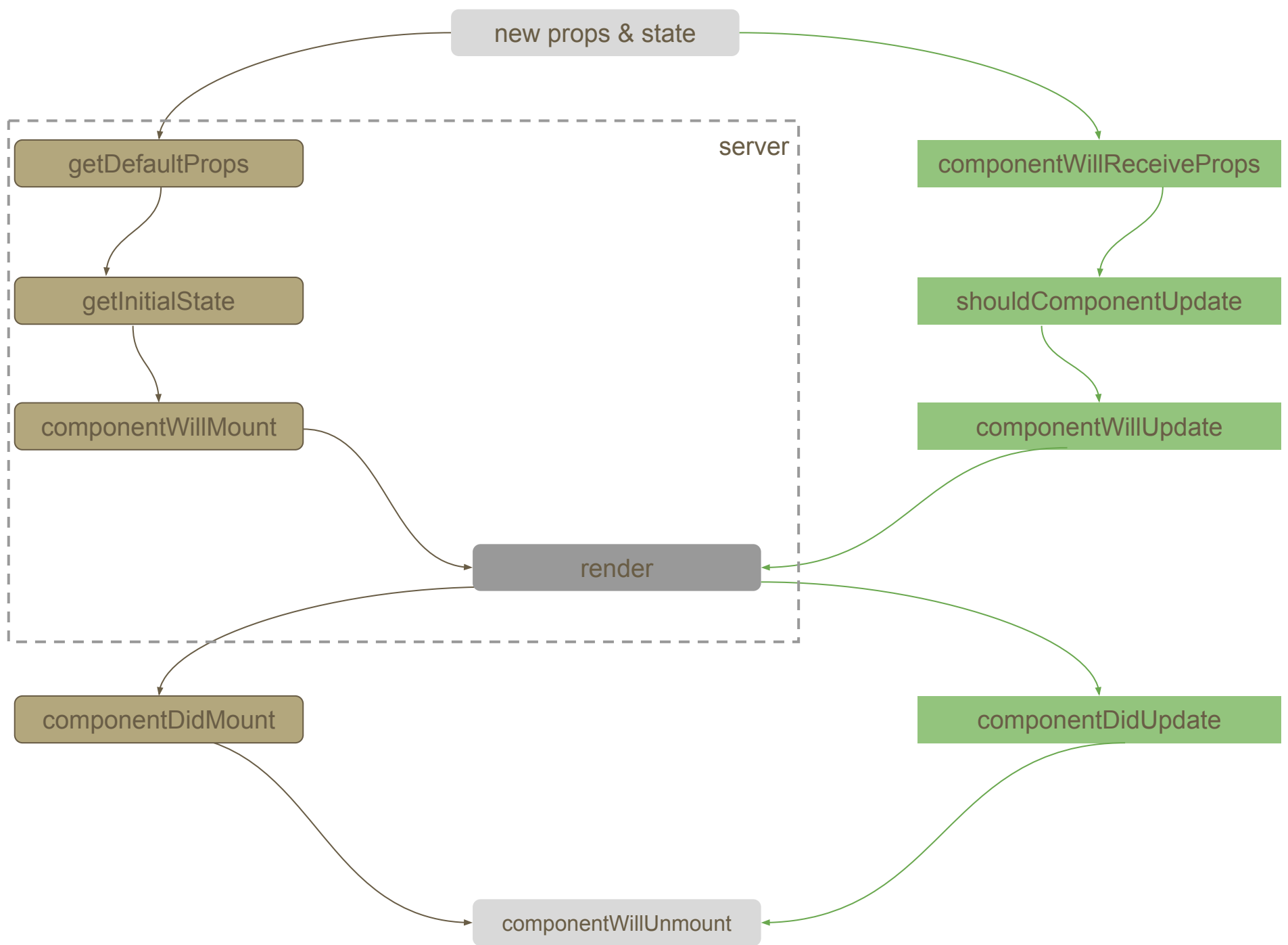
# Thinking in React

```
class Face extends React.Component {  
  render () {  
    return (  
      <div>  
        <img src={this.props.url} />  
  
        <span>  
          {this.props.name}  
        </span>  
      </div>  
    );  
  }  
}
```

# Thinking in React

```
let url = 'http://a.com/b.png';
let name = 'foo';

ReactDOM.render(
  (
    <Page>
      <Header>
        <Face url={url} name={name} />
      </Header>
      <Tab />
      <Content />
      ...
    </Page>
  ),
  document.body
);
```





# Thinking in Redux

# Thinking in Redux

- 复杂页面的状态变化
  - 组件间通信
- 
- 单一数据源
  - 单向数据流

# Thinking in Redux



UI

```
{
  user: {
    name: 'foo',
    url: 'bar.png'
  },
  tab: {
    activeIndex: 1,
    list: ['tab1', 'tab2', 'tab3']
  }
}
```

store

state =  
**reducer**  
(data,preState)

data = dispatch(**action**)

# 业务场景



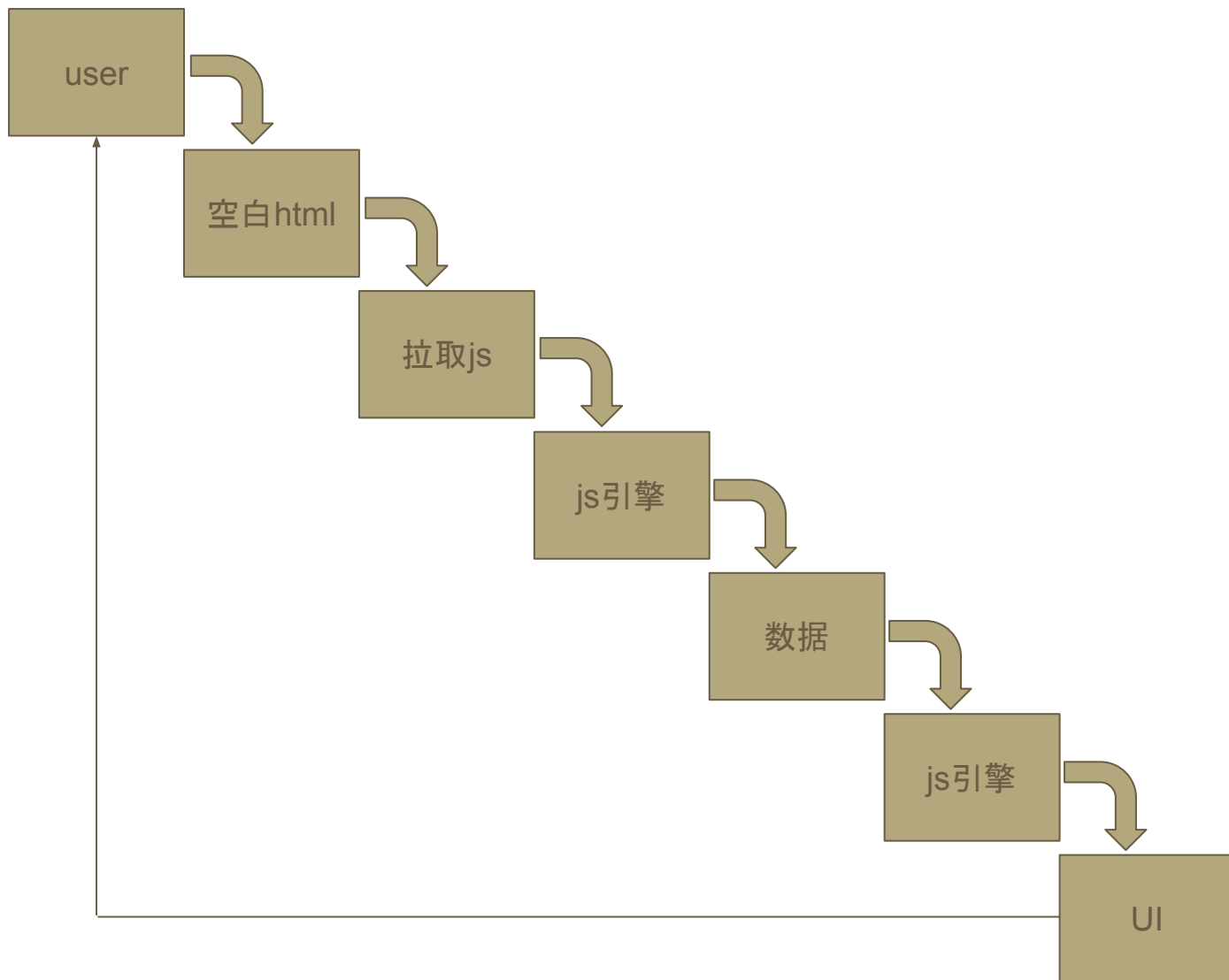
企鹅辅导

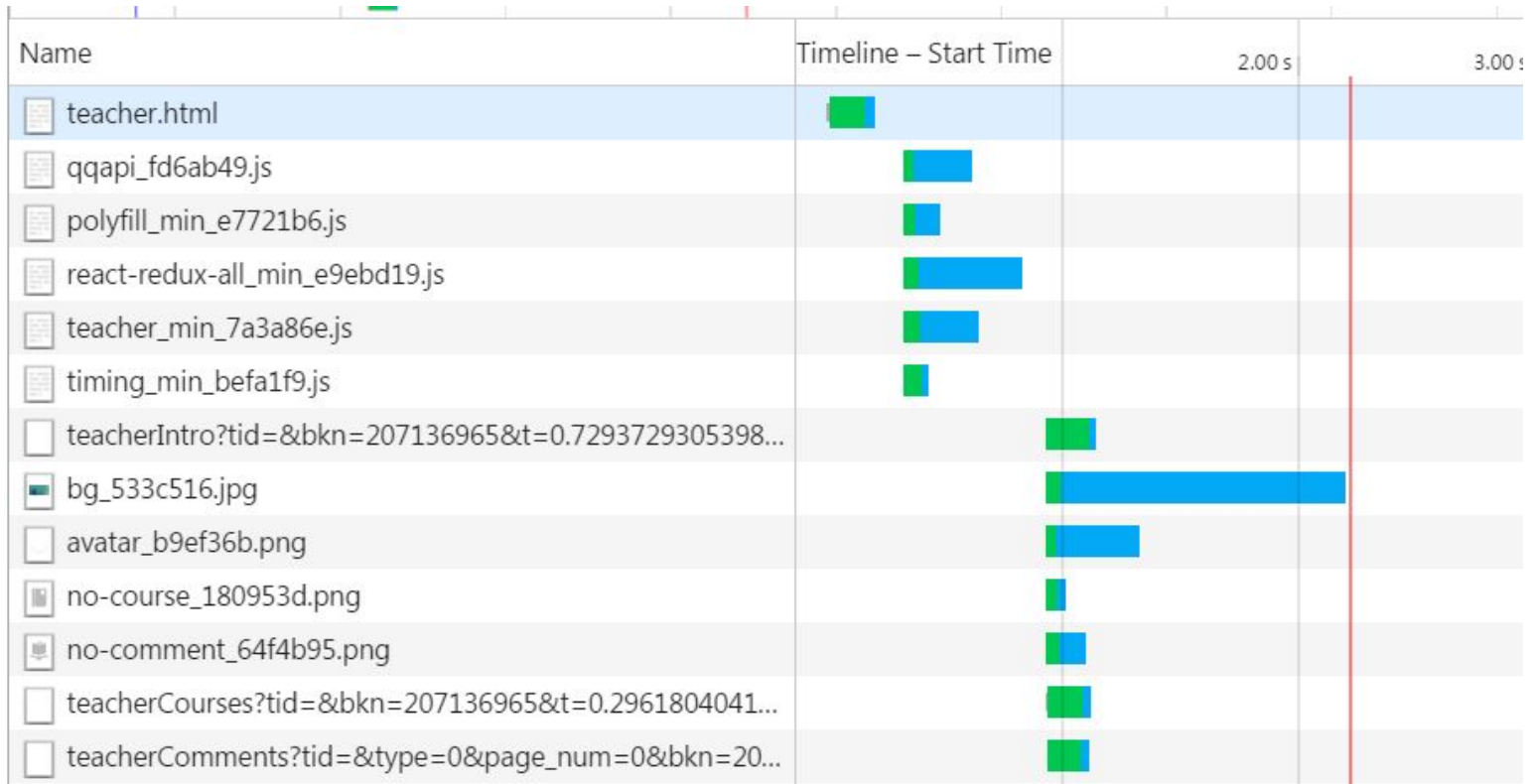
- Hybrid APP ( $\approx 100\%$  离线包)
- 手Q ( $\approx 30\%$  离线包)
- 需要内容尽快展示
- SEO
- .....

# React + Redux

- 数据 $\rightleftharpoons$ UI解耦
- 架构清晰
- 维护方便
- 组件很容易前后端复用

# 最初做法

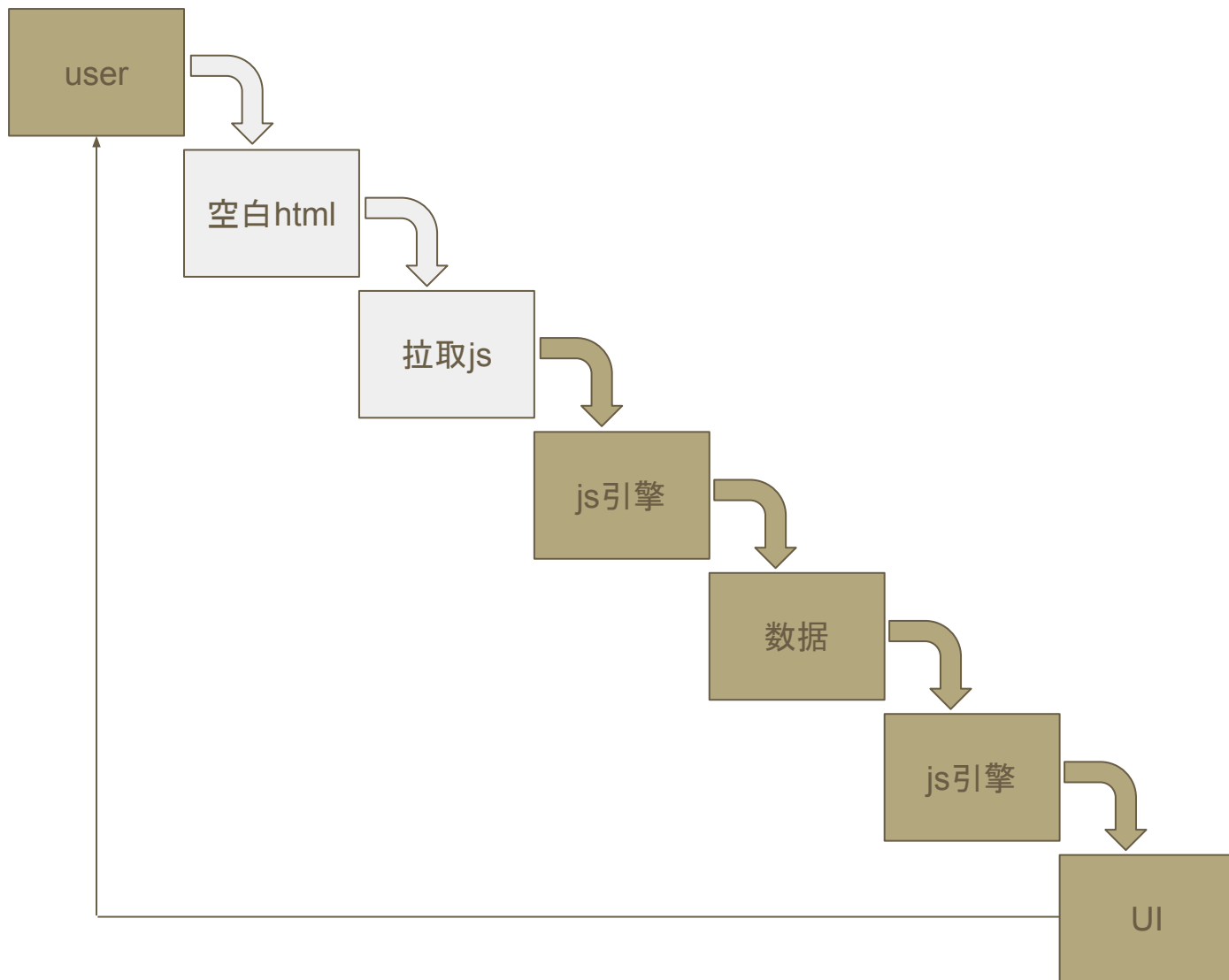








# 离线包



# React引擎初始化

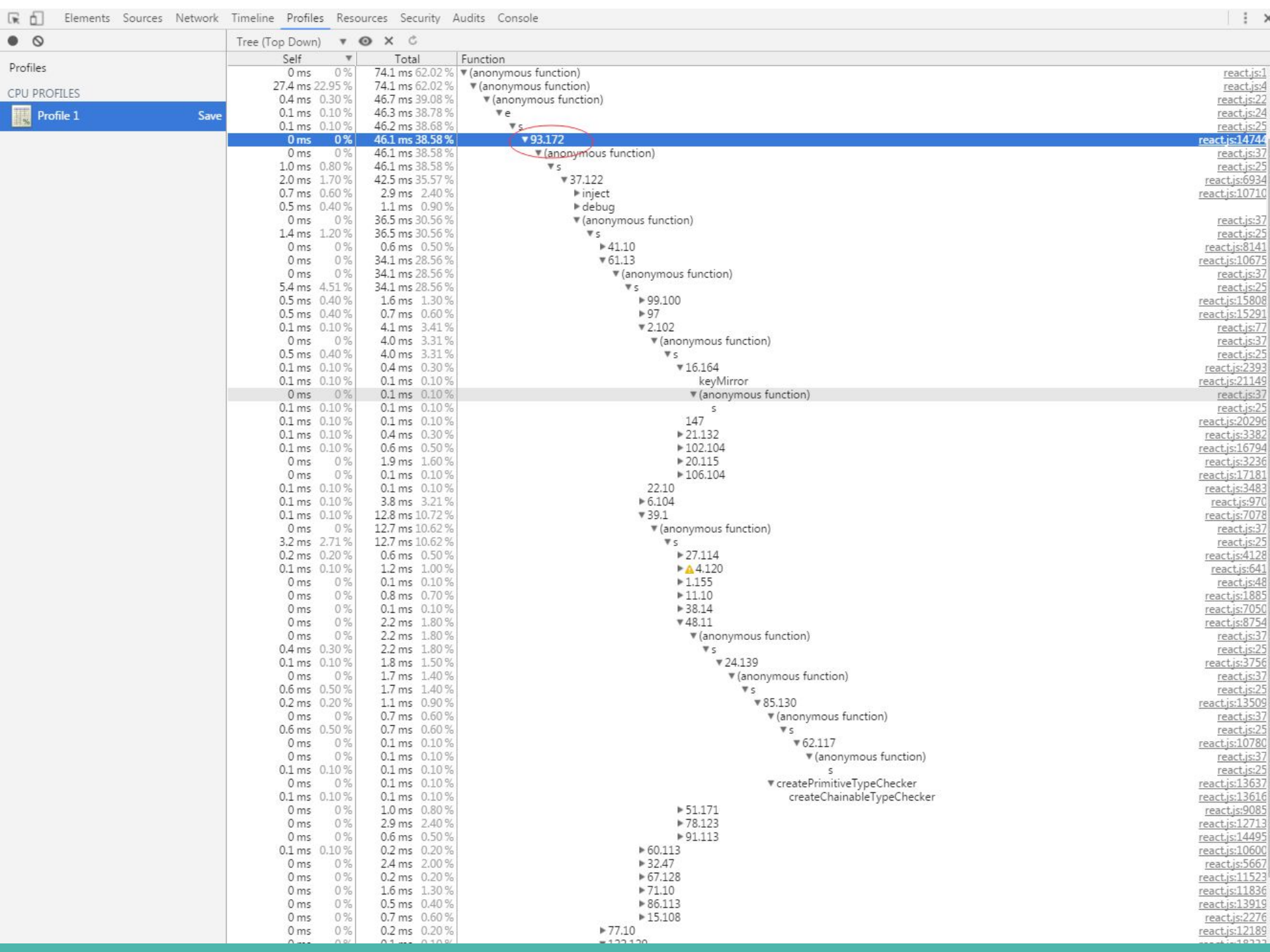
```
<body>

  <script>var T = Date.now();</script>

  <script src="react.js?__inline"></script>

  <script>
    var cost = (Date.now() - T);
  </script>

</body>
```



```
93: [function(_deref_, module, exports) {
  /**
   * Copyright 2013-present, Facebook, Inc.
   * All rights reserved.
   *
   * This source code is licensed under the BSD-style license found in the
   * LICENSE file in the root directory of this source tree. An additional grant
   * of patent rights can be found in the PATENTS file in the same directory.
   *
   * @providesModule ReactUMDEntry
   */

  'use strict';

  var _assign = _deref_(172);

  var ReactDOM = _deref_(37);
  var ReactDOMServer = _deref_(54);
  var React = _deref_(26);
```

react.js:1  
react.js:4  
react.js:22  
react.js:24  
react.js:25  
react.js:14744  
react.js:37  
react.js:25  
react.js:6934  
react.js:10710  
  
react.js:3  
react.js:25  
react.js:8141  
react.js:10675  
react.js:37  
react.js:25  
react.js:15808  
react.js:15291  
react.js:77  
react.js:37  
react.js:25  
react.js:2393  
react.js:21149  
react.js:37  
react.js:25  
react.js:20296  
react.js:3382  
react.js:16794  
react.js:3236  
react.js:17181  
react.js:3483  
react.js:970  
react.js:7078  
react.js:37  
react.js:25  
react.js:4128  
react.js:641  
react.js:48  
react.js:1885  
react.js:7050  
react.js:8754  
react.js:37  
react.js:25  
react.js:3756  
react.js:37  
react.js:25  
react.js:13509  
react.js:37  
react.js:25  
react.js:10780  
react.js:37  
react.js:25  
react.js:13637  
react.js:13616  
react.js:90618  
react.js:12713  
react.js:14495  
react.js:10600  
react.js:5667  
react.js:11523  
react.js:11836  
react.js:13919  
react.js:2276  
react.js:12189

# React引擎初始化

ements Sources Network Timeline Profiles Resources Security Audits Conso				
Tree (Top Down) ▼ 🔍 ✕ ↺				
Self ▼		Total		Function
88.6 ms		1188.6 ms		(idle)
38.7 ms	32.36 %	38.7 ms	32.36 %	(program)
5.6 ms	4.71 %	5.6 ms	4.71 %	(garbage collector)
0.2 ms	0.20 %	0.2 ms	0.20 %	(anonymous function)
0 ms	0 %	0.6 ms	0.50 %	▶ (anonymous function)
0 ms	0 %	74.1 ms	62.02 %	▼ (anonymous function)
27.4 ms	22.95 %	74.1 ms	62.02 %	▼ (anonymous function)
0.4 ms	0.30 %	46.7 ms	39.08 %	▼ (anonymous function)
0.1 ms	0.10 %	46.3 ms	38.78 %	▼ e
0.1 ms	0.10 %	46.2 ms	38.68 %	▼ s
0 ms	0 %	46.1 ms	38.58 %	▼ 93.172
0 ms	0 %	46.1 ms	38.58 %	▼ (anonymous function)
1.0 ms	0.80 %	46.1 ms	38.58 %	▼ s
2.0 ms	1.70 %	42.5 ms	35.57 %	▶ 37.122
0.1 ms	0.10 %	0.4 ms	0.30 %	▶ 172
0 ms	0 %	0.2 ms	0.20 %	▶ 54.61
0 ms	0 %	2.0 ms	1.70 %	▶ 26.137

# React引擎初始化

Elements Sources Network Timeline Profiles Resources Security Audits Console

Tree (Top Down) ▼ 🔍 ✕ ↺				
Self ▼		Total		Function

88.6 ms		1188.6 ms		(idle)
38.7 ms	32.36 %	38.7 ms	32.36 %	(program)
5.6 ms	4.71 %	5.6 ms	4.71 %	(garbage collector)
0.2 ms	0.20 %	0.2 ms	0.20 %	(anonymous function)
0 ms	0 %	0.6 ms	0.50 %	▶ (anonymous function)
0 ms	0 %	74.1 ms	62.02 %	▼ (anonymous function)
27.4 ms	22.95 %	74.1 ms	62.02 %	▼ (anonymous function)
0.4 ms	0.30 %	46.7 ms	39.08 %	▼ (anonymous function)
0.1 ms	0.10 %	46.3 ms	38.78 %	▼ e

0.1 ms	6934
0 ms	6935
0 ms	6936
1.0 ms	6937
2.0 ms	6938
0.1 ms	6939
0 ms	6940
0 ms	6941
	6942
	6943
	6944
	6945
	6946
	6947
	6948
	6949
	6950

```
37: [function(_dereq_, module, exports) {  
  /**  
   * Copyright 2013-present, Facebook, Inc.  
   * All rights reserved.  
   *  
   * This source code is licensed under the BSD-style license found in the  
   * LICENSE file in the root directory of this source tree. An additional gr  
   * of patent rights can be found in the PATENTS file in the same directory.  
   *  
   * @providesModule ReactDOM  
  */  
  
  /* globals __REACT_DEVTOOLS_GLOBAL_HOOK__*/  
  
  'use strict';  
  
  var ReactDOMComponentTree = _dereq_(41);
```



# React引擎初始化

Elements Sources Network Timeline Profiles Resources Security Audits Console									
Tree (Top Down) ▼ 🔍 ✕ ↺									
Self ▼		Total		Function					
1188.6 ms		1188.6 ms		(idle)					
38.7 ms	32.36 %	38.7 ms	32.36 %	(program)					
5.6 ms	4.71 %	5.6 ms	4.71 %	(garbage collector)					
0.2 ms	0.20 %	0.2 ms	0.20 %	(anonymous function)					
0 ms	0 %	0.6 ms	0.50 %	▶ (anonymous function)					
0 ms	0 %	74.1 ms	62.02 %	▼ (anonymous function)					
27.4 ms	22.95 %	74.1 ms	62.02 %	▼ (anonymous function)					
0.4 ms	0.30 %	46.7 ms	39.08 %	▼ (anonymous function)					
0.1 ms	0.10 %	46.3 ms	38.78 %	▼ e					
0.1 ms	0.10 %	46.2 ms	38.68 %	▼ s					
0 ms	0 %	46.1 ms	38.58 %	▼ 93.172					
0 ms	0 %	46.1 ms	38.58 %	▼ (anonymous function)					
1.0 ms	0.80 %	46.1 ms	38.58 %	▼ s					
2.0 ms	1.70 %	42.5 ms	35.57 %	▼ 37.122					
0.7 ms	0.60 %	2.9 ms	2.40 %	▶ inject					
0.5 ms	0.40 %	1.1 ms	0.90 %	▶ debug					
0 ms	0 %	36.5 ms	30.56 %	▼ (anonymous function)					
1.4 ms	1.20 %	36.5 ms	30.56 %	▼ s					
0 ms	0 %	0.6 ms	0.50 %	▶ 41.10					
0 ms	0 %	34.1 ms	28.56 %	▶ 61.13					
0 ms	0 %	0.2 ms	0.20 %	▶ 77.10					
0 ms	0 %	0.1 ms	0.10 %	▶ 122.129					
0.1 ms	0.10 %	0.4 ms	0.30 %	▶ 172					
0 ms	0 %	0.2 ms	0.20 %	▶ 54.61					
0 ms	0 %	2.0 ms	1.70 %	▶ 26.137					

# React引擎初始化

Elements Sources Network Timeline Profiles Resources Security Audits Console

Tree (Top Down) ▼ 🔍 ✕ ↺

Self ▼	Total	Function
1188.6 ms	1188.6 ms	(idle)
38.7 ms 32.36 %	38.7 ms 32.36 %	(program)
5.6 ms 4.71 %	5.6 ms 4.71 %	(garbage collector)
0.2 ms 0.20 %	0.2 ms 0.20 %	(anonymous function)
0 ms 0 %	0.6 ms 0.50 %	▶ (anonymous function)
0 ms 0 %	74.1 ms 62.02 %	▼ (anonymous function)
27.4 ms 22.95 %	74.1 ms 62.02 %	▼ (anonymous function)
0.4 ms 0.30 %	46.7 ms 39.08 %	▼ (anonymous function)
0.1 ms 0.10 %	46.3 ms 38.78 %	▼ e
0.1 ms 0.10 %	46.2 ms 38.68 %	▼ s
0 ms 0 %	46.1 ms 38.58 %	▼ 93.172
0 ms 0 %	46.1 ms 38.58 %	▼ (anonymous function)
1.0 ms 0.80 %	46.1 ms 38.58 %	▼ s
2.0 ms 1.70 %	42.5 10676	61: [function(_dereq_, module, exports) {
0.7 ms 0.60 %	2.9 10677	/**
0.5 ms 0.40 %	1.1 10678	* Copyright 2013-present, Facebook, Inc.
0 ms 0 %	36.5 10679	* All rights reserved.
1.4 ms 1.20 %	36.5 10680	*
0 ms 0 %	0.6 10681	* This source code is licensed under the BSD-style license found in the
0 ms 0 %	34.1 10682	* LICENSE file in the root directory of this source tree. An additional grant
0 ms 0 %	0.2 10683	* of patent rights can be found in the PATENTS file in the same directory.
0 ms 0 %	0.1 10684	*
0.1 ms 0.10 %	0.4 10685	* @providesModule ReactDefaultInjection
0 ms 0 %	0.2 10686	*/
0 ms 0 %	2.0 10687	
	10688	'use strict';
	10689	
	10690	var BeforeInputEventPlugin = _dereq_(2);
	10691	var ChangeEventPlugin = _dereq_(6);
	10692	var DefaultEventPluginOrder = _dereq_(13);
	10693	var EnterLeaveEventPlugin = _dereq_(15);



# React引擎初始化

Elements	Sources	Network	Timeline	Profiles	Resources	Security	Audits	Console
Tree (Top Down) ▼ 🔍 ✕ ↺								
	Self		Total		Function			
	3.8 ms	4.71 %	3.8 ms	4.71 %	(garbage collector)			
	0.2 ms	0.20 %	0.2 ms	0.20 %	(anonymous function)			
	0 ms	0 %	0.6 ms	0.50 %	▶ (anonymous function)			
	0 ms	0 %	74.1 ms	62.02 %	▼ (anonymous function)			
	27.4 ms	22.95 %	74.1 ms	62.02 %	▼ (anonymous function)			
	0.4 ms	0.30 %	46.7 ms	39.08 %	▼ (anonymous function)			
	0.1 ms	0.10 %	46.3 ms	38.78 %	▼ e			
	0.1 ms	0.10 %	46.2 ms	38.68 %	▼ s			
	0 ms	0 %	46.1 ms	38.58 %	▼ 93.172			
	0 ms	0 %	46.1 ms	38.58 %	▼ (anonymous function)			
	1.0 ms	0.80 %	46.1 ms	38.58 %	▼ s			
	2.0 ms	1.70 %	42.5 ms	35.57 %	▼ 37.122			
	0.7 ms	0.60 %	2.9 ms	2.40 %	▶ inject			
	0.5 ms	0.40 %	1.1 ms	0.90 %	▶ debug			
	0 ms	0 %	36.5 ms	30.56 %	▼ (anonymous function)			
	1.4 ms	1.20 %	36.5 ms	30.56 %	▼ s			
	0 ms	0 %	0.6 ms	0.50 %	▶ 41.10			
	0 ms	0 %	34.1 ms	28.56 %	▼ 61.13			
	0 ms	0 %	34.1 ms	28.56 %	▼ (anonymous function)			
	5.4 ms	4.51 %	34.1 ms	28.56 %	▼ s			
	0.5 ms	0.40 %	1.6 ms	1.30 %	▶ 99.100			
	0.5 ms	0.40 %	0.7 ms	0.60 %	▶ 97			
	0.1 ms	0.10 %	4.1 ms	3.41 %	▶ 2.102			
	0.1 ms	0.10 %	0.1 ms	0.10 %	22.10			
	0.1 ms	0.10 %	3.8 ms	3.21 %	▶ 6.104			
	0.1 ms	0.10 %	12.8 ms	10.72 %	▶ 39.1			
	0.1 ms	0.10 %	0.2 ms	0.20 %	▶ 60.113			
	0 ms	0 %	2.4 ms	2.00 %	▶ 32.47			
	0 ms	0 %	0.2 ms	0.20 %	▶ 67.128			
	0 ms	0 %	1.6 ms	1.30 %	▶ 71.10			

# React引擎初始化

Elements Sources Network Timeline Profiles Resources Security Audits Console

Tree (Top Down) ▼ 🔍 ✕ ↺

	Self	Total	Function
	3.8 ms 4.71 %	3.8 ms 4.71 %	(garbage collector)
	0.2 ms 0.20 %	0.2 ms 0.20 %	(anonymous function)
	0 ms 0 %	0.6 ms 0.50 %	▶ (anonymous function)
	0 ms 0 %	74.1 ms 62.02 %	▼ (anonymous function)
	27.4 ms 22.95 %	74.1 ms 62.02 %	▼ (anonymous function)
	0.4 ms 0.30 %	46.7 ms 39.08 %	▼ (anonymous function)
	0.1 ms 0.10 %	46.3 ms 38.78 %	▼ e
	0.1 ms 0.10 %	46.2 ms 38.68 %	▼ s
	0 ms 0 %	46.1 ms 38.58 %	▼ 93.172
	0 ms 0 %	46.1 ms 38.58 %	▼ (anonymous function)
	1.0 ms 0.80 %	46.1 ms 38.58 %	▼ s
	2.0 ms 1.70 %	42.5 ms 35.28 %	-----
	0.7 ms 0.60 %	2.9 7079	39: [function(_dereq_, module, exports) {
	0.5 ms 0.40 %	1.1 7080	/**
	0 ms 0 %	36.5 7081	* Copyright 2013-present, Facebook, Inc.
	1.4 ms 1.20 %	36.5 7082	* All rights reserved.
	0 ms 0 %	0.6 7083	*
	0 ms 0 %	34.1 7084	* This source code is licensed under the BSD-style license found in the
	0 ms 0 %	34.1 7085	* LICENSE file in the root directory of this source tree. An additional grant
	5.4 ms 4.51 %	34.1 7086	* of patent rights can be found in the PATENTS file in the same directory.
	0.5 ms 0.40 %	1.6 7087	*
	0.5 ms 0.40 %	0.7 7088	* @providesModule ReactDOMComponent
	0.1 ms 0.10 %	4.1 7089	*/
	0.1 ms 0.10 %	0.1 7090	
	0.1 ms 0.10 %	3.8 7091	/* global hasOwnProperty:true */
	0.1 ms 0.10 %	0.2 7092	
	0 ms 0 %	2.4 7093	'use strict';
	0 ms 0 %	0.2 7094	
	0 ms 0 %	1.6 7095	var _prodInvariant = _dereq_(139),
		7096	_assign = _dereq_(172);
		7097	
		7098	var AutoFocusUtils = _dereq_(1);

# 现实更严峻

```
<body>

  <script>var T = Date.now();</script>

  <script src="qqapi.js"></script>
  <script src="react.js"></script>
  <script src="react-dom.js"></script>
  <script src="redux.js"></script>
  <script src="react-redux.js"></script>
  <script src="redux-thunk.js"></script>
  <script src="react-fastclick.js"></script>

  <script>
    var cost = (Date.now() - T);
  </script>

</body>
```

# 现实更严峻

实测平均近400+ms!

```
<body>

  <script>var T = Date.now();</script>

  <script src="qqapi.js"></script>
  <script src="react.js"></script>
  <script src="react-dom.js"></script>
  <script src="redux.js"></script>
  <script src="react-redux.js"></script>
  <script src="redux-thunk.js"></script>
  <script src="react-fastclick.js"></script>

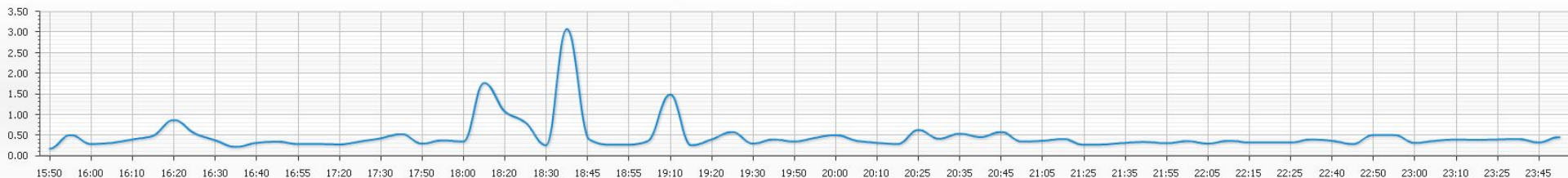
  <script>
    var cost = (Date.now() - T);
  </script>

</body>
```

# 另外的时间消耗

某ajax请求耗时曲线

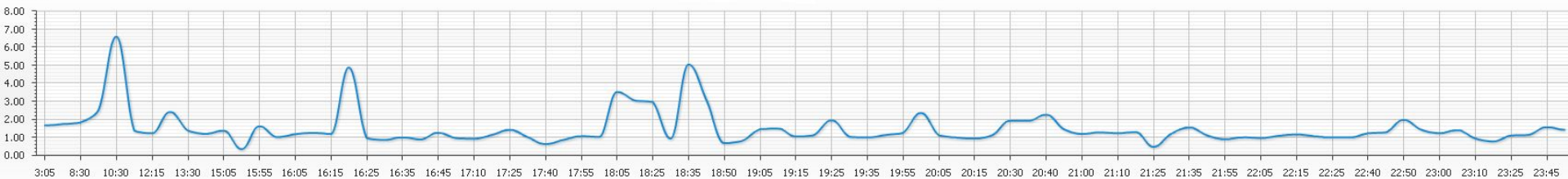
单位:秒



# 首屏统计

某页面首屏时间  $\approx 1.2\text{s}$

单位: 秒

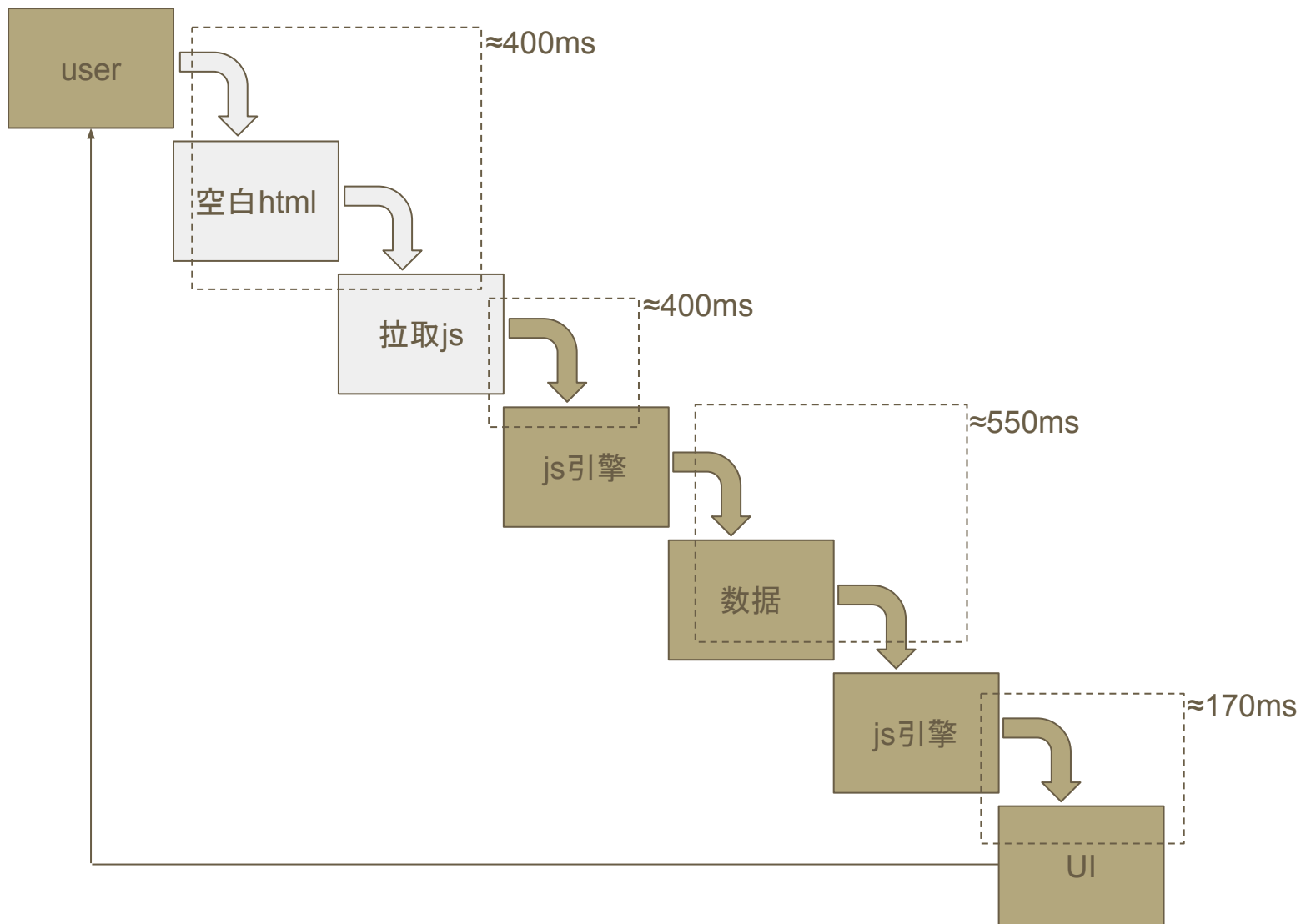


# 时间都去哪了

React&Redux + qqapi

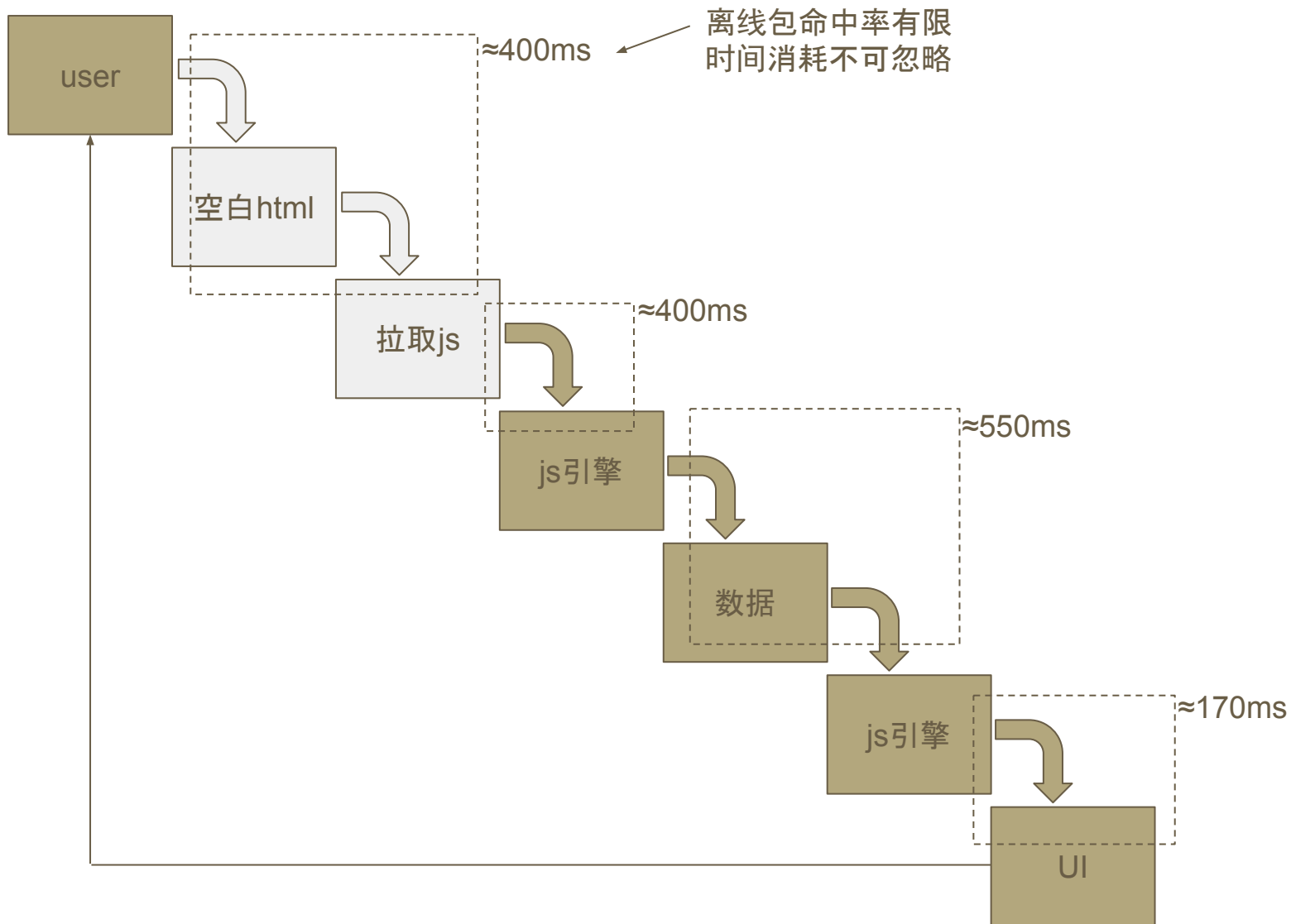
- 体积大 (gzip: 50.6+14.4kb)
  - 基础框架初始化慢 (400+ms)
  - ajax请求耗时 ( $\approx 550$ ms)
- 
- 秒开几乎不可能

# 时间都去哪了

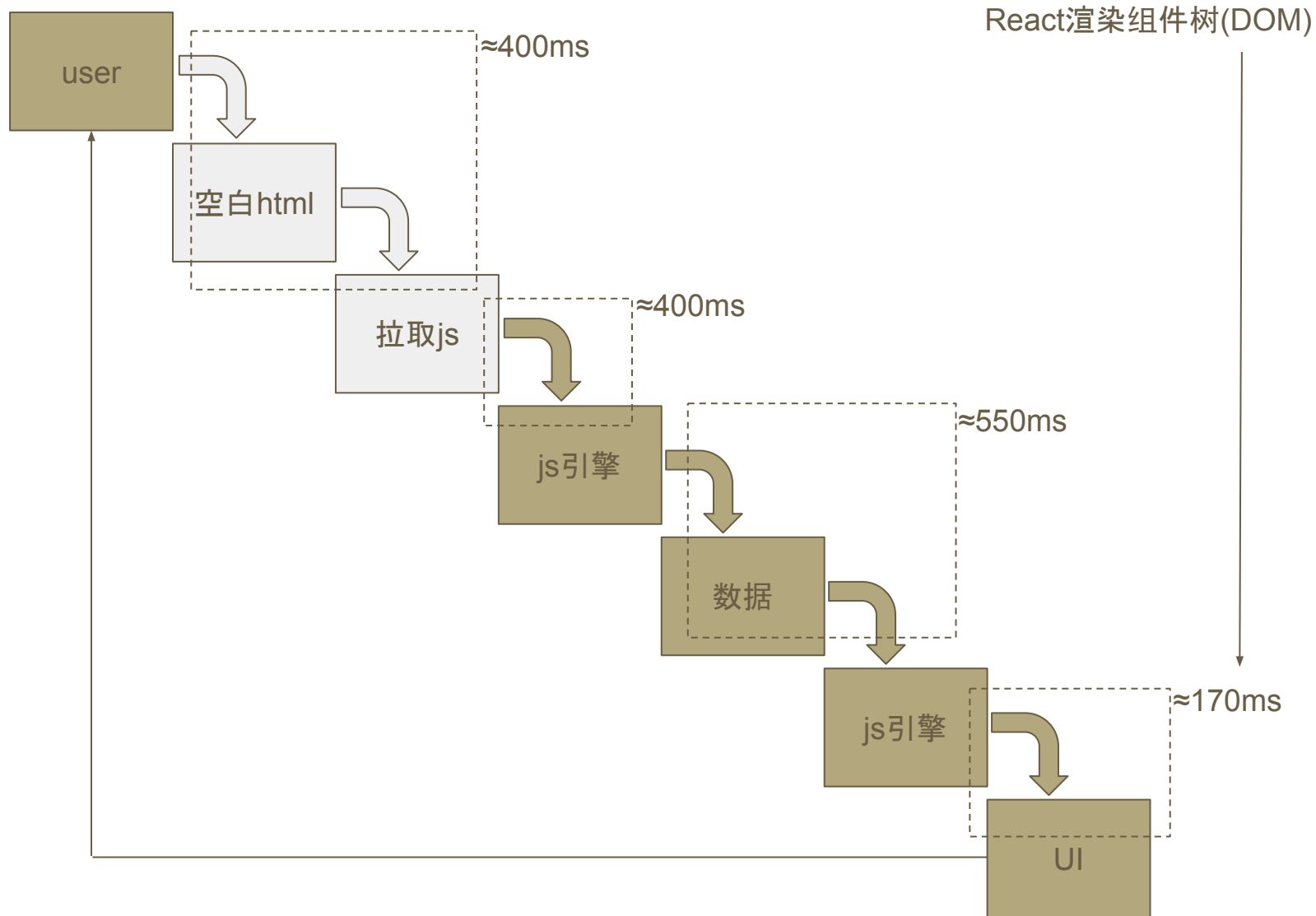







# 时间都去哪了



# React真的很快吗？



Tree (Top Down)   

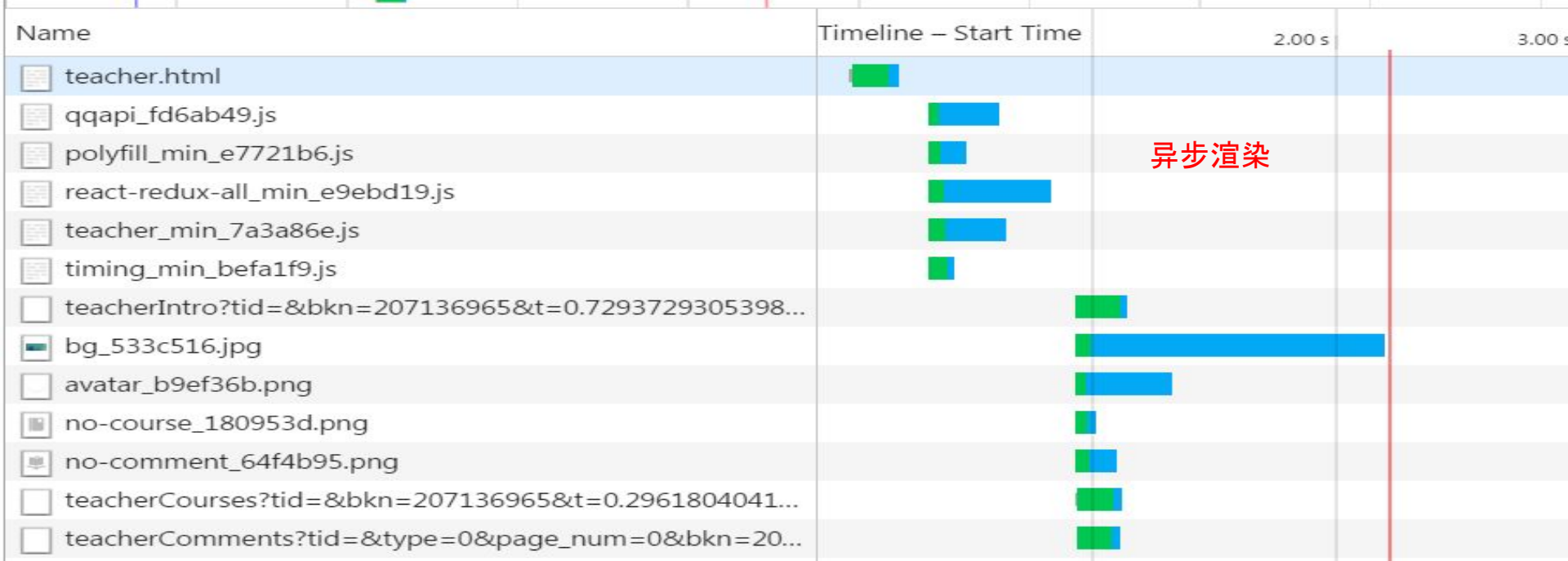
Self ▼		Total	Function
0 ms	0 %	59.3 ms	16.04 %
0.1 ms	0.04 %	59.3 ms	16.04 %
0.3 ms	0.07 %	59.0 ms	15.96 %
0.1 ms	0.04 %	58.6 ms	15.86 %
0 ms	0 %	58.5 ms	15.82 %
0.1 ms	0.04 %	58.5 ms	15.82 %
0 ms	0 %	52.6 ms	14.23 %
0.3 ms	0.07 %	52.5 ms	14.20 %
0.3 ms	0.07 %	0.3 ms	0.07 %
0.1 ms	0.04 %	19.5 ms	5.26 %
0.1 ms	0.04 %	0.1 ms	0.04 %
0.1 ms	0.04 %	18.3 ms	4.94 %
0.3 ms	0.07 %	18.1 ms	4.90 %
0.1 ms	0.04 %	17.9 ms	4.83 %
0.1 ms	0.04 %	0.1 ms	0.04 %
0 ms	0 %	17.6 ms	4.76 %
0 ms	0 %	17.6 ms	4.76 %
0 ms	0 %	0.3 ms	0.07 %
0 ms	0 %	17.3 ms	4.68 %
0 ms	0 %	0.1 ms	0.04 %
0 ms	0 %	17.2 ms	4.65 %
0.3 ms	0.07 %	17.2 ms	4.65 %
0 ms	0 %	0.3 ms	0.07 %
0 ms	0 %	16.7 ms	4.50 %
0 ms	0 %	0.9 ms	0.25 %
0 ms	0 %	15.7 ms	4.25 %
0.1 ms	0.04 %	15.7 ms	4.25 %
0 ms	0 %	0.1 ms	0.04 %
0 ms	0 %	15.5 ms	4.18 %
0.4 ms	0.11 %	14.9 ms	4.04 %
0.5 ms	0.14 %	14.5 ms	3.93 %
0 ms	0 %	0.1 ms	0.04 %
0 ms	0 %	13.9 ms	3.75 %
0 ms	0 %	13.9 ms	3.75 %
0 ms	0 %	0.4 ms	0.11 %
0 ms	0 %	13.5 ms	3.64 %



render:  
 组件树从上往下  
 一层层依次渲染/更新

# 解决问题

- 第一个请求中就返回首屏
- 耗时的操作放到首屏展示后
  - 初始JS
  - 拉取非首屏数据
  - 渲染非首屏



# 服务端React

服务端React

服务端React

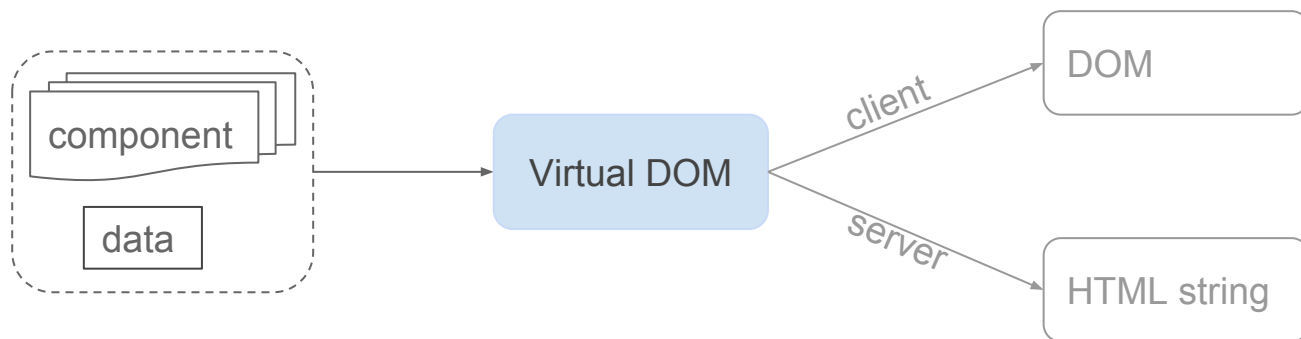
服务端React

服务端React

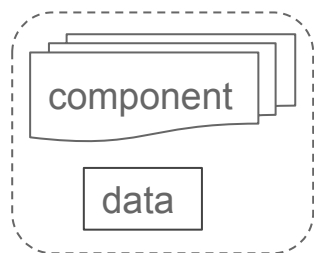
服务端React

服务端React

# 服务端React



# 服务端React



Virtual DOM

client

DOM

server

HTML string

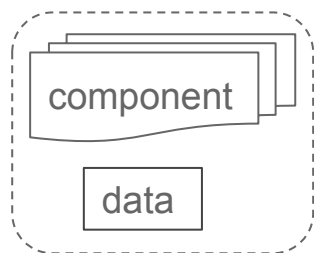
```
// client

import React from 'react';
import ReactDOM from 'react-dom';
import { Provider } from 'react-redux';

ReactDOM.render(
  (
    <Provider store={store}>
      <App />
    </Provider>
  ), //
  document.querySelector('#root')
);
```



# 服务端React



Virtual DOM

client

DOM

server

HTML string

```
// client
```

```
import React from 'react';
import ReactDOM from 'react-dom';
import { Provider } from 'react-redux';
```

```
ReactDOM.render(
  (
    <Provider store={store}>
      <App />
    </Provider>
  ), //
  document.querySelector('#root')
);
```

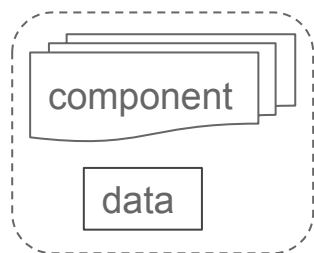
```
// server
```

```
import React from 'react';
import ReactDOM from 'react-dom/server';
import { Provider } from 'react-redux';
```

```
let rootElement = React.createElement(Provider, {
  store: store
}, App);
```

```
let html = ReactDOM.renderToString(rootElement);
```

# 服务端React



Virtual DOM

client

DOM

server

HTML string

```
// client

import React from 'react';
import ReactDOM from 'react-dom';
import { Provider } from 'react-redux';

ReactDOM.render(
  (
    <Provider store={store}>
      <App />
    </Provider>
  ), //
  document.querySelector('#root')
);
```

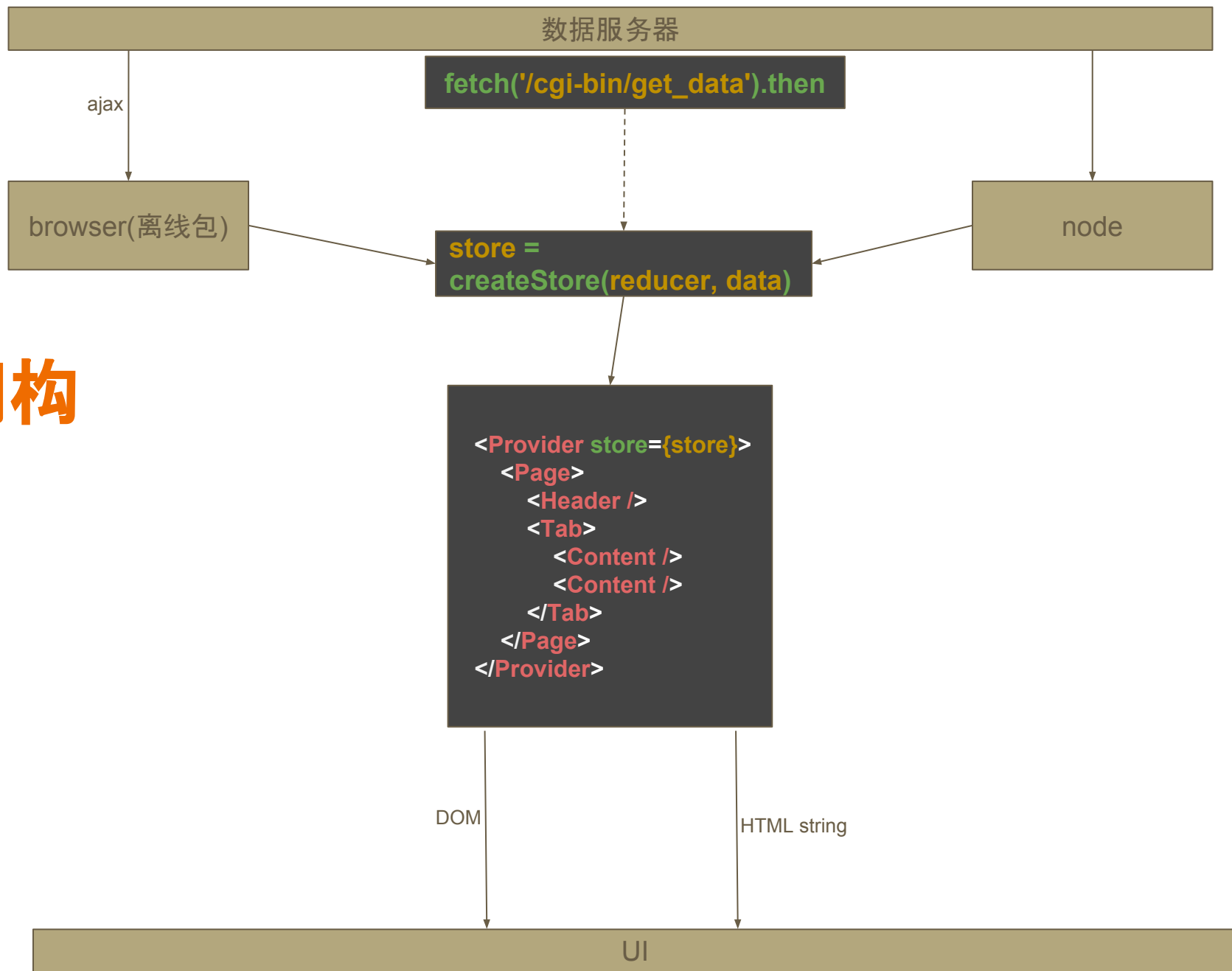
```
// server

import React from 'react';
import ReactDOM from 'react-dom/server';
import { Provider } from 'react-redux';

let rootElement = React.createElement(Provider, {
  store: store
}, App);

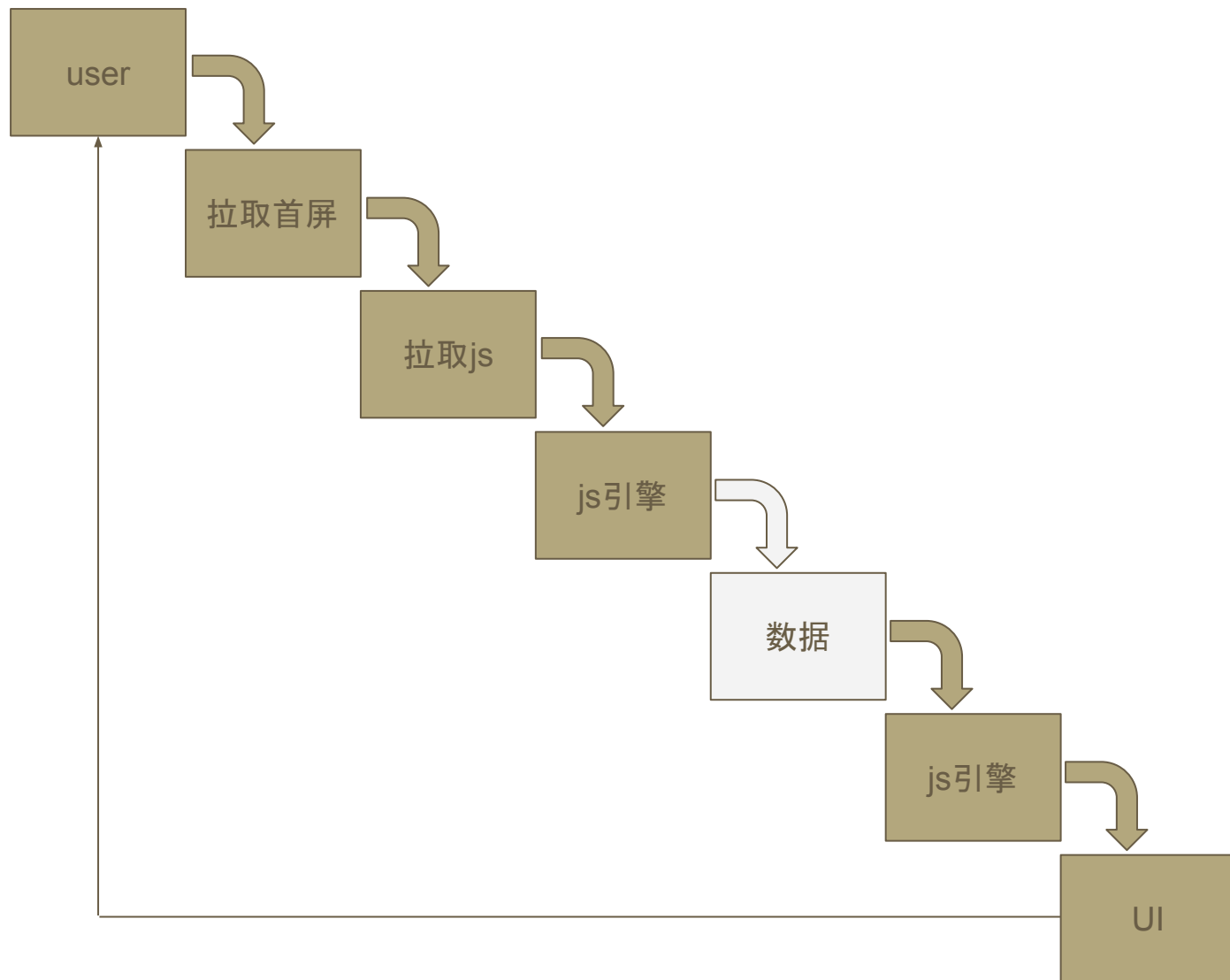
let html = ReactDOM.renderToString(rootElement);
```

# 同构

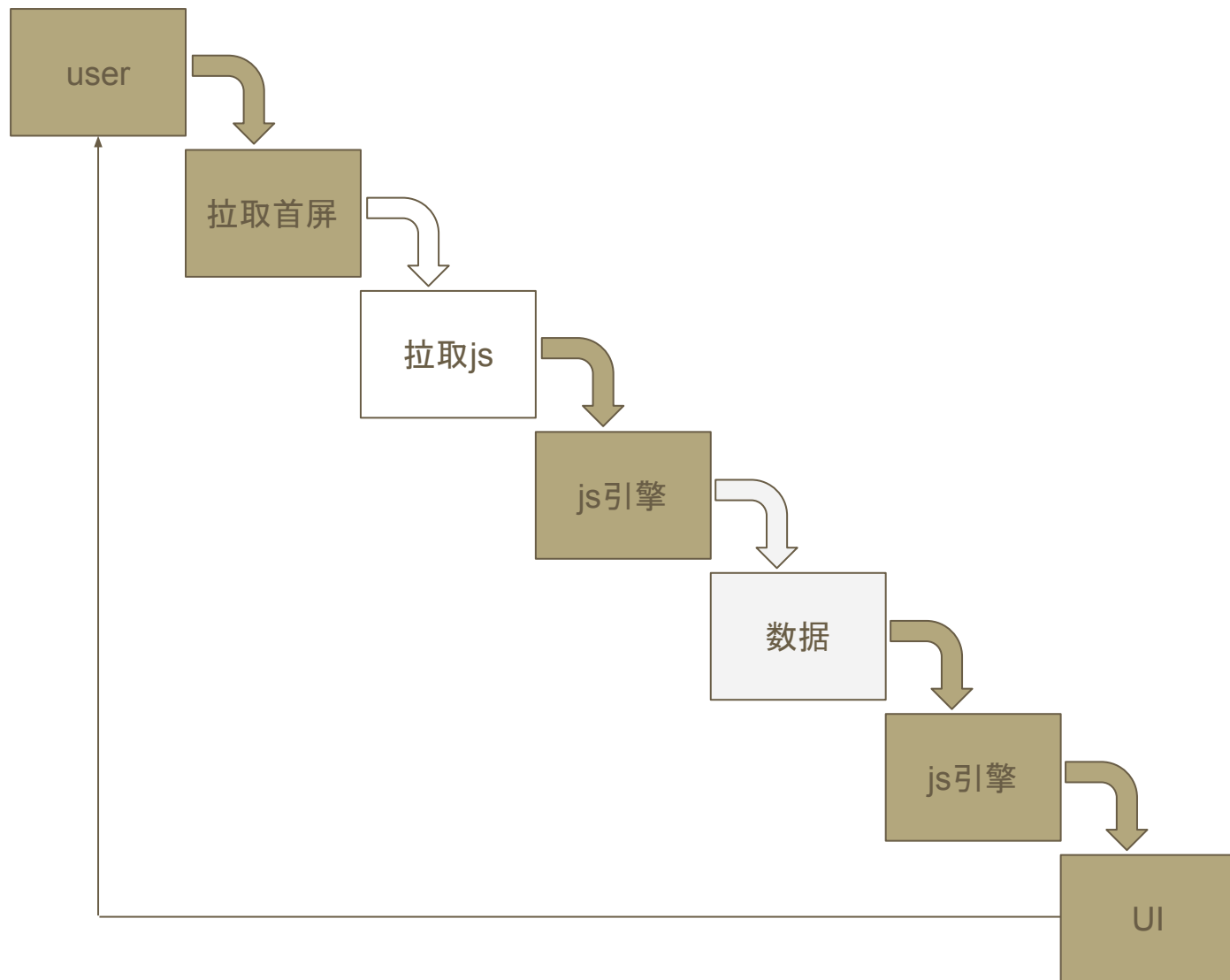




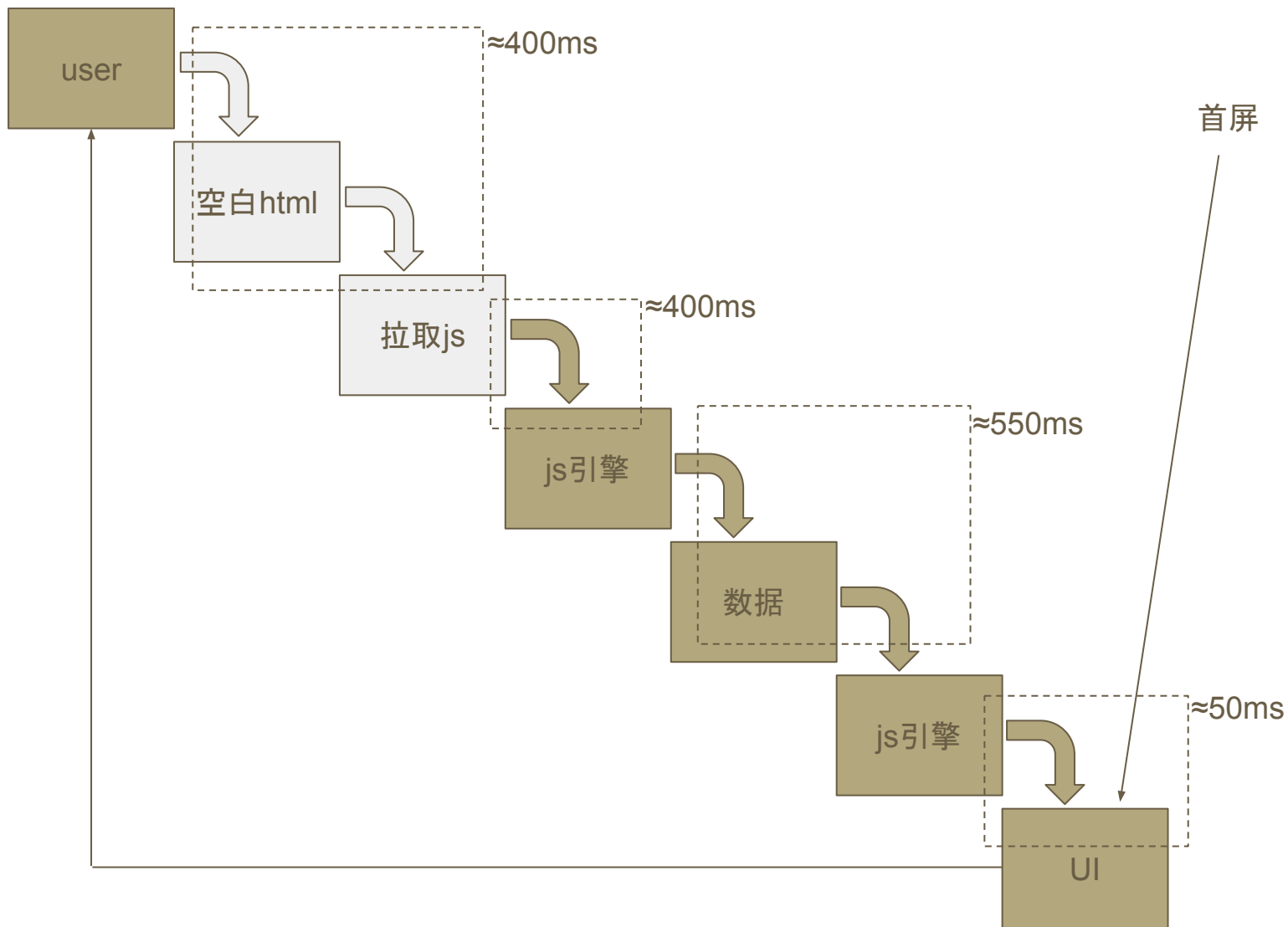
# 服务端渲染



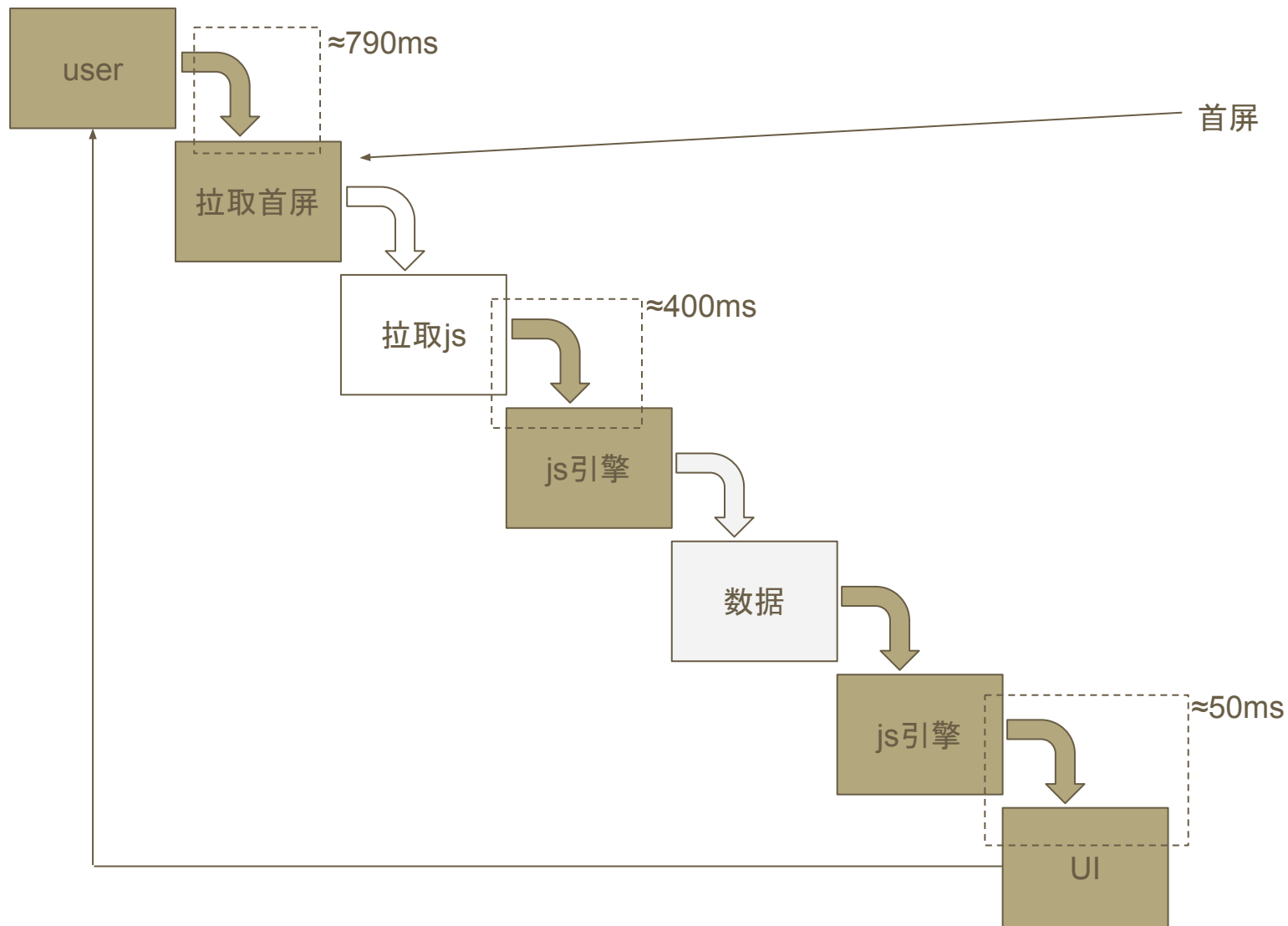
# 服务端 + 离线包



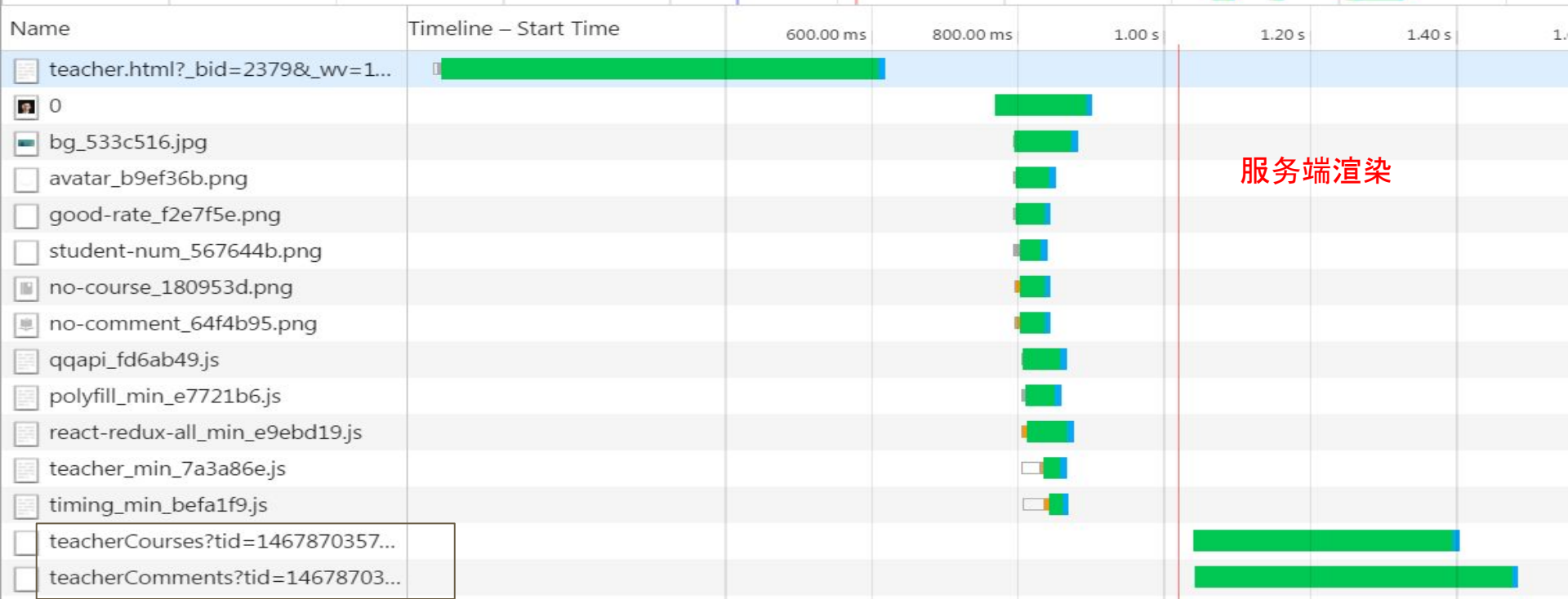
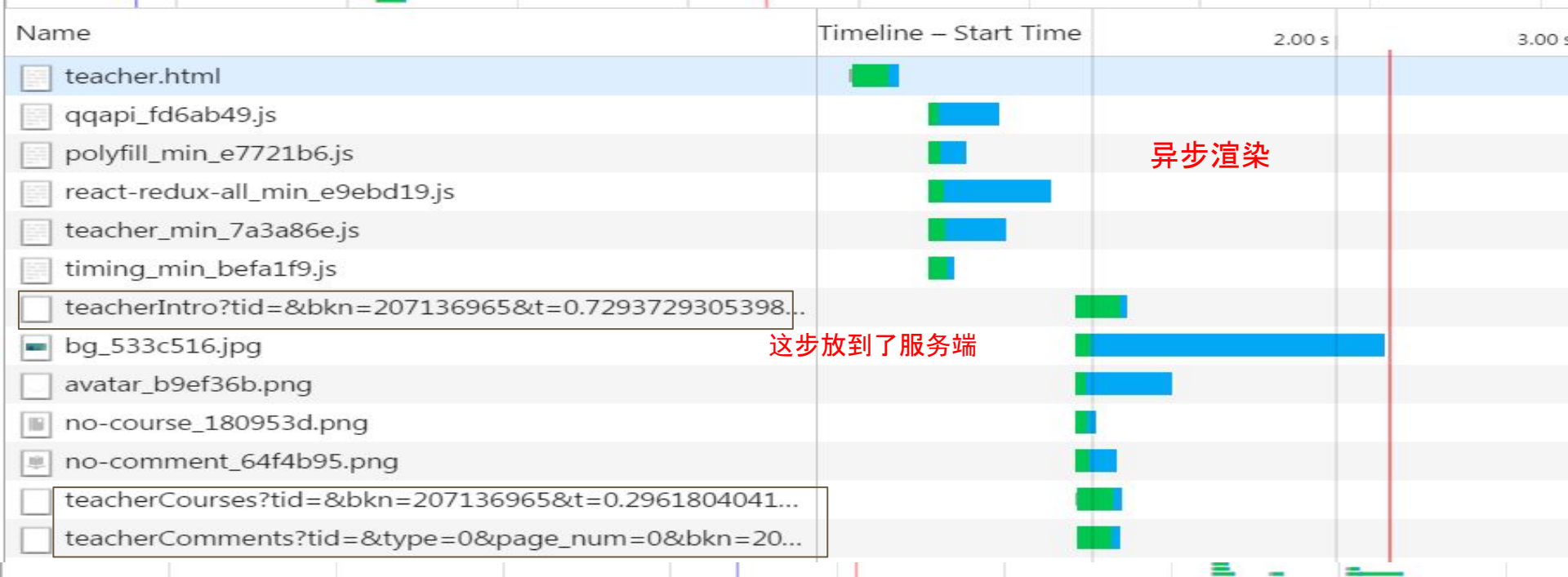
# 离线包&异步渲染



# 服务端 + 离线包

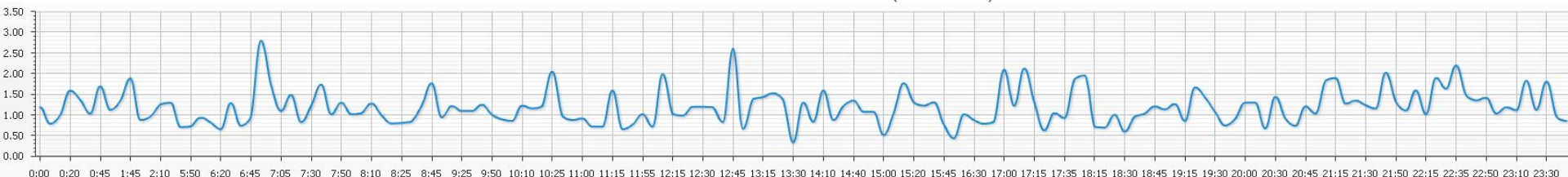




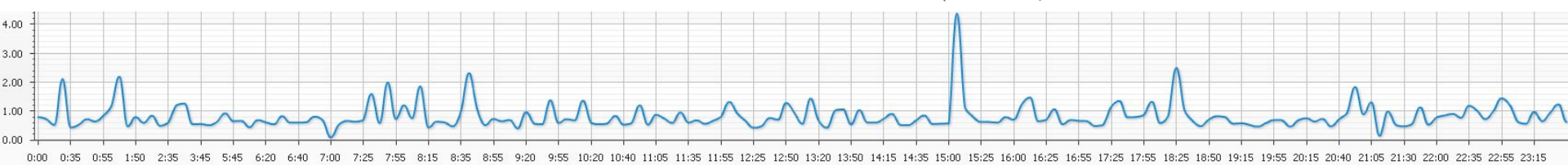


# 效果

首屏时延-离线包 (1.17021s)



首屏时延-服务端渲染 (0.795304s)



# 效果

服务端代码只占了全部业务代码的  $\approx 5\%$

一个需求开发仅需写非常少量的node代码

# 一些Tips

- 服务端只生成首屏
  - 只为了减少白屏时间
  - node拉取数据也消耗时间



简介

课程(4)

评价(151)

### 老师标签

高中英语

10年教龄

青岛最受欢迎的独立教师

### 教学经历

- 2013-至今  
于瑶捷径英语首席讲师
- 2006-2012  
中高考百人班教师,未来支点教育创始人

### 教学内容

首屏



简介

课程(4)

评价(151)

已售216/限额2000

出题天使的“套路” 讲座

8月2日 21:30-22:30

¥1

已售179/限额2000

英语新学法-扎实篇 系统课

8月3日 - 8月11日, 每两天一节, 每晚9点...

¥36

单词语法欢乐通 讲座

¥1



简介

课程(4)

评价(151)

全部(151) 好评(119) 中评(13) 差评(19)

4\*\*\*  
7-27 23:38



人挺漂亮, 课也讲的好

《英语新学法-导入篇》

愿\*\*\*  
7-29 23:39



老师讲的棒棒哒

《英语新学法-基础篇》

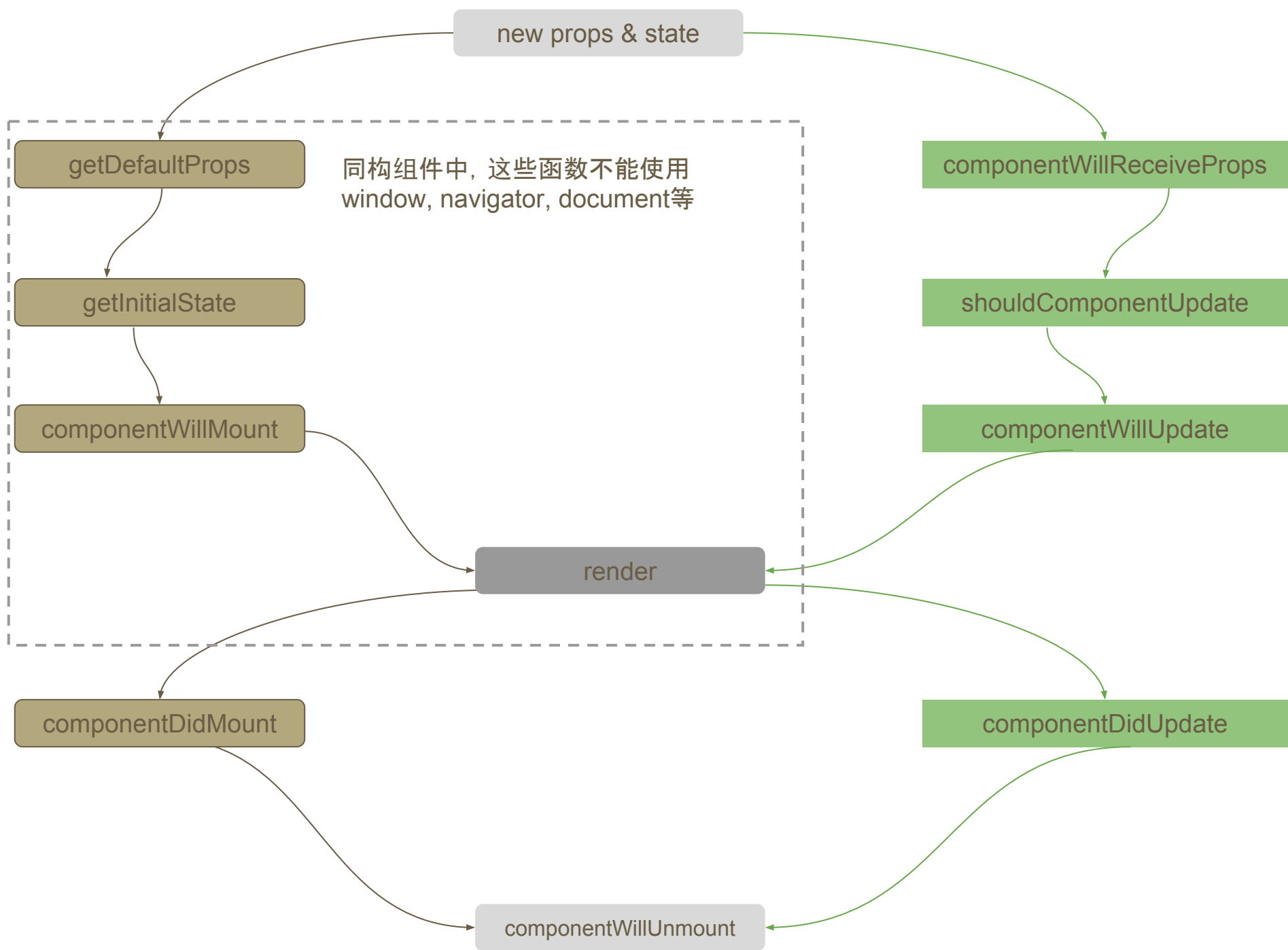
洋\*\*\*  
7-29 08:05



非常满意

# 一些Tips

- 服务端只生成首屏
  - 只为了减少白屏时间
  - node拉取数据也消耗时间
- 客户端API 要谨慎



# 新的挑战



# 新的挑战

```
{  
  "cost": [195, 197, 194, 186, 169, 228, 186, 182, 207, 169, 77, 155, 159, 69, 81, 175,  
    , 104, 69, 71, 82, 68, 193, 93, 141, 74, 177, 74, 69, 95, 104, 156, 201, 69, 69, 9  
    102, 157, 158, 104, 112, 149, 152, 91, 101, 111, 144, 143, 207, 153, 153, 138, 198  
    71, 271, 72, 69, 95, 74, 83, 174, 97, 93, 105, 175, 106, 91, 157, 158, 144, 144, 1  
    108, 134, 146, 186, 108, 81, 109, 155, 141, 133, 68, 151, 142, 107, 185, 72, 185,  
    , 67, 72, 92, 81, 68, 173, 69, 141, 143, 82, 108, 198, 92, 93, 84, 86, 108, 144, 1  
    111, 179, 149, 154, 146, 199, 198, 111, 145, 177, 109, 108, 196, 111, 185, 148, 15  
    186, 143, 174, 153, 190, 139, 192, 176, 144, 186, 143, 187, 145, 185, 156, 179, 13  
    188, 136, 208, 152, 196, 149, 186, 151, 153, 185, 148, 180, 185, 151, 195, 178, 14  
    167, 136, 225, 145, 150, 69, 83, 132, 69, 75, 219, 164, 97, 79, 209, 73, 224, 85,  
    91, 252, 151, 69, 214, 112, 155, 83, 88, 194, 159, 106, 89, 80, 82, 88, 83, 217, 8  
    81, 240, 131, 94, 268, 187, 214, 177, 159, 177, 85, 98, 209, 92, 73, 205, 160, 85,  
    229, 70, 72, 221, 71, 172, 68, 217, 271, 265, 123, 132, 85, 222, 171, 73, 171, 142  
    , 217, 91, 69, 83, 155, 160, 100, 105, 149, 100, 108, 144, 144, 72, 170, 75, 195,  
    104, 250, 68, 179, 140, 136, 92, 145, 143, 193, 72, 179, 177, 207, 212, 175, 151,
```

node端拉取数据耗时log

# 新的挑战

```
{  
  "cost": [195, 197, 194, 186, 169, 228, 186, 182, 207, 169, 77, 155, 159, 69, 81, 175,  
    , 104, 69, 71, 82, 68, 193, 93, 141, 74, 177, 74, 69, 95, 104, 156, 201, 69, 69, 9  
    102, 157, 158, 104, 112, 149, 152, 91, 101, 111, 144, 143, 207, 153, 153, 138, 198  
    71, 271, 72, 69, 95, 74, 83, 174, 97, 93, 105, 175, 106, 91, 157, 158, 144, 144, 1  
    108, 134, 146, 186, 108, 81, 109, 155, 141, 133, 68, 151, 142, 107, 185, 72, 185,  
    , 67, 72, 92, 81, 68, 173, 69, 141, 143, 82, 108, 198, 92, 93, 84, 86, 108, 144, 1  
    111, 179, 149, 154, 146, 199, 198, 111, 145, 177, 109, 108, 196, 111, 185, 148, 15  
    186, 143, 174, 153, 190, 139, 192, 176, 144, 186, 143, 187, 145, 185, 156, 179, 13  
    188, 136, 208, 152, 196, 149, 186, 151, 153, 185, 148, 180, 185, 151, 195, 178, 14  
    167, 136, 225, 145, 150, 69, 83, 132, 69, 75, 219, 164, 97, 79, 209, 73, 224, 85,  
    91, 252, 151, 69, 214, 112, 155, 83, 88, 194, 159, 106, 89, 80, 82, 88, 83, 217, 8  
    81, 240, 131, 94, 268, 187, 214, 177, 159, 177, 85, 98, 209, 92, 73, 205, 160, 85,  
    229, 70, 72, 221, 71, 172, 68, 217, 271, 265, 123, 132, 85, 222, 171, 73, 171, 142  
    , 217, 91, 69, 83, 155, 160, 100, 105, 149, 100, 108, 144, 144, 72, 170, 75, 195,  
    104, 250, 68, 179, 140, 136, 92, 145, 143, 193, 72, 179, 177, 207, 212, 175, 151,
```

平均耗时≈125ms

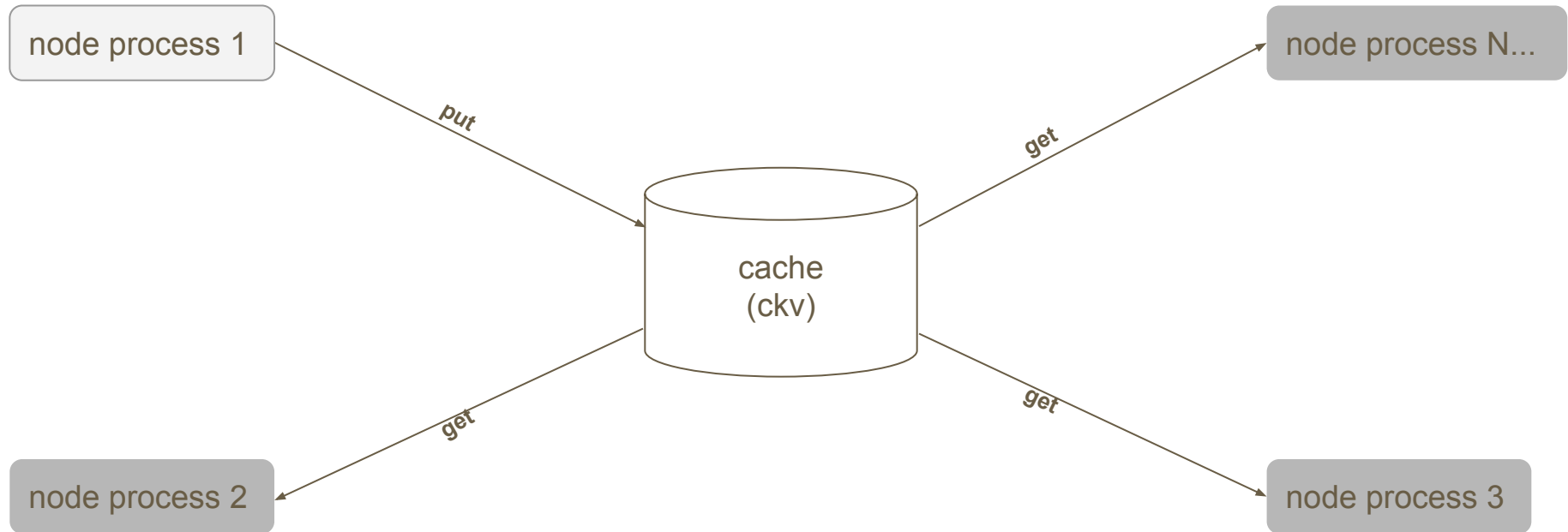
# 新的挑战

```
{  
  "cost": [195, 197, 194, 186, 169, 228, 186, 182, 207, 169, 77, 155, 159, 69, 81, 175,  
    , 104, 69, 71, 82, 68, 193, 93, 141, 74, 177, 74, 69, 95, 104, 156, 201, 69, 69, 9  
    102, 157, 158, 104, 112, 149, 152, 91, 101, 111, 144, 143, 207, 153, 153, 138, 198  
    71, 271, 72, 69, 95, 74, 83, 174, 97, 93, 105, 175, 106, 91, 157, 158, 144, 144, 1  
    108, 134, 146, 186, 108, 81, 109, 155, 141, 133, 68, 151, 142, 107, 185, 72, 185,  
    , 67, 72, 92, 81, 68, 173, 69, 141, 143, 82, 108, 198, 92, 93, 84, 86, 108, 144, 1  
    111, 179, 149, 154, 146, 199, 198, 111, 145, 177, 109, 108, 196, 111, 185, 148, 15  
    186, 143, 174, 153, 190, 139, 192, 176, 144, 186, 143, 187, 145, 185, 156, 179, 13  
    188, 136, 208, 152, 196, 149, 186, 151, 153, 185, 148, 180, 185, 151, 195, 178, 14  
    167, 136, 225, 145, 150, 69, 83, 132, 69, 75, 219, 164, 97, 79, 209, 73, 224, 85,  
    91, 252, 151, 69, 214, 112, 155, 83, 88, 194, 159, 106, 89, 80, 82, 88, 83, 217, 8  
    81, 240, 131, 94, 268, 187, 214, 177, 159, 177, 85, 98, 209, 92, 73, 205, 160, 85,  
    229, 70, 72, 221, 71, 172, 68, 217, 271, 265, 123, 132, 85, 222, 171, 73, 171, 142  
    , 217, 91, 69, 83, 155, 160, 100, 105, 149, 100, 108, 144, 144, 72, 170, 75, 195,  
    104, 250, 68, 179, 140, 136, 92, 145, 143, 193, 72, 179, 177, 207, 212, 175, 151,
```

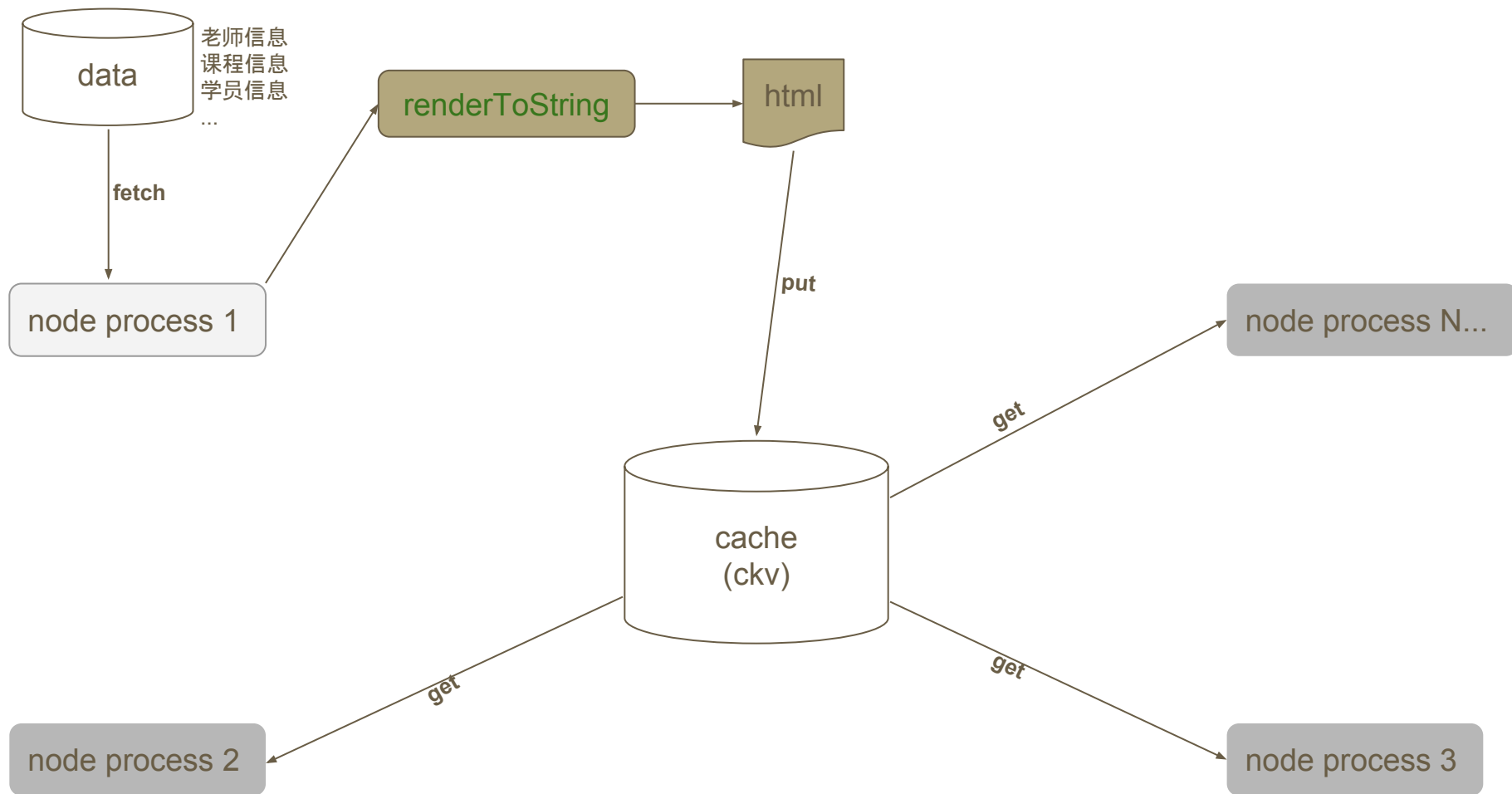
平均耗时≈125ms

虽然比用户端拉取快很多, 但是 还是相当耗时

# Cache



# Cache: 相对固定的内容



# Cache进阶

## 专题课-单词语法 系统课

9月30日，上午11点

距报名截止还剩：73天

### 适用人群

学习阶段: 高中

适合地区: 全国

教材版本: 所有版本教材

### 课程目标

掌握名词性从句分类，掌握部分题目出题套路以及应对策略

### 主要知识点

名词性从句

### 目录

#### 01 单词语法快乐通-名词从句篇

10-28 周五 11:00-14:30 / 于瑶

### 老师介绍



于瑶

自学三个月考出托福118雅思8.5分的高分创立了自己的大学生职业培训公司，3年内分公司开设5家，遍布山东，营业额破3000万短短4年成为青岛最受欢迎的独立教师，从不招生却场...

### 课程描述

还记得上期课在定语从句中提到的野蛮人二爷和圣骑士斌哥

¥99

购买

```
{  
    is_login: false,  
    end: '2016-08-18',  
    has_bought: false,  
    course_detail: {  
        .....  
    }  
}
```

## 专题课-单词语法 系统课

9月30日，上午11点

已购买

### 适用人群

学习阶段: 高中

适合地区: 全国

教材版本: 所有版本教材

### 课程目标

掌握名词性从句分类，掌握部分题目出题套路以及应对策略

### 主要知识点

名词性从句

### 目录

#### 01 单词语法快乐通-名词从句篇

10-28 周五 11:00-14:30 / 于瑶

### 老师介绍



于瑶

自学三个月考出托福118雅思8.5分的高分创立了自己的大学生职业培训公司，3年内分公司开设5家，遍布山东，营业额破3000万短短4年成为青岛最受欢迎的独立教师，从不招生却场...

### 课程描述

还记得上期课在定语从句中提到的野蛮人二爷和圣骑士斌哥吗？这节课不仅帮你复习定语从句，还帮你总结了定语从句的出题套路...

¥99

已购买

```
{  
  is_login: true,  
  end: '2016-08-18',  
  has_bought: true,  
  course_detail: {  
    .....  
  }  
}
```



缓存有难度/风险:

- 变动频繁
- 与登录态关联
- .....

专题课-单词语法 (系统课)

9月30日，上午11点  
距报名截止还剩：73天

适用人群

学习阶段: 高中  
适合地区: 全国  
教材版本: 所有版本教材

课程目标

掌握名词性从句分类，掌握部分题目出题套路以及应对策略

主要知识点

名词性从句

目录

01 单词语法快乐通-名词从句篇  
10-28 周五 11:00-14:30 / 于瑶

老师介绍



于瑶  
自学三个月考出托福118雅思8.5分的高分创立了自己的大学生职业培训公司，3年内分公司开设5家，遍布山东，营业额破3000万短短4年成为青岛最受欢迎的独立教师，从不招生却场...

课程描述

还记得上期课在定语从句中提到的野蛮人二爷和圣骑士斌哥么？这次还是那群人，只不过这次要做名词性从句的题...

¥99

购买

专题课-单词语法 (系统课)

9月30日，上午11点  
已购买

适用人群

学习阶段: 高中  
适合地区: 全国  
教材版本: 所有版本教材

课程目标

掌握名词性从句分类，掌握部分题目出题套路以及应对策略

主要知识点

名词性从句

目录

01 单词语法快乐通-名词从句篇  
10-28 周五 11:00-14:30 / 于瑶

老师介绍



于瑶  
自学三个月考出托福118雅思8.5分的高分创立了自己的大学生职业培训公司，3年内分公司开设5家，遍布山东，营业额破3000万短短4年成为青岛最受欢迎的独立教师，从不招生却场...

课程描述

还记得上期课在定语从句中提到的野蛮人二爷和圣骑士斌哥么？这次还是那群人，只不过这次要做名词性从句的题...

¥99

已购买

# CACHE进阶

- 选择性缓存某部分
- 与登录态相关/变动频繁的信息不作直出
- 二次渲染再拉取数据作 UI 更新

```
{  
  is_login: true,  
  end: '2016-08-18',  
  has_bought: true,  
  course_detail: {  
    .....  
  }  
}
```

## 专题课-单词语法

系统课

9月30日，上午11点

### 适用人群

学习阶段: 高中

适合地区: 全国

教材版本: 所有版本教材

### 课程目标

掌握名词性从句分类，掌握部分题目出题套路以及应对策略

### 主要知识点

名词性从句

## 服务端渲染结果

### 目录

#### 01 单词语法快乐通-名词从句篇

10-28 周五 11:00-14:30 / 于瑶

### 老师介绍



于瑶

自学三个月考出托福118雅思8.5分的高分创立了自己的大学生职业培训公司，3年内分公司开设5家，遍布山东，营业额破3000万短短4年成为青岛最受欢迎的独立教师，从不招生却场...

### 课程描述

还记得上期课在定语从句中提到的野蛮人二爷和圣骑士斌哥么？这次还是那帮人，只不过换了个叫做名词性从句的游戏在玩，短短一小时一半讲魔幻游戏一半衍生依旧无节操脑洞大开的单词速记，小伙伴们做好掌握名词性从句的准备吧

## 专题课-单词语法

系统课

9月30日，上午11点

已购买

### 适用人群

学习阶段: 高中

适合地区: 全国

教材版本: 所有版本教材

### 课程目标

掌握名词性从句分类，掌握部分题目出题套路以及应对策略

### 主要知识点

名词性从句

## 二次服务端渲染结果

### 目录

#### 01 单词语法快乐通-名词从句篇

10-28 周五 11:00-14:30 / 于瑶

### 老师介绍



于瑶

自学三个月考出托福118雅思8.5分的高分创立了自己的大学生职业培训公司，3年内分公司开设5家，遍布山东，营业额破3000万短短4年成为青岛最受欢迎的独立教师，从不招生却场...

### 课程描述

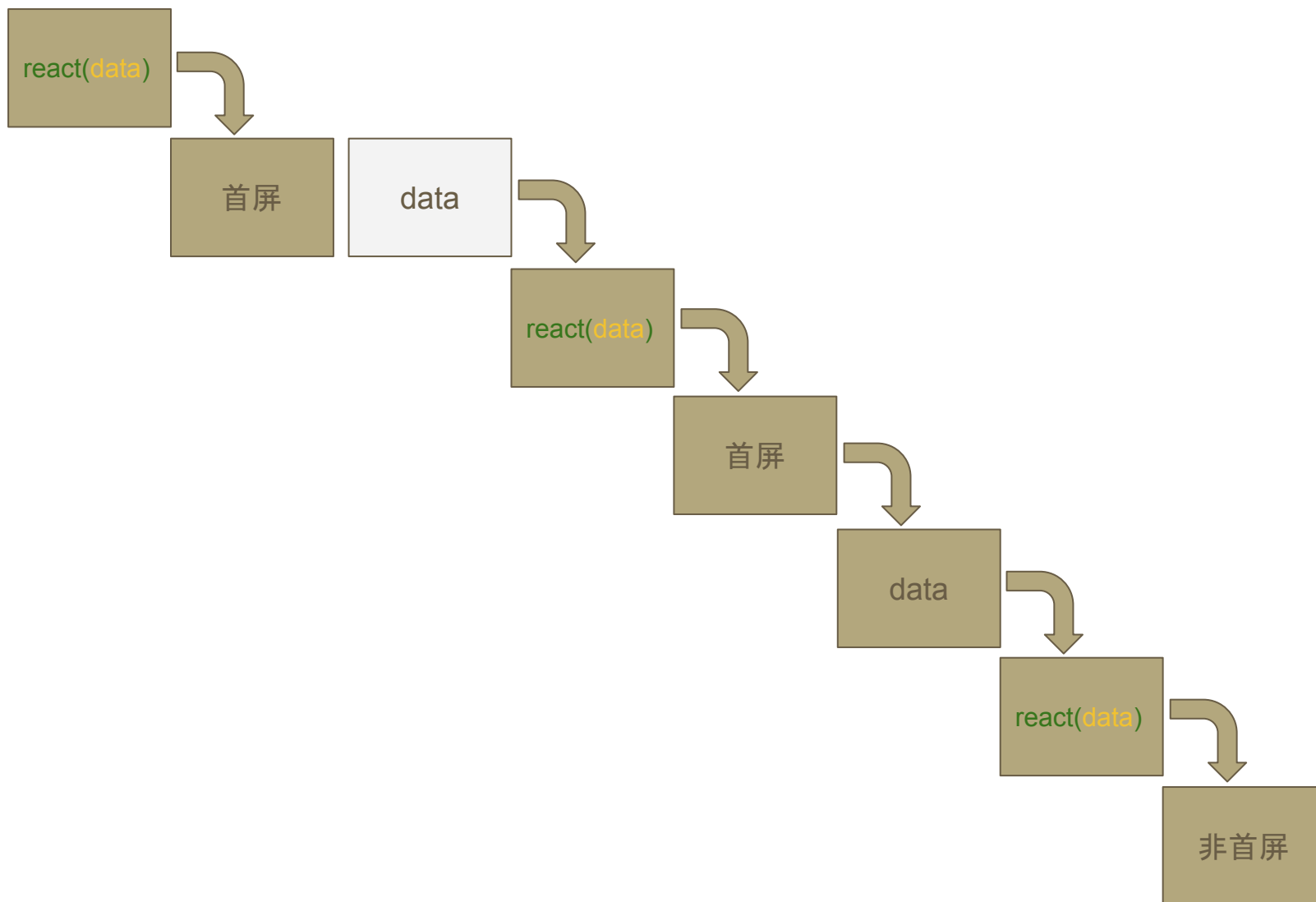
还记得上期课在定语从句中提到的野蛮人二爷和圣骑士斌哥么？这次还是那帮人，只不过换了个叫做名词性从句的游戏在玩，短短一小时一半讲魔幻游戏一半衍生依旧无节操脑洞大开的单词速记，小伙伴们做好掌握名词性从句的准备吧

¥99

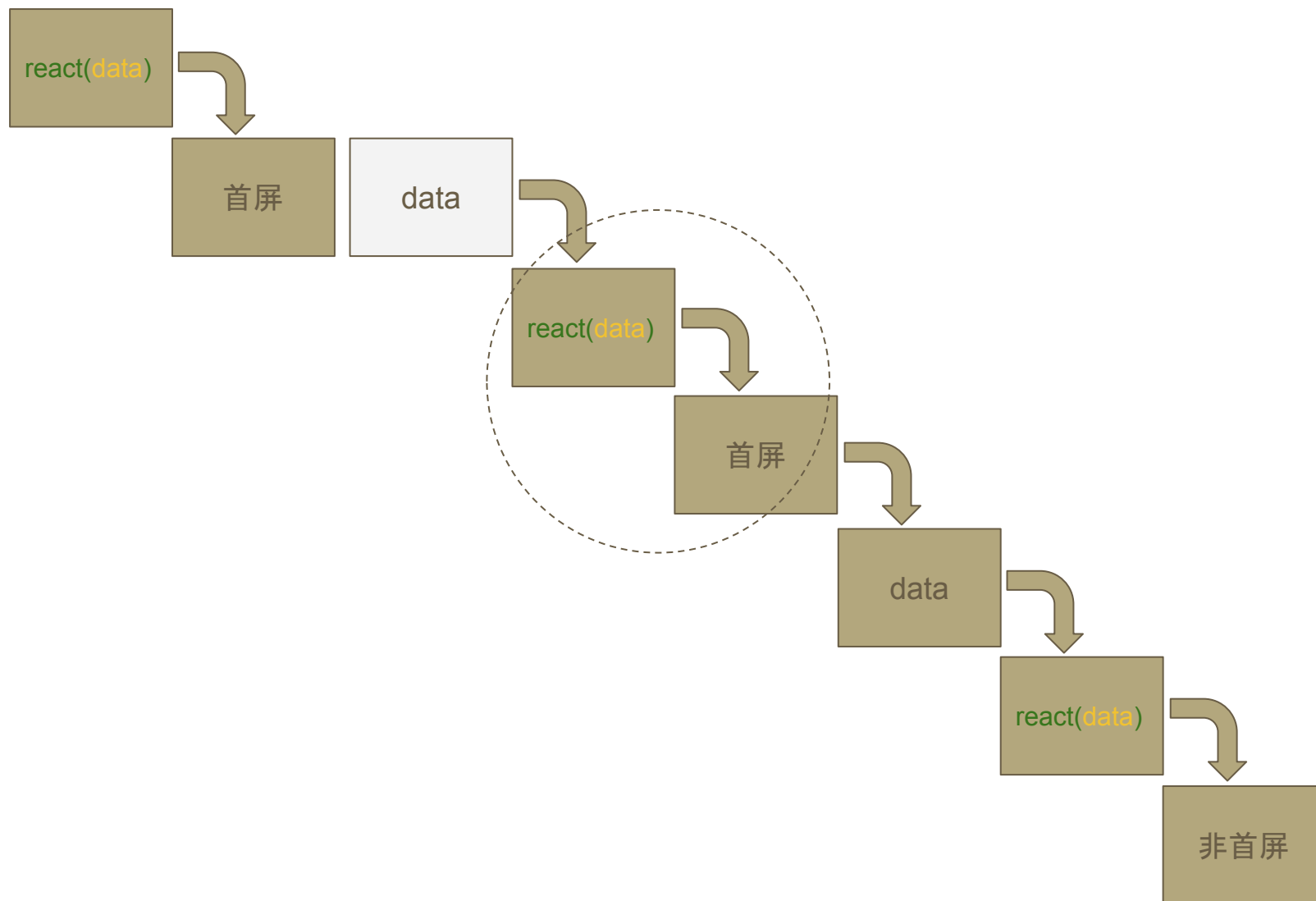
已购买

# 重复渲染？

# 重复渲染？



# 重复渲染？



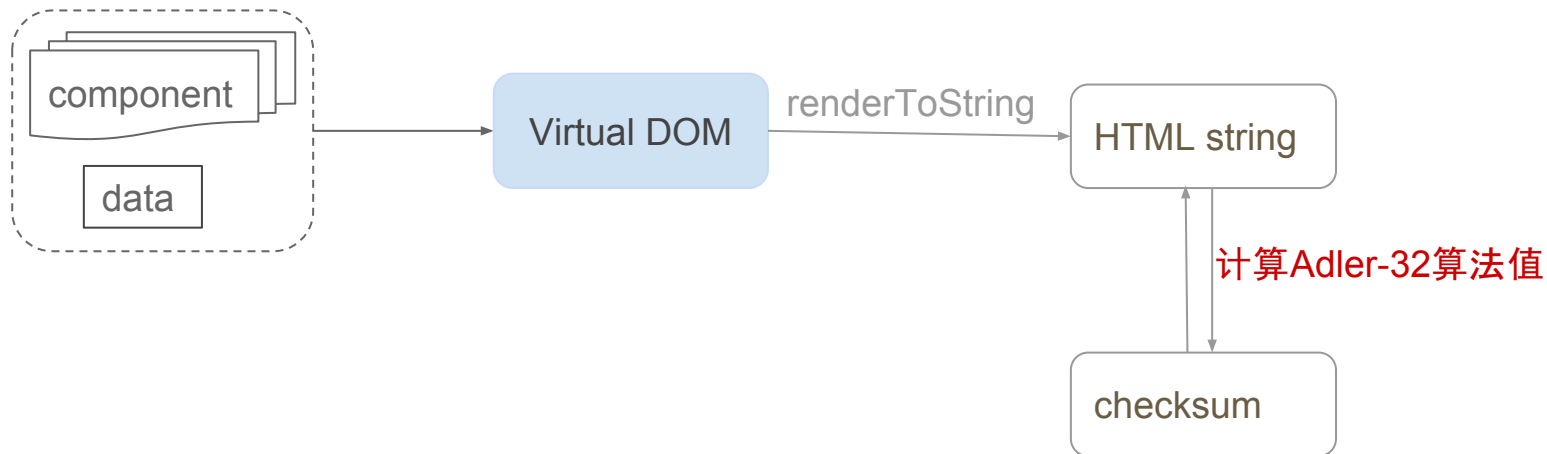
## 服务端首屏文本

```
<div data-reactroot="" data-reactid="1" data-react-checksum="-51068744"><div class="teacher-facade" data-reactid="2"><div class="avatar" data-reactid="3"><
  reactid="5"/></div><span class="teacher-name" data-reactid="6">林传洋</span></div><div class="info" data-reactid="7"><div class="good-rate" data-reactid="8">
  data-reactid="10"><ul data-reactid="11"><li class="" data-modid="introtab" data-reactid="12"><a href="javascript:;" data-reactid="13"><!-- react-text: 14
  <!-- react-text: 17 -->课程<!-- /react-text --><span data-reactid="18"><!-- react-text: 19 --><!-- /react-text --><!-- react-text: 20 -->0<!-- /react-text
  href="javascript:;" data-reactid="23"><!-- react-text: 24 -->评价<!-- /react-text --><span data-reactid="25"><!-- react-text: 26 --><!-- /react-text --><!--
  container" data-reactid="29"><div data-reactid="30"><div class="teacher-intro" data-reactid="31"><section class="teacher-intro-item clearfix" data-reactid="
  class="teacher-label" data-reactid="36">语文due哥</span></li><li data-reactid="37"><span class="teacher-label" data-reactid="38">百场千人讲座</span></li><li
  class="teacher-label" data-reactid="42">高考作文押题王</span></li></ul></section><section class="teacher-intro-item clearfix" data-reactid="43"><h3 data-re
  reactid="47"><span class="dot" data-reactid="48"></span><!-- react-text: 49 -->2015-2016<!-- /react-text --><p><p class="teacher-exp-des" data-reactid="50">
  reactid="53"></span><!-- react-text: 54 -->2009-2015<!-- /react-text --></p><p class="teacher-exp-des" data-reactid="55">快乐学习教育科技集团</p></li><li d
  >2005-2009<!-- /react-text --></p><p class="teacher-exp-des" data-reactid="60">福建师大附中</p></li></ul></section><section class="teacher-intro-item clear
  content-item clearfix" data-reactid="64"><label data-reactid="65">学习阶段:</label><p data-reactid="66">高中</p></div><div class="teacher-content-item clea
  class="teacher-intro-item clearfix" data-reactid="70"><h3 data-reactid="71">个人介绍</h3><p class="teacher-intro-introduction" data-reactid="72">十一年福建
  就是DUE哥<br><br>DUE哥不玩那些高大上的理论概念，DUE哥只教你最“狠辣”的解题技巧<br>把每一篇枯燥的文言文讲成相声，把每一首难懂的诗歌鉴赏演绎成小品<br>集专业
  证！</p></section></div></div><div data-reactid="73"><div class="teacher-courses" data-reactid="74"><div class="course-list-container" data-reactid="75"><ul
  reactid="78"><p class="title" data-reactid="79">暂无课程</p><p class="description" data-reactid="80">这位老师还在备课中，请耐心等待</p></div></li></ul></div>
  data-reactid="84"><li data-modid="allcomment" data-reactid="85"><a href="javascript:;" class="disabled" data-reactid="86"><!-- react-text: 87 -->全部<!--
  react-text: 91 --><!-- /react-text --></span></a></li><li data-modid="positive" data-reactid="92"><a href="javascript:;" class="disabled" data-reactid="93">
```



## checksum: html内容的Adler-32算法值

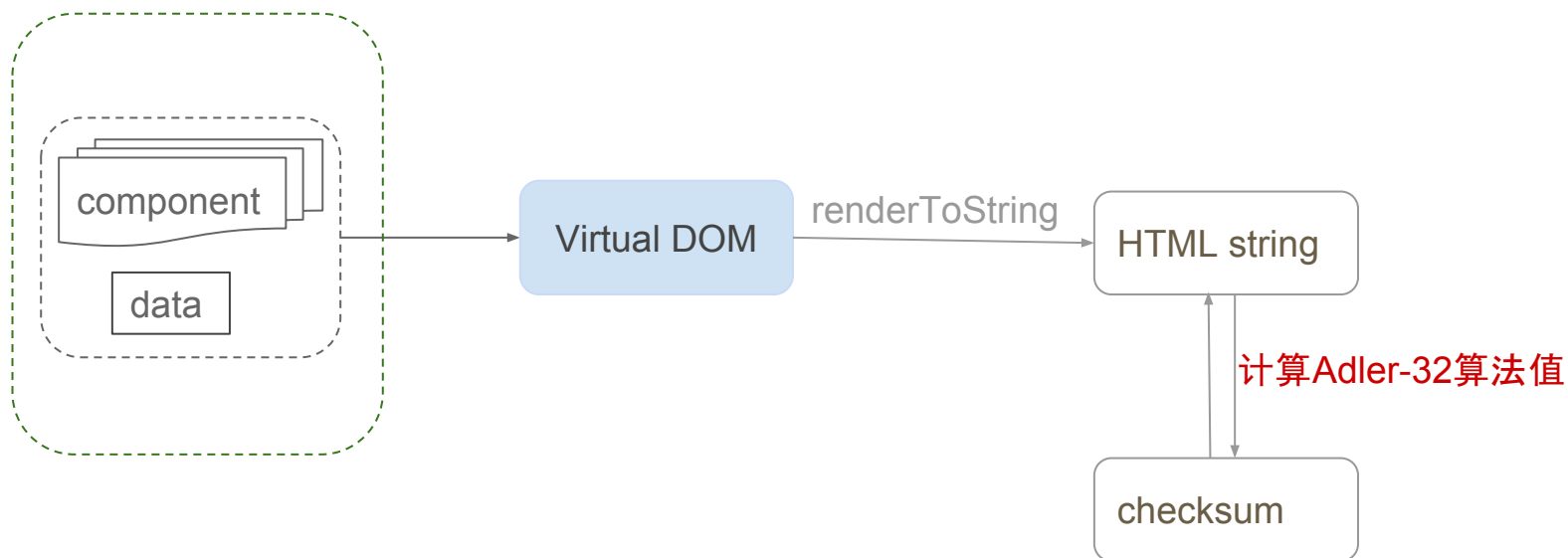
```
<div data-reactroot="" data-reactid="1" data-react-checksum="-51068744"><div class="teacher-facade" data-reactid="2"><div class="avatar" data-reactid="3"><
  reactid="5"/></div><span class="teacher-name" data-reactid="6">林传洋</span></div><div class="info" data-reactid="7"><div class="good-rate" data-reactid="8">
  data-reactid="10"><ul data-reactid="11"><li class="" data-modalid="introtab" data-reactid="12"><a href="javascript:;" data-reactid="13">!-- react-text: 14
  -->课程<!-- /react-text --><span data-reactid="18"><!-- react-text: 19 --><!-- /react-text --><!-- react-text: 20 -->0<!-- /react-text
  href="javascript:;" data-reactid="23"><!-- react-text: 24 -->评价<!-- /react-text --><span data-reactid="25"><!-- react-text: 26 --><!-- /react-text --><!--
  container" data-reactid="29"><div data-reactid="30"><div class="teacher-intro" data-reactid="31"><section class="teacher-intro-item clearfix" data-reactid="
  class="teacher-label" data-reactid="36">语文due哥</span></li><li data-reactid="37"><span class="teacher-label" data-reactid="38">百场千人讲座</span></li><li
  class="teacher-label" data-reactid="42">高考作文押题王</span></li></ul></section><section class="teacher-intro-item clearfix" data-reactid="43"><h3 data-re
  reactid="47"><span class="dot" data-reactid="48"></span><!-- react-text: 49 -->2015-2016<!-- /react-text --><p><p class="teacher-exp-des" data-reactid="50">
  reactid="53"></span><!-- react-text: 54 -->2009-2015<!-- /react-text --></p><p class="teacher-exp-des" data-reactid="55">快乐学习教育科技集团</p></li><li d
  >2005-2009<!-- /react-text --></p><p class="teacher-exp-des" data-reactid="60">福建师大附中</p></li></ul></section><section class="teacher-intro-item clear
  content-item clearfix" data-reactid="64"><label data-reactid="65">学习阶段:</label><p data-reactid="66">高中</p></div><div class="teacher-content-item clea
  class="teacher-intro-item clearfix" data-reactid="70"><h3 data-reactid="71">个人介绍</h3><p class="teacher-intro-introduction" data-reactid="72">十一年福建
  就是DUE哥<br><br>DUE哥不玩那些高大上的理论概念，DUE哥只教你最“狠辣”的解题技巧<br>把每一篇枯燥的文言文讲成相声，把每一首难懂的诗歌鉴赏演绎成小品<br>集专业
  证！</p></section></div></div><div data-reactid="73"><div class="teacher-courses" data-reactid="74"><div class="course-list-container" data-reactid="75"><ul
  reactid="78"><p class="title" data-reactid="79">暂无课程</p><p class="description" data-reactid="80">这位老师还在备课中，请耐心等待</p></div></li></ul></div>
  data-reactid="84"><li data-modalid="allcomment" data-reactid="85"><a href="javascript:;" class="disabled" data-reactid="86"><!-- react-text: 87 -->全部<!--
  react-text: 91 --><!-- /react-text --></span></li></ul><div data-modalid="positive" data-reactid="92"><a href="javascript:;" class="disabled" data-reactid="93">
```





# checksum: html内容的Adler-32算法值

只要组件、数据一致, 前端 **checksum** 也会和后台一致  
直接跳过render(重复渲染)



# 小结

- react + redux: 畅快的开发体验 & 更好的代码解耦
- 性能问题
- 离线包优化但还是不够
- 服务端React提升首屏速度
- 服务端缓存渲染优化
- 防止重复渲染原理

THE END