

对前端『新』技术的实践

郭永峰

2016来了，这是前端技术走向更成熟的一年

技术关键字

ECMAScript 6 Babel Webpack React React-Router Redux Less Material-
ui Mocha

拥抱ES6

不要犹豫了，ES6的规范早已落地，新的规范从语言层面解决了很多由来已久的问题。

现在有原生的 class extends 语法，从语言层面进行统一

```
export default class Point {  
  // constructor方法是类的默认方法  
  constructor(x, y) {  
    this.x = x;  
    this.y = y;  
  }  
  
  // 这里定义的都是类的公共方法  
  // 等同于Point.prototype.toString  
  toString() {  
    return '(' + this.x + ', ' + this.y + ')';  
  }  
}
```

箭头函数避免了 this 上下文的坑

```
class Animal {  
  constructor(name){  
    this.name = name;  
  }  
  drink(type = 'water'){  
    setInterval( () => {  
      console.log(`${this.name} want to drink ${type}`)  
    }, 1000)  
  }  
}
```

块级的 let/const 避免了 var hoisting 的坑

```
console.log(a);  
console.log(b);  
  
let a = 1;  
var b = 2;
```

templateString 避免繁琐的手动字符串拼接

```
let name = 'bull';  
let type = 'water';  
  
console.log(`${name} want to drink ${type}`)
```


ES6的 export 和 import 轻松提供良好的模块机制

```
'use strict';  
import React, { Component } from 'react';  
  
class Comment extends Component {  
  render(){  
    return <h1>hello world</h1>  
  }  
}  
  
export default Comment
```

学习资料推荐

1. ES6入门
2. ES6语法快餐

用Babel来解析你的代码

Babel 是一个 JavaScript 编译器，强大的Babel解析器及其生态，助推ES6和React的推广，让我们可以更更早更舒服的拥抱新技术。

- 模块 : `babel-core` `babel-cli` `babel-polyfill`
- 预设 presets : `babel-preset-es2015` `babel-preset-react` `babel-preset-stage-0`
- 插件 plugins : `babel-plugin-transform-runtime` `babel-plugin-transform-es2015-modules-amd` `babel-plugin-syntax-decorators`

- 你可以这样来用

```
$ cd my-project  
$ touch .babelrc
```

- 配置你的 `.baberrc`

```
{  
  "presets": ["react", "es2015"],  
  "plugins": []  
}
```

- 下载相应的插件

```
$ npm install --save-dev babel-core babel-preset-react babel-preset-2015 babel-cli
```

- 运行解析

```
$ babel src -d lib
```

学习资料推荐

1. Babel中文网
2. Babel小书

Webpack

A bundler for javascript and friends. Packs many modules into a few bundled assets.

Webpack是一款模块加载器兼打包工具，它能把各种资源，例如JS（含JSX）、coffee、样式（含less/sass）、图片等都作为模块来使用和处理。

我们一般需要创建一个 `webpack.config.js` , 比如这样

```
var path = require('path');
var webpack = require('webpack');

module.exports = {
  entry: path.resolve(__dirname, 'app/index.js'),
  output: {
    path: path.resolve(__dirname, 'public'),
    filename: 'bundle.js',
  },
  module: {
    loaders: [{
      test: /\.js$/,
      loaders: ['react-hot', 'babel'],
      exclude: path.resolve(__dirname, 'node_modules')
    }]
  },
  plugins: []
};
```


学习资料推荐

1. [Webpack官网](#)
2. [手把手教你配置Webpack系列](#)

在项目中使用React

无论React火不火，我们真正最应该关心的是它能帮助我们解决哪些业务问题。

我选择的理由

- **组件化、虚拟DOM** 在 **复用** 以及 **性能** 上带来的一般好处
- React思想使得开发者之间更好的 **分工与合作**，在 **配合上非常顺畅**。
- **规范的接口** 以及极强的约束，使得整个 **代码结构清晰**，不同开发者的代码高度一致。
- **技术生态**。越来越多的开发调试工具以及JavaScript框架正在依附着react生长。
- 配合ES6的使用，可以大幅 **提高编程效率**，降低很多之前的代码坑。
- 有很 **优秀的开源组件库** 可以使用

同时也面临一些问题

- 基础组件的建设需要加强
- 服务端渲染
- 在数据层的实践经验不够

```
import React, { Component } from 'react';

class Button extends Component {
  static displayName = 'Button'

  static propTypes = {
    onClick: React.PropTypes.func
  }

  state = {
    disabled: this.props.disabled,
  }

  handleClick = () => {
    if (this.props.onClick) {
      this.props.onClick()
    }
  }

  render() {
    return (
      <button onClick={this.handleClick}
        disabled={this.state.disabled}
      </button>
    )
  }
}

export default Button
```

学习资料推荐

1. [React官网](#)
2. [React基础入门](#)

结合React-router

react-router is a complete routing library for React.

app/index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import { Router, useRouterHistory } from 'react-router';
import routes from './routes';
// Manage browser history with JavaScript
import { createHashHistory } from 'history';

const appHistory = useRouterHistory( createHashHistory )({
  queryKey: false
});

let root = document.getElementById('app');

ReactDOM.render(
  <Router
    history={appHistory}
    routes={routes}
  />, root
);
```


app/routes/index.js

```
import React from 'react';
import { Route, IndexRoute } from 'react-router';
import { App, Home, Shift, AttendancePlace } from '../containers';

export default (
  <Route path="/" component={App}>
    <Route path="home" component={Home}>
      <IndexRoute component={Shift} />
    </Route>
    <Route path="shift" component={Shift} />
    <Route path="attendancePlace" component={AttendancePlace} />
    <IndexRoute component={Home} />
    <NotFoundRoute component={NotFound} />
  </Route>
);
```

学习资料推荐

1. [React-router on github](#)
2. [react-router-tutorial](#)
3. [docs](#)
4. [history](#)

QAQ

THANKS