



介绍 Meteor

Meteor 项目中安装 React

react-router 路由控制

当输入框“受控”

积极 UI

全栈随动

## 大揭底：到底差别在什么地方

到了本课程最后一集了，来总结梳理一下。下面我一共总结了三点 Meteor 和 Express 实现 Ajax 效果的区别。

**Meteor 是以实时为默认**

由于有了实时数据的订阅，一旦服务器上的 `comments` 列表有了变化。那么客户端的 `comments` 数据就会自动变化，而 `React` 组件的特性是，一旦 `state` 值（对应当前情况，就是 `comments`）有了变化，界面就能自动刷新（内置的 `socket.io`）。

所以最终的结果是：一旦有其他人发布了评论，我自己的浏览器中是会自动显示出来的，也就是 `Meteor-React` 实现的这个评论框，就是一个聊天室。

## React 和 jQuery/Zepto 做事方式完全不同

我们案例启示对 `Express` 很不公平，因为 `Express` 也一样可以通过前后端分离的思路，配合 `React` 来使用，那样也是现代感爆棚的。但是我们这里却是把 `jQuery/Zepto+Express` 跟 `React+Meteor` 做了对比。所以 `Meteor` 自然就显得更加逼格出众。

`jQuery/Zepto` 实现 `Ajax` 的方式是“命令式”的，也就是我们会用 `js` 代码去直接操作修改 `DOM` 节点，功能复杂了，这样写起来代码就会很乱，因为人的脑子对随时变化的一个过程很容易产生混淆和印象模糊。而 `React` 这里，`UI` 会自动随着数据的变化而“刷新”，也就是“声明式”的，通过 `state` 概念的引入，组件在我们操作它的每一个点上都是一个静止不变的 `UI`，而人脑对于静止的东西是很容易把控的，所以用 `React` 开发很复杂的组件也会觉得很简单清楚。

## Express 可以做 API 服务器

`Express` 所代表的“小框架”，相对于 `Meteor` 这种大而全的框架是有着高度的自定义的灵活性的。比如用 `React` 做前端（使用 `Webpack` 构建）加上 `Express` 做后台 `API` 服务器，这种架构也是非常流行的。

## 最终课程代码

- 。课程所有的最终代码在 [coding.net](https://coding.net) 的仓库中

欢迎添加 Peter 的微信：happypeter1983