

React.js 小书

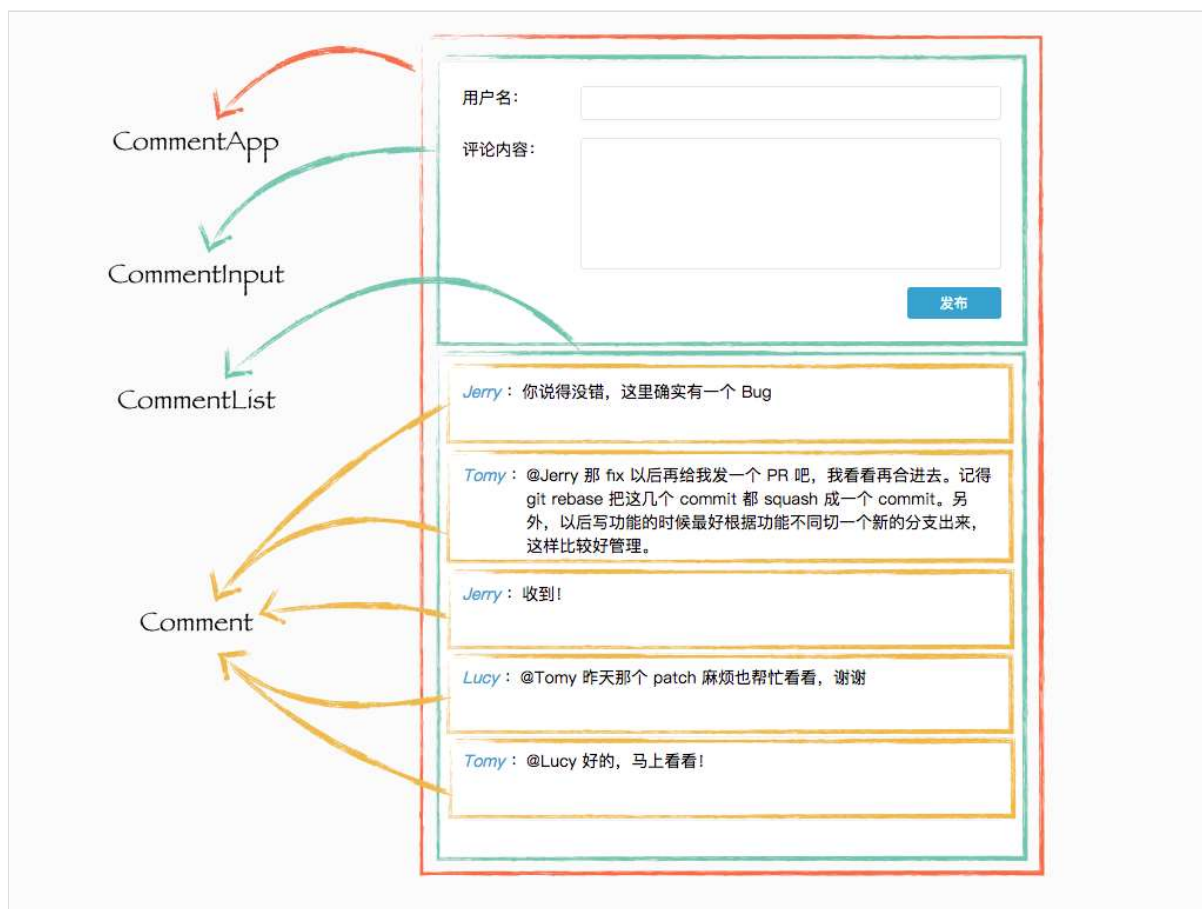
[<-- 返回首页](#)

实战分析：评论功能（八）

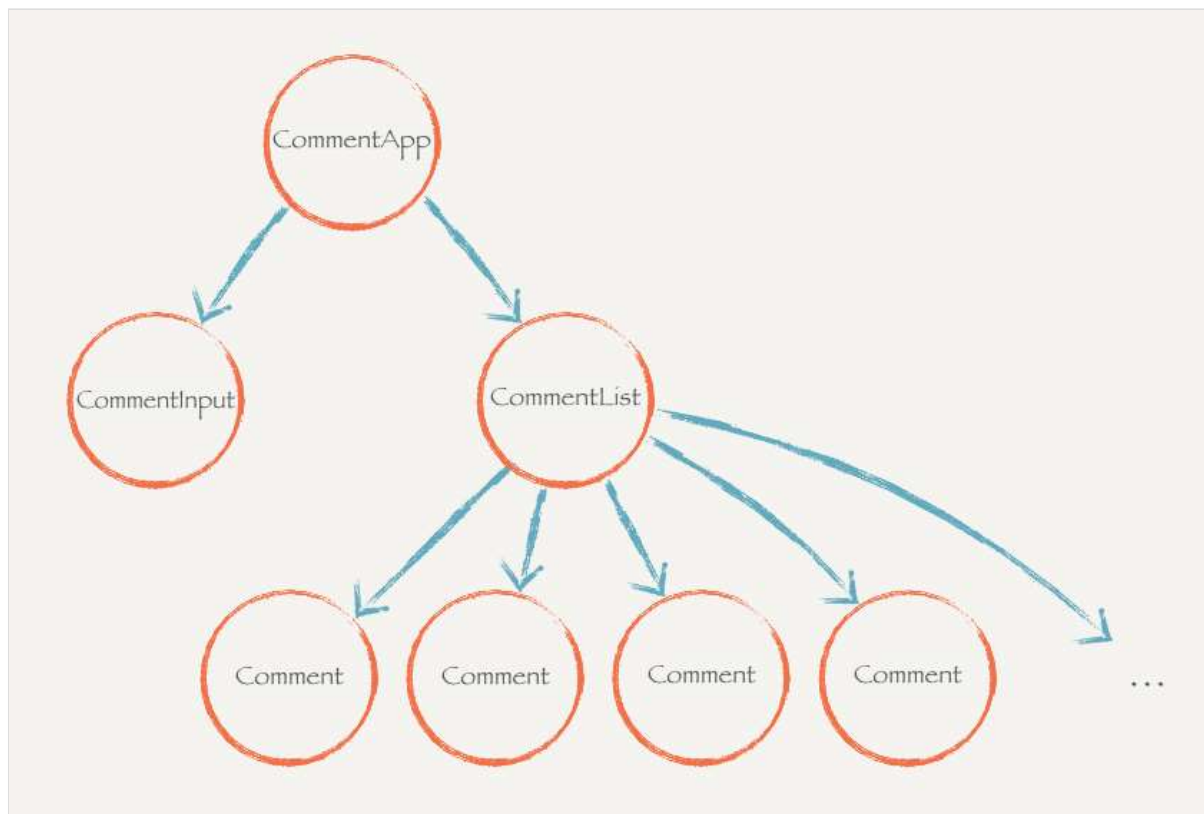
- 作者：[胡子大哈](#)
- 原文链接：<http://huziketang.com/books/react/lesson45>
- 转载请注明出处，保留原文链接和作者信息。

（本文未审核）

接下可以重构组件部分的内容了。先回顾一下之前我们是怎么划分组件的：



组件树：



这样划分方式当然是没错的。但是在组件的实现上有些问题，我们之前并没有太多地考虑复用性问题。所以现在可以看看 `comment-app2` 的 `CommentInput` 组件，你会发现它里面有一些 `LocalStorage` 操作：

```
...
_loadUsername () {
  const username = localStorage.getItem('username')
  if (username) {
    this.setState({ username })
  }
}

_saveUsername (username) {
  localStorage.setItem('username', username)
}

handleUsernameBlur (event) {
  this._saveUsername(event.target.value)
}

handleUsernameChange (event) {
  this.setState({
    username: event.target.value
  })
}
...
```

它是一个依赖 `LocalStorage` 数据的 `Smart` 组件。如果别的地方想使用这个组件，但是数据却不是从 `LocalStorage` 里面取的，而是从服务器取的，那么这个组件就没法复用了。

所以现在需要从复用性角度重新思考如何实现和组织这些组件。假定在目前的场景下，`CommentInput`、`CommentList`、`Comment` 组件都是需要复用的，我们就要把它们做成 Dumb 组件。

幸运的是，我们发现其实 `CommentList` 和 `Comment` 本来就是 Dumb 组件，直接把它们俩移动到 `components` 目录下即可。而 `CommentInput` 就需要好好重构一下了。我们把它里面和 `LocalStorage` 操作相关的代码全部删除，让它从 `props` 获取数据，变成一个 Dumb 组件，然后移动到 `src/components/CommentInput.js` 文件内：

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

export default class CommentInput extends Component {
  static propTypes = {
    username: PropTypes.any,
    onSubmit: PropTypes.func,
    onUserNameInputBlur: PropTypes.func
  }

  static defaultProps = {
    username: ''
  }

  constructor (props) {
    super(props)
    this.state = {
      username: props.username, // 从 props 上取 username 字段
      content: ''
    }
  }

  componentDidMount () {
    this.textarea.focus()
  }

  handleUsernameBlur (event) {
    if (this.props.onUserNameInputBlur) {
      this.props.onUserNameInputBlur(event.target.value)
    }
  }

  handleUsernameChange (event) {
    this.setState({
      username: event.target.value
    })
  }

  handleContentChange (event) {
    this.setState({
      content: event.target.value
    })
  }
}
```

```
handleSubmit () {  
  if (this.props.onSubmit) {  
    this.props.onSubmit({  
      username: this.state.username,  
      content: this.state.content,  
      createTime: +new Date()  
    })  
  }  
  this.setState({ content: '' })  
}  
  
render () {  
  // render 方法保持不变  
  // ...  
}
```

其实改动不多。原来 `CommentInput` 需要从 `LocalStorage` 中获取 `username` 字段，现在让它从 `props` 里面去取；而原来用户名的输入框 `blur` 的时候需要保存 `username` 到 `LocalStorage` 的行为也通过 `props.onUserNameInputBlur` 传递到上层去做。现在 `CommentInput` 是一个 Dumb 组件了，它的所有渲染操作都只依赖于 `props` 来完成。

现在三个 Dumb 组件 `CommentInput`、`CommentList`、`Comment` 都已经就位了。但是单靠 Dumb 组件是没办法完成应用逻辑的，所以接下来我们要构建 Smart 组件来带领它们完成任务。

因为第三方评论工具有问题，对本章节有任何疑问的朋友可以移步到 [React.js 小书的论坛](#) 发帖，我会回答大家的疑问。

[下一节：实战分析：评论功能（九）](#)

[上一节：实战分析：评论功能（七）](#)

如果你觉得小书写得还不错，可以请胡子大哈喝杯茶 :)

赞赏

或者传播一下知识也是一个很好的选择