

React.js 小书

[<-- 返回首页](#)

实战分析：评论功能（一）

- 作者：[胡子大哈](#)
- 原文链接：<http://huziketang.com/books/react/lesson14>
- 转载请注明出处，保留原文链接和作者信息。

课程到这里大家已经掌握了 `React.js` 的基础知识和组件的基本写法了。现在可以把我们所学到的内容应用于实战当中。这里给大家提供一个实战的案例：一个评论功能。效果如下：

用户名:

评论内容:

发布

Jerry: 你说得没错，这里确实有一个 Bug

Tomy: @Jerry 那 fix 以后再给我发一个 PR 吧，我看看再合进去。记得 git rebase 把这几个 commit 都 squash 成一个 commit。另外，以后写功能的时候最好根据功能不同切一个新的分支出来，这样比较好管理。

Jerry: 收到!

Lucy: @Tomy 昨天那个 patch 麻烦也帮忙看看，谢谢

Tomy: @Lucy 好的，马上看看!

[在线演示地址](#)

接下来会带大家一起来学习如何分析、编写这个功能。在这个过程中会补充一些之前没有提及的知识点，虽然这些知识点之前没有单独拿出来讲解，但是这些知识点也很关键。

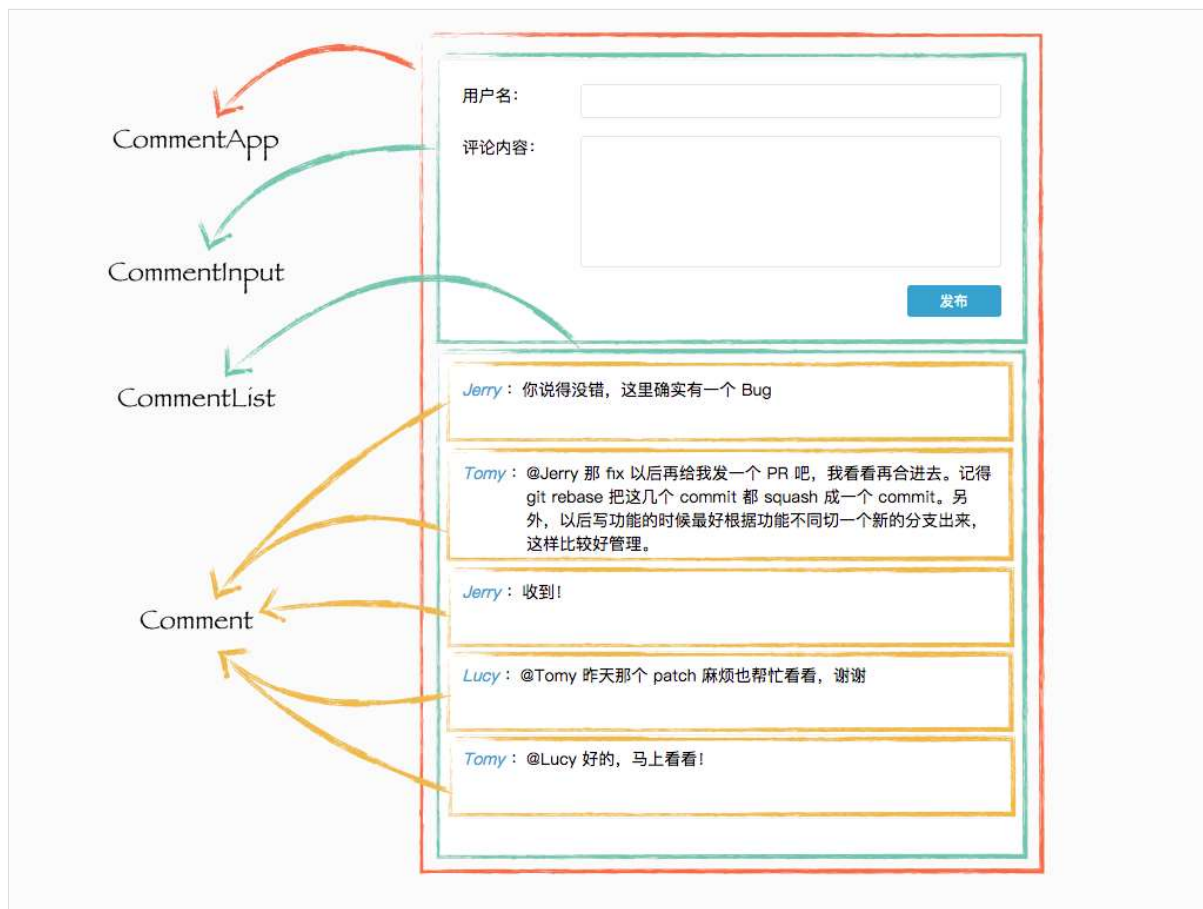
组件划分

React.js 中一切都是组件，用 React.js 构建的功能其实也就是由各种组件组合而成。所以拿到一个需求以后，我们要做的第一件事情就是理解需求、分析需求、划分这个需求由哪些组件构成。

组件的划分没有特别明确的标准。划分组件的目的性是为了代码可复用性、可维护性。只要某个部分有可能复用到别的地方，你都可以把它抽离出来当成一个组件；或

者把某一部分抽离出来对代码的组织和管理会带来帮助，你也可以毫不犹豫地把它抽离出来。

对于上面这个评论功能，可以粗略地划分成以下几部分：



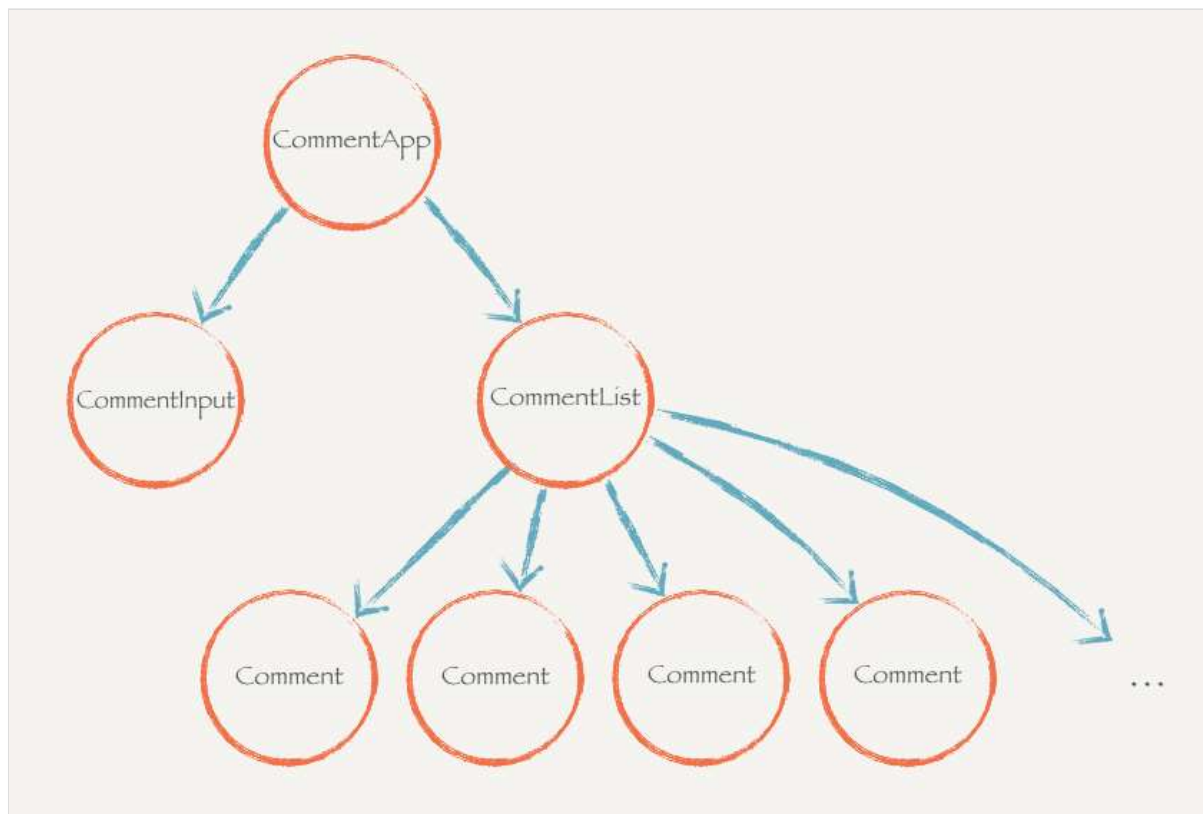
CommentApp：评论功能的整体用一个叫 **CommentApp** 的组件包含起来。**CommentApp** 包含上部和下部两部分。

CommentInput：上面部分是负责用户输入可操作的输入区域，包括输入评论的用户名、评论内容和发布按钮，这一部分功能划分到一个单独的组件 **CommentInput** 中。

CommentList：下面部分是评论列表，用一个叫 **CommentList** 的组件负责列表的展示。

Comment：每个评论列表项由独立的组件 **Comment** 负责显示，这个组件被 **CommentList** 所使用。

所以这个评论功能划分成四种组件，**CommentApp**、**CommentInput**、**CommentList**、**Comment**。用组件树表示：



现在就可以尝试编写代码了。

组件实现

在写代码之前，我们先用 `create-react-app` 构建一个新的工程目录。所有的评论功能在这个工程内完成：

```
create-react-app comment-app
```

然后在工程目录下的 `src/` 目录下新建四个文件，每个文件对应的是上述的四个组件。

```
src/  
  CommentApp.js  
  CommentInput.js  
  CommentList.js  
  Comment.js  
  ...
```

你可以注意到，这里的文件名的开头是大写字母。我们遵循一个原则：如果一个文件导出的是一个类，那么这个文件名就用大写开头。四个组件类文件导出都是类，所以都是大写字母开头。

我们先铺垫一些基础代码，让组件之间的关系清晰起来。遵循“自顶而下，逐步求精”的原则，我们从组件的顶层开始，再一步步往下构建组件树。先修改

`CommentApp.js` 如下：

```
import React, { Component } from 'react'
import CommentInput from './CommentInput'
import CommentList from './CommentList'

class CommentApp extends Component {
  render() {
    return (
      <div>
        <CommentInput />
        <CommentList />
      </div>
    )
  }
}

export default CommentApp
```

`CommentApp` 现在暂时还很简单，文件顶部引入了 `CommentInput` 和 `CommentList` 。然后按照上面的需求，应用在了 `CommentApp` 返回的 JSX 结构中，上面是用户输入区域，下面是评论列表。

现在来修改 `CommentInput.js` 中的内容：

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

class CommentInput extends Component {
  render() {
    return (
      <div>CommentInput</div>
    )
  }
}

export default CommentInput
```

这里暂时让它只简单返回 `<div>` 结构，同样地修改 `CommentList.js` :

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

class CommentList extends Component {
  render() {
    return (
      <div>CommentList</div>
    )
  }
}

export default CommentList
```

现在可以把这个简单的结构渲染到页面上看看什么效果，修改 `src/index.js` :

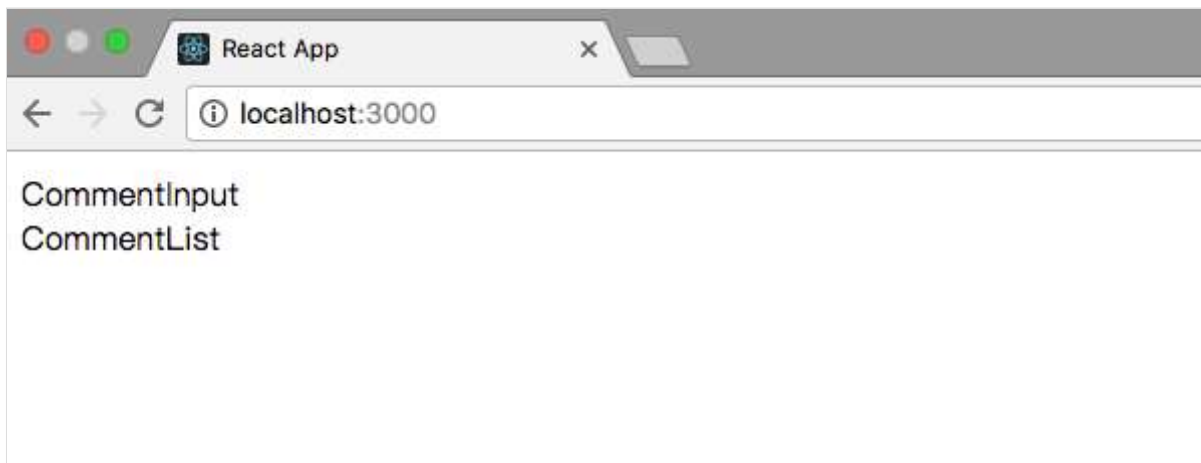
```
import React from 'react'
import ReactDOM from 'react-dom'
import CommentApp from './CommentApp'
import './index.css'

ReactDOM.render(
  <CommentApp />,
  document.getElementById('root')
)
```

然后进入工程目录启动工程：

```
npm run start
```

在浏览器中可以看到，基本的结构已经渲染到了页面上了：



添加样式

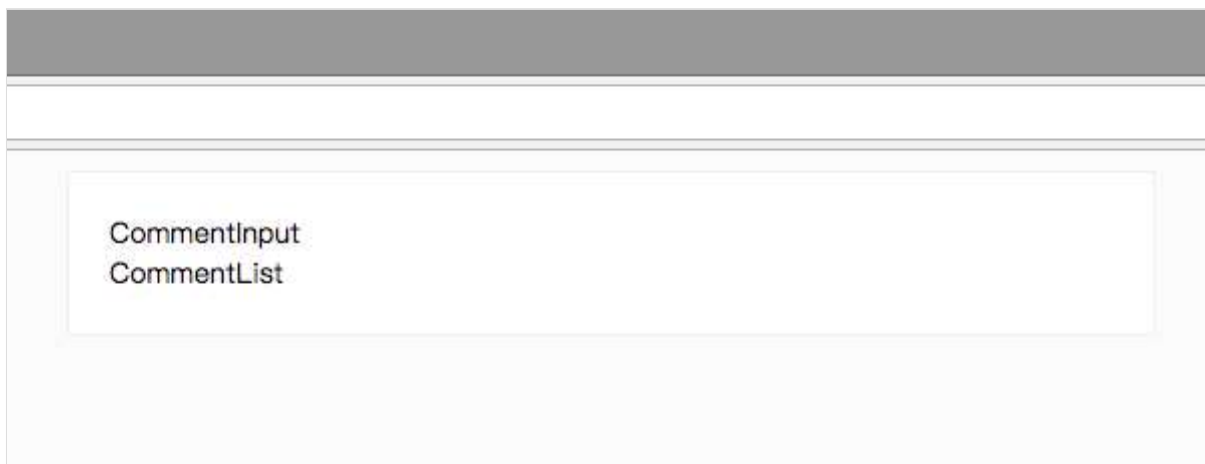
现在想让这个结构在浏览器中居中显示，我们就要给 `CommentApp` 里面的 `<div>` 添加样式。修改 `CommentApp` 中的 `render` 方法，给它添加一个 `wrapper` 类名：

```
...
class CommentApp extends Component {
  render() {
    return (
      <div className='wrapper'>
        <CommentInput />
        <CommentList />
      </div>
    )
  }
}
...
```

然后在 `index.css` 文件中添加样式：

```
.wrapper {  
  width: 500px;  
  margin: 10px auto;  
  font-size: 14px;  
  background-color: #fff;  
  border: 1px solid #f1f1f1;  
  padding: 20px;  
}
```

在浏览器中可以看到样式生效了：



评论功能案例的所有样式都是通过这种方式进行添加。由于我们专注点在于 React.js，本案例后续不会在样式上过于纠缠。这里写好了一个样式文件（[index.css](#)）提供给大家，可以复制到 `index.css` 当中。后续只需要在元素上加上类名就可以了。

如何在 React.js 中使用样式有很多种方式，也是一个比较大的话题，有很多种不同的方式也有很多不同的争论，这个话题后续有机会会重点讲解。

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（二）](#)

[上一节：渲染列表数据](#)

如果你觉得小书写得还不错，可以请胡子大哈喝杯茶 :)

赞赏

3 条评论，3 人参与。

★ 4



我有话说...

使用社交帐号登录

发布前先点击左边的按钮登录

最新评论



时间被海绵吃了 • 4月17日 14:16
想问下大神图是用什么软件画的~
顶 • 回复 • 分享»



pang • 4月8日 16:02
虽然我用Vue的，但看了前面十几章，感觉写的非常好，很适合入门。
顶 • 回复 • 分享»



iq72 • 3月31日 09:47
很想知道那个彩色的树图是用什么话的🤔
顶 • 回复 • 分享»

友言?