

React.js 小书

[<-- 返回首页](#)

实战分析：评论功能（五）

- 作者：[胡子大哈](#)
- 原文链接：<http://huziketang.com/books/react/lesson26>
- 转载请注明出处，保留原文链接和作者信息。

（本文未审核）

持久化评论

同样地，可以通过类似于用户名持久化的方式对评论列表内容进行持久化，让用户发布的评论在刷新页面以后依然可以存在。修改 `src/CommentApp.js`：

```
class CommentApp extends Component {
  constructor () {
    super()
    this.state = {
      comments: []
    }
  }

  componentWillMount () {
    this._loadComments()
  }

  _loadComments () {
    let comments = localStorage.getItem('comments')
    if (comments) {
      comments = JSON.parse(comments)
      this.setState({ comments })
    }
  }

  _saveComments (comments) {
    localStorage.setItem('comments', JSON.stringify(comments))
  }

  handleSubmitComment (comment) {
    if (!comment) return
    if (!comment.username) return alert('请输入用户名')
    if (!comment.content) return alert('请输入评论内容')
    const comments = this.state.comments
    comments.push(comment)
    this.setState({ comments })
    this._saveComments(comments)
  }
}
```

```
}  
...
```

我们增加了 `_loadComments` 和 `_saveComments` 分别用于加载和保存评论列表数据。用户每次提交评论都会把评论列表数据保存一次，所以我们在 `handleSubmitComment` 调用 `_saveComments` 方法；而在 `componentWillMount` 中调用 `_loadComments` 方法，在组件开始挂载的时候把评论列表数据加载出来 `setState` 到 `this.state` 当中，组件就可以渲染从 `LocalStorage` 从加载出来的评论列表数据了。

现在发布评论，然后刷新可以看到我们的评论并不会像以前一样消失。非常的不错，持久化评论的功能也完成了。

显示评论发布时间

现在我们给每条评论都加上发布的日期，并且在评论列表项上显示已经发表了多久，例如“1 秒前”、“30分钟前”，并且会每隔 5 秒进行更新。修改 `src/CommentInput.js` 当用户点击发布按钮的时候，传出去的评论数据带上评论发布的时间戳：

```
...  
handleSubmit () {  
  if (this.props.onSubmit) {  
    this.props.onSubmit({  
      username: this.state.username,  
      content: this.state.content,  
      createTime: +new Date()  
    })  
  }  
  this.setState({ content: '' })  
}  
...
```

在评论列表项上显示评论，修改 `src/comment.js`：

```
class Comment extends Component {  
  static propTypes = {  
    comment: PropTypes.object.isRequired  
  }  
  
  constructor () {  
    super()  
    this.state = { timeString: '' }  
  }  
  
  componentWillMount () {  
    this._updateTimeString()  
  }  
  
  _updateTimeString () {  
    const comment = this.props.comment
```

```
const duration = (+Date.now() - comment.createdAt) / 1000
this.setState({
  timeString: duration > 60
    ? `${Math.round(duration / 60)} 分钟前`
    : `${Math.round(Math.max(duration, 1))} 秒前`
})
}

render () {
  return (
    <div className='comment'>
      <div className='comment-user'>
        <span>{this.props.comment.username} </span>:
      </div>
      <p>{this.props.comment.content}</p>
      <span className='comment-createdtime'>
        {this.state.timeString}
      </span>
    </div>
  )
}
```

每个 `Comment` 组件实例会保存一个 `timeString` 状态，用于该评论显示发布了多久。`_updateTimeString` 这个私有方法会根据 `props.comment` 里面的 `createdAt` 来更新这个 `timeString`：计算当前时间和评论发布的时间差，如果已经发布 60 秒以上就显示分钟，否则就显示秒。然后 `componentWillMount` 会在组件挂载阶段调用 `_updateTimeString` 更新一下这个字符串，`render()` 方法就把这个显示时间差的字符串渲染到一个 `` 上。

再看看页面显示：

这时候的时间是不会自动更新的。除非你手动刷新页面，否则永远显示“1 秒前”。我们可以在 `componentWillMount` 中启动一个定时器，每隔 5 秒调用一下

`_updateTimeString`，让它去通过 `setState` 更新 `timeString`：

```
...
componentWillMount () {
  this._updateTimeString()
  this._timer = setInterval(
    this._updateTimeString.bind(this),
    5000
  )
}
...
```

这样就可以做到评论的发布时间自动刷新了，到这里前 4 个需求都已经完成了。

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（六）](#)

[上一节：实战分析：评论功能（四）](#)

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