

React AJAX

React 组件的数据可以通过 `componentDidMount` 方法中的 Ajax 来获取，当从服务端获取数据时可以将数据存储在 `state` 中，再用 `this.setState` 方法重新渲染 UI。

当使用异步加载数据时，在组件卸载前使用 `componentWillUnmount` 来取消未完成的请求。

以下实例演示了获取 Github 用户最新 gist 共享描述：

React 实例

```
class UserGist extends React.Component {
  constructor(props) {
    super(props);
    this.state = {username: '', lastGistUrl: ''};
  }
  componentDidMount() {
    this.serverRequest = $.get(this.props.source, function (result) {
      var lastGist = result[0];
      this.setState({
        username: lastGist.owner.login,
        lastGistUrl: lastGist.html_url
      });
    }).bind(this);
  }
  componentWillUnmount() {
    this.serverRequest.abort();
  }
  render() {
    return (
      <div>
        {this.state.username} 用户最新的 Gist 共享地址：
        <a href={this.state.lastGistUrl}>{this.state.lastGistUrl}</a>
      </div>
    );
  }
}

ReactDOM.render(
  <UserGist source="https://api.github.com/users/octocat/gists" />,
  document.getElementById('example')
);
```

[尝试一下 »](#)

以上代码使用 jQuery 完成 Ajax 请求。



2 篇笔记

写笔记



上述事件的另外一种写法：

```
class UserGist extends React.Component {
  constructor(props) {
    super(props);
    //在state设置两个属性，以便后续通过state对象来对其进行修改
    this.state = {username: '', lastGistUrl: ''};
    //绑定挂载事件
    this.componentDidMount = this.componentDidMount.bind(this);
  }

  componentDidMount() {
    //接下来操作state时上下文对象发生改变，此处拿到操作句柄
    var that = this;
    //ajax请求
    this.serverRequest = $.ajax({
      url: this.props.source,
      type: "GET",
      dataType: 'jsonp',
      success: function (result) {
        console.log(result.data);
        var lastGist = result.data[0];
        //此处的上下文发生改变，this的指针指向发生了变化，通过上述定义的that来操作
        that.setState({
          username: lastGist.owner.login,
          lastGistUrl: lastGist.html_url
        })
      }
    })
  }

  //卸载React组件时，把多余请求关闭，以免影响其他框架或组件的操作
  componentWillUnmount() {
    this.serverRequest.abort();
  }

  render() {
    return (
      <div>
        {this.state.username} 用户最新的 Gist 共享地址：
        <a href={this.state.lastGistUrl} rel="nofollow">{this.state.lastGistUrl}</a>
      </div>
    );
  }
}
```

```
ReactDOM.render(  
  <UserGist source="https://api.github.com/users/octocat/gists" />,  
  document.getElementById('example')  
);
```

[尝试一下 »](#)

白日放歌 3个月前 [12-20]



可以通过箭头函数绑定 this:

```
class UserGist extends React.Component {  
  constructor(props) {  
    super(props);  
    this.state = {username: '', lastGistUrl: ''};  
  }  
  
  componentDidMount() {  
    //可以通过箭头函数绑定this  
    this.serverRequest = $.get(this.props.source, (result)=> {  
      var lastGist = result[0];  
      this.setState({  
        username: lastGist.owner.login,  
        lastGistUrl: lastGist.html_url  
      });  
    });  
  }  
  
  componentWillUnmount() {  
    this.serverRequest.abort();  
  }  
  
  render() {  
    return (  
      <div>  
        {this.state.username} 用户最新的 Gist 共享地址:  
        <a href={this.state.lastGistUrl} rel="nofollow">{this.state.lastGistUrl}</a>  
      </div>  
    );  
  }  
}  
  
ReactDOM.render(  
  <UserGist source="https://api.github.com/users/octocat/gists" />,  
  document.getElementById('example')  
);
```

光头加暴击 1个月前 **[02-06]**