

# PureComponent拓展

## 资源

### React.PureComponent

`React.PureComponent` 与 `React.Component` 很相似。两者的区别在于 `React.Component` 并未实现 `shouldComponentUpdate()`，而 `React.PureComponent` 中以浅层对比 `prop` 和 `state` 的方式来实现了该函数。

如果赋予 `React` 组件相同的 `props` 和 `state`，`render()` 函数会渲染相同的内容，那么在某些情况下使用 `React.PureComponent` 可提高性能。

#### 注意

`React.PureComponent` 中的 `shouldComponentUpdate()` 仅作对象的浅层比较。如果对象中包含复杂的数据结构，则有可能因为无法检查深层的差别，产生错误的比对结果。仅在你的 `props` 和 `state` 较为简单时，才使用 `React.PureComponent`，或者在深层数据结构发生变化时调用 `forceUpdate()` 来确保组件被正确地更新。你也可以考虑使用 [immutable 对象](#) 加速嵌套数据的比较。

此外，`React.PureComponent` 中的 `shouldComponentUpdate()` 将跳过所有子组件树的 `prop` 更新。因此，请确保所有子组件也都是“纯”的组件。

```
export default class PureComponentPage extends PureComponent {
  constructor(props) { ...
  }
  shouldComponentUpdate() {
    console.log("xixi", arguments);
    return true;
  }
}
```

✖ ▶ Warning: PureComponentPage has a `index.js:1375` method called `shouldComponentUpdate()`. `shouldComponentUpdate` should not be used when extending `React.PureComponent`. Please extend `React.Component` if `shouldComponentUpdate` is used.

例子：

```
import React, { Component, PureComponent } from "react";

export default class PureComponentPage extends Component {
  // export default class PureComponentPage extends PureComponent {
  constructor(props) {
    super(props);
    this.state = {
      counter: 0,
    };
  }
  setCounter = () => {
```

```

    this.setState({
      counter: 100,
    });
  };
  render() {
    const { counter } = this.state;
    console.log("render");
    return (
      <div>
        <h1>PureComponentPage</h1>
        <div onClick={this.setCounter}>counter: {counter}</div>
        <PureChild counter={counter} />
      </div>
    );
  }
}

// class PureChild extends Component {
class PureChild extends PureComponent {
  render() {
    console.log(
      "PureChild render",
      this.isPureReactComponent,
      this.isReactComponent,
    );
    return (
      <div>
        <h1>PureChild</h1>
        <p>counter: {this.props.counter}</p>
        { /* 当是纯组件的时候, 当前时间不再更新 */ }
        <p>date: {new Date().toLocaleTimeString()}</p>
      </div>
    );
  }
}

```