## 打包(Bundling)

## 大多数的 React 应用程序使用[Webpack](https://webpack.js.org/)或 [Browserify](http://browserify.org/) 等工具来 “打包” 文件。打包(Bundling) 是一个处理过程，跟踪导入的文件并将其合并到单个文件：“包” 。然后，这个包文件可以包含到网页上，这样可以一次性加载整个应用程序。

示例

App:

// app.js

import { add } from './math.js';

console.log(add(16, 26)); // 42

// math.js

export function add(a, b) {

return a + b;

}

打包后的包文件:

function add(a, b) {

return a + b;

}

console.log(add(16, 26)); // 42

**代码拆分**

**import()**

将代码拆分引入到应用程序中的最好方法是通过动态 import() 语法。

之前:

import { add } from './math';

console.log(add(16, 26));

以后:

import("./math").then(math => {

console.log(math.add(16, 26));

});

**React.lazy**

React.lazy 函数能让你像渲染常规组件一样处理动态引入（的组件）。

使用之前：

import OtherComponent from './OtherComponent';

使用之后：

const OtherComponent = React.lazy(() => import('./OtherComponent'));

此代码将会在组件首次渲染时，自动导入包含 OtherComponent 组件的包。

示例：

import React, { Suspense } from 'react';

// 当 OtherComponent 要被挂载时，它会从服务器中请求./OtherComponent文件

const OtherComponent = React.lazy(() => import('./OtherComponent'));

function MyComponent() {

return (

<div>

<Suspense fallback={<div>Loading...</div>}>

<OtherComponent />

</Suspense>

</div>

);

}

Suspense组件的fallback 属性接受任何在组件加载过程中你想展示的 React 元素。你可以用一个 Suspense 组件包裹多个懒加载组件。

**基于路由的代码分割**

如下是基于路由的分隔

import React, { Suspense, lazy } from 'react';

import { BrowserRouter as Router, Route, Switch } from 'react-router-dom';

const Home = lazy(() => import('./routes/Home'));

const About = lazy(() => import('./routes/About'));

const App = () => (

<Router>

<Suspense fallback={<div>Loading...</div>}>

<Switch>

// 当访问 / 时，会从服务器中加载 ./routes/Home

<Route exact path="/" component={Home}/>

<Route path="/about" component={About}/>

</Switch>

</Suspense>

</Router>

);