

1、Import() 函数 资源懒加载

2、路由懒加载

首先，可以将异步组件定义为返回一个 Promise 的工厂函数（该函数返回的 Promise 应该 resolve 组件本身）：

```
const Foo = () => .resolve({ /* 组件定义对象 */ })
```

第二，在 Webpack 2 中，我们可以使用[动态 import](#)语法来定义代码分块点 (split point)：

```
import('./Foo.vue') // 返回 Promise
```

注意

如果您使用的是 Babel，你将需要添加 `syntax-dynamic-import` 插件，才能使 Babel 可以正确地解析语法。

结合这两者，这就是如何定义一个能够被 Webpack 自动代码分割的异步组件。

```
const Foo = () => import('./Foo.vue')
```

在路由配置中什么都不需要改变，只需要像往常一样使用 `Foo`：

```
const router = new VueRouter({
  routes: [
    {
      path: '/foo',
      component: Foo
    }
  ]
})
```

把组件按组分块

有时候我们想把某个路由下的所有组件都打包在同个异步块 (chunk) 中。只需要使用 [命名 chunk](#)，一个特殊的注释语法来提供 chunk name（需要 Webpack > 2.4）。

```
const Foo = () => import(/* webpackChunkName: "group-foo" */
  './Foo.vue')
const Bar = () => import(/* webpackChunkName: "group-foo" */
  './Bar.vue')
const Baz = () => import(/* webpackChunkName: "group-foo" */
  './Baz.vue')
```

Webpack 会将任何一个异步模块与相同的块名称组合到相同的异步块中。