

# 前端路由和异步组件

## 1、前端路由的方式 2、异步组件介绍

路由是不同的url, 显示不同的页面或内容的功能.

### 前端路由

hash history

hash

```
http://xxx.abc.com/#/xx

location.hash

function hashAndUpdate () {
  // todo
}

window.addEventListener('hashchange', hashAndUpdate);
```

### history 路由

http://xxx.abc.com/path?param=test

pushState replaceState popstate

```
window.addEventListener('popstate', (event) => {
  console.log("location: " + document.location + ", state: " +
    JSON.stringify(event.state));
});
history.pushState({page: 1}, "title 1", "?page=1");
history.pushState({page: 2}, "title 2", "?page=2");
history.replaceState({page: 3}, "title 3", "?page=3");
history.back();
history.back();
history.go(2);
```

### 实现简单的hash路由和history 路由

demo

### nginx 配置差别

```
## history
location / {
    root    /var/www;
    index index.php index.html index.htm;
    try_files $uri $uri/ /index.html;
}
```

## 异步组件，动态组件

1、为什么使用异步组件 2、怎么使用 3、Vue 实现的原理

定义方式：vue2.0

```
Vue.component('async-example', function (resolve, reject) {
  setTimeout(function () {
    // 向 `resolve` 回调传递组件定义
    resolve({
      template: '<div>I am async!</div>'
    })
  }, 1000)
})
```

```
Vue.component('async-webpack-example', function (resolve) {
  // 这个特殊的 `require` 语法将会告诉 webpack
  // 自动将你的构建代码切割成多个包，这些包
  // 会通过 Ajax 请求加载
  require(['./my-async-component'], resolve)
})
```

```
Vue.component(
  'async-webpack-example',
  // 这个动态导入会返回一个 `Promise` 对象。
  () => import('./my-async-component')
)
```

2.3.0+ 新增

```
const AsyncComponent = () => ({
  // 需要加载的组件（应该是一个 `Promise` 对象）
  component: import('./MyComponent.vue'),
  // 异步组件加载时使用的组件
  loading: LoadingComponent,
  // 加载失败时使用的组件
})
```

```
error: ErrorComponent,  
// 展示加载时组件的延时时间。默认值是 200（毫秒）  
delay: 200,  
// 如果提供了超时时间且组件加载也超时了,  
// 则使用加载失败时使用的组件。默认值是: `Infinity`  
timeout: 3000  
})
```

定义方式: vue3.0

```
const { createApp, defineAsyncComponent } = Vue  
  
const app = createApp({})  
  
const AsyncComp = defineAsyncComponent(  
  () =>  
    new Promise((resolve, reject) => {  
      resolve({  
        template: '<div>I am async!</div>'  
      })  
    })  
)  
  
app.component('async-example', AsyncComp)
```

```
import { defineAsyncComponent } from 'vue'  
  
const AsyncComp = defineAsyncComponent(() =>  
  import('./components/AsyncComponent.vue')  
)  
  
app.component('async-component', AsyncComp)
```

```
import { createApp, defineAsyncComponent } from 'vue'  
  
createApp({  
  // ...  
  components: {  
    AsyncComponent: defineAsyncComponent(() =>  
      import('./components/AsyncComponent.vue')  
    )  
  }  
})
```

vue2

vue3

vue2 实现异步组件原理

回顾总结

hash history 差异、实现 异步组件 原理，使用

感谢大家！