<https://router.vuejs.org/zh-cn/advanced/scroll-behavior.html>

# 滚动行为

使用前端路由，当切换到新路由时，想要页面滚到顶部，或者是保持原先的滚动位置，就像重新加载页面那样。 vue-router 能做到，而且更好，它让你可以自定义路由切换时页面如何滚动。

**注意: 这个功能只在支持 history.pushState 的浏览器中可用。**

当创建一个 Router 实例，你可以提供一个 scrollBehavior 方法：

const router = new VueRouter({

routes: [...],

scrollBehavior (to, from, savedPosition) {

// return 期望滚动到哪个的位置

}

})

scrollBehavior 方法接收 to 和 from 路由对象。第三个参数 savedPosition 当且仅当 popstate 导航 (通过浏览器的 前进/后退 按钮触发) 时才可用。

这个方法返回滚动位置的对象信息，长这样：

* { x: number, y: number }
* { selector: string, offset? : { x: number, y: number }} (offset 只在 2.6.0+ 支持)

如果返回一个 falsy (译者注：falsy 不是 false，**[参考这里](https://developer.mozilla.org/zh-CN/docs/Glossary/Falsy" \t "_blank)**)的值，或者是一个空对象，那么不会发生滚动。

举例：

scrollBehavior (to, from, savedPosition) {

return { x: 0, y: 0 }

}

对于所有路由导航，简单地让页面滚动到顶部。

返回 savedPosition，在按下 后退/前进 按钮时，就会像浏览器的原生表现那样：

scrollBehavior (to, from, savedPosition) {

if (savedPosition) {

return savedPosition

} else {

return { x: 0, y: 0 }

}

}

如果你要模拟『滚动到锚点』的行为：

scrollBehavior (to, from, savedPosition) {

if (to.hash) {

return {

selector: to.hash

}

}

}

我们还可以利用[**路由元信息**](https://router.vuejs.org/zh-cn/advanced/meta.html)更细颗粒度地控制滚动。查看完整例子请[**移步这里**](https://github.com/vuejs/vue-router/blob/next/examples/scroll-behavior/app.js)。

### 异步滚动

**2.8.0 新增**

你也可以返回一个 Promise 来得出预期的位置描述：

scrollBehavior (to, from, savedPosition) {

return new Promise((resolve, reject) => {

setTimeout(() => {

resolve({ x: 0, y: 0 })

}, 500)

})

}

将其挂载到从页面级别的过渡组件的事件上，令其滚动行为和页面过渡一起良好运行是可能的。但是考虑到用例的多样性和复杂性，我们仅提供这个原始的接口，以支持不同用户场景的具体实现。