

## 触发钩子的完整顺序：

将路由导航、`keep-alive`、和组件生命周期钩子结合起来的，触发顺序，假设是从a组件离开，第一次进入b组件：

1. `beforeRouteLeave` :路由组件的组件离开路由前钩子，可取消路由离开。
2. `beforeEach` :路由全局前置守卫，可用于登录验证、全局路由loading等。
3. `beforeEnter` :路由独享守卫
4. `beforeRouteEnter` :路由组件的组件进入路由前钩子。
5. `beforeResolve` :路由全局解析守卫
6. `afterEach` :路由全局后置钩子
7. `beforeCreate` :组件生命周期，不能访问 `this`。
8. `created` :组件生命周期，可以访问 `this`，不能访问dom。
9. `beforeMount` :组件生命周期
10. `deactivated` :离开缓存组件a，或者触发a的 `beforeDestroy` 和 `destroyed` 组件销毁钩子。
11. `mounted` :访问/操作dom。
12. `activated` :进入缓存组件，进入a的嵌套子组件(如果有的话)。
13. 执行beforeRouteEnter回调函数next。

### 完整的路由导航解析流程(不包括其他生命周期):

1. 触发进入其他路由。
2. 调用要离开路由的组件守卫 `beforeRouteLeave`
3. 调用局前置守卫: `beforeEach`
4. 在重用的组件里调用 `beforeRouteUpdate`
5. 调用路由独享守卫 `beforeEnter`。
6. 解析异步路由组件。
7. 在将要进入的路由组件中调用 `beforeRouteEnter`
8. 调用全局解析守卫 `beforeResolve`
9. 导航被确认。
10. 调用全局后置钩子的 `afterEach` 钩子。
11. 触发DOM更新(`mounted`)。
12. 执行 `beforeRouteEnter` 守卫中传给 next 的回调函数