

1. 路由系统选择策略?  
控制点. 基于钩子 -

## 2. 流程逻辑

- `Vue.use(VueRouter)`
  - 执行 `VueRouter` 的 `install` 方法
    - 缓存 `Vue` 实例
    - 通过 `mixin`, 注册 `beforeCreate`
    - 原型上挂载 `$router` 为 `Vue` 的私有属性 `_router` (此时可能没值)
    - 原型上挂载 `$route` 为 `Vue` 的私有属性 `_route` (此时可能没值)
    - 注册 `RouterLink` 组件
    - 注册 `RouterView` 组件
- 执行 `new Router(routes)`
  - 缓存传入的 `routes` 规则到 `_routes`
  - 创建 `macher` 对象
    - 生成 `map`
    - 构造 `match` 方法
    - 构造 `addRoute` 方法
  - 根据 `mode`, 创建 `history` 对象
    - `new HashHistory(this)`
      - 缓存传入的 `this (router)` 到 `router`
      - 通过 `createRouter (path为初始"/")` 创建 `current` 对象
      - 创建 `cb` 函数, 初始值赋值为 `null`
      - 初始化浏览器当前 `hash` 为 `"/"`
- `new Vue` 时传
- 在 `Vue` 的 `beforeCreate` 钩子中
  - 根据当前实例是否传入 `router` 对象
    - 传入
      - 将当前 `vue` 实例缓存到 `vue` 的 `_routerRoot` 私有变量中
      - 将传入的 `router` 对象缓存到 `_router` 私有变量中
      - 执行传入的 `router` 的 `init` 方法
        - 调用 `history` 的 `transitionTo` 方法, 传入当前路径和完成时回调
          - 将当前 `current` 赋值为传入路径对应的 `record`
          - `cb` 存在时调用 `cb` (此时尚赋值)
          - 完成时回调存在时, 调用完成时回调 (此时存在)
            - 监听 `hashchange` 事件, 事件为 `tansionTo(currentPath)` (未传入第二个参数)
        - 调用 `history` 的 `listen` 方法, 给 `cb` 赋值
        - 在当前 `vue` 对象上挂载响应式私有变量 `_route` 为传入的 `router` 对象 `history` 属性的 `current` 属性
  - 在 `hash` 变化时, 调用 `hashchange` 注册的回调