

搭建过程:

```
npm install --global vue-cli
```

```
vue init webpack 名字(vuedemo)
```

进入上面的文件夹后安装依赖cnpm install

(1) main.js是主文件，也是文件入口，他将app实例挂在到index.html的<div id="app"></div>中，

即:

```
new Vue({  
  el: '#app',           //找到主ID  
  router,               //加载路由，注入到跟实例中  
  template: '<App/>',    //添加模板标签  
  components: { App }   //在模板中插入组件  
})
```

(2) app.vue 存放三个东西: 1. 模板template 2. script 导出名字 3. style

app.vue 本身就是一个组件，但是他可以继续引入其他组件（方法是通过路由引入其他组件）

<router-link to="">相当于<a>标签，to后面跟着的是地址，这个地址就是routes(路由文件)中定义的路径（随意取的），他负责引入路由配置导航

<router-link>相当于htmlde <a>, 在vue中就相当于路由<router-view>, <router-view>只有一个时才靠谱

(3) router 主要是指定地址并分配路径名字——path和名字——name具体形式如下:

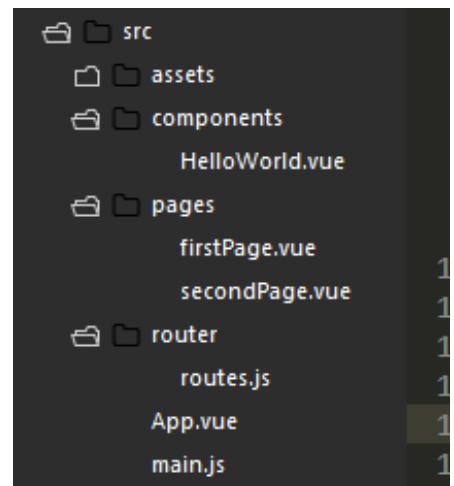
```
{  
  path: '/second',  
  name: '2',  
  component: secondPage  
}
```

其中secondPage 是组件，通过import secondPage from “具体路径” 的方式 导入，定义的path供app.vue的<router-link>使用，name没啥用;

(4) 其他的.vue文件都可以看做是组件，包括，template, script, style, 其中script主要是export default {} 数据，其中的name一般无效。

(5) 目录结构

app.vue 入口文件
main.js 主配置文件
.vue 组件
routes.js 路由文件



(6) vue-router 懒加载

重定向问题：在routes文件中，

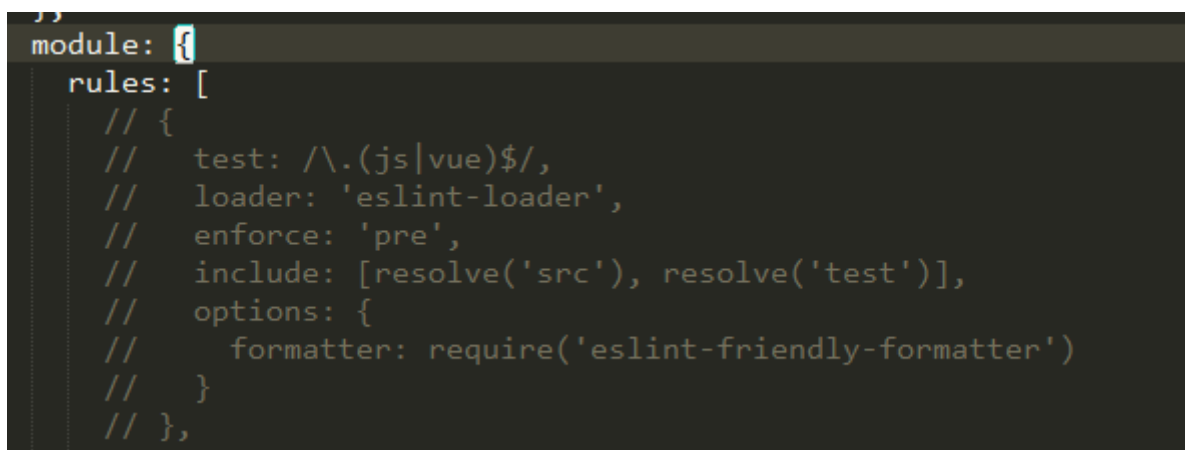
```
routes: [  
  {  
    path: '/',  
    name: 'Hello',  
    component: Hello  
  }  
]
```

'/' 当首次进入页面时，页面没有显示任何组件，让页面一进来就显示这个页面（组件）

通过 `<router-view></router-view>` 标签默认展示

踩坑

1. 一般情况下，vue-cil 对格式要求比较严格，`export default{ }` 很容易出错，此时注释掉 `webpack.base.config.js` 文件中的（下图）即可。



2. vue-router 一个文件可以有多个vue-router，但是需要区分。通过name方法：

app. vue中：

```
</div>
<router-view name="a"></router-view>
<router-view name="b"></router-view>
</div>
```

router中：

```
routes: [
  {
    path: '/',
    components: {
      default: Hello,
      a: Hello,
      b: firstPage,
    },
  },
]
```

默认显示hello
组件

name='a'时，显
示Hello组件

3. v-cloak 用来解决vue页面加载闪烁的问题，作用是加载完成后再显示。v-cloak是css样式，需要在css中定义

```
[v-cloak] {
  display: none;
}
```

4.