

Vue3整合Echarts流程

创建项目

```
vue create 项目名称
```

安装项目所需依赖

```
cd 项目名称  
cnpm install
```

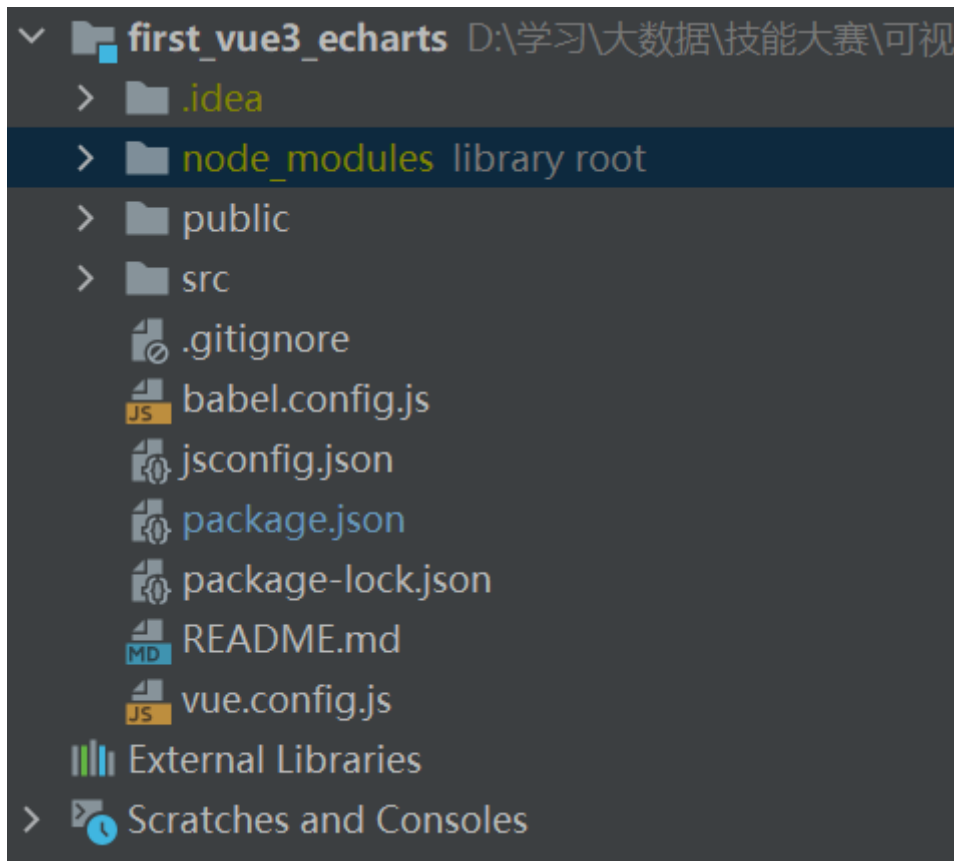
安装项目所需功能依赖

```
cnpm install echarts  
cnpm install axios  
cnpm install vue-router
```

- echarts可视化展示
- axios异步通信
- vue-router配置路由

使用编辑器打开项目

查看项目结构



- node_modules: 存放项目所需功能依赖
- public: 存放公开的静态文件

- src: 存放主代码文件

向main.js入口文件中配置相关功能模块

全局引入echarts

```
import * as echarts from 'echarts'  
// 配置全局使用echarts  
app.config.globalProperties.$echarts = echarts
```

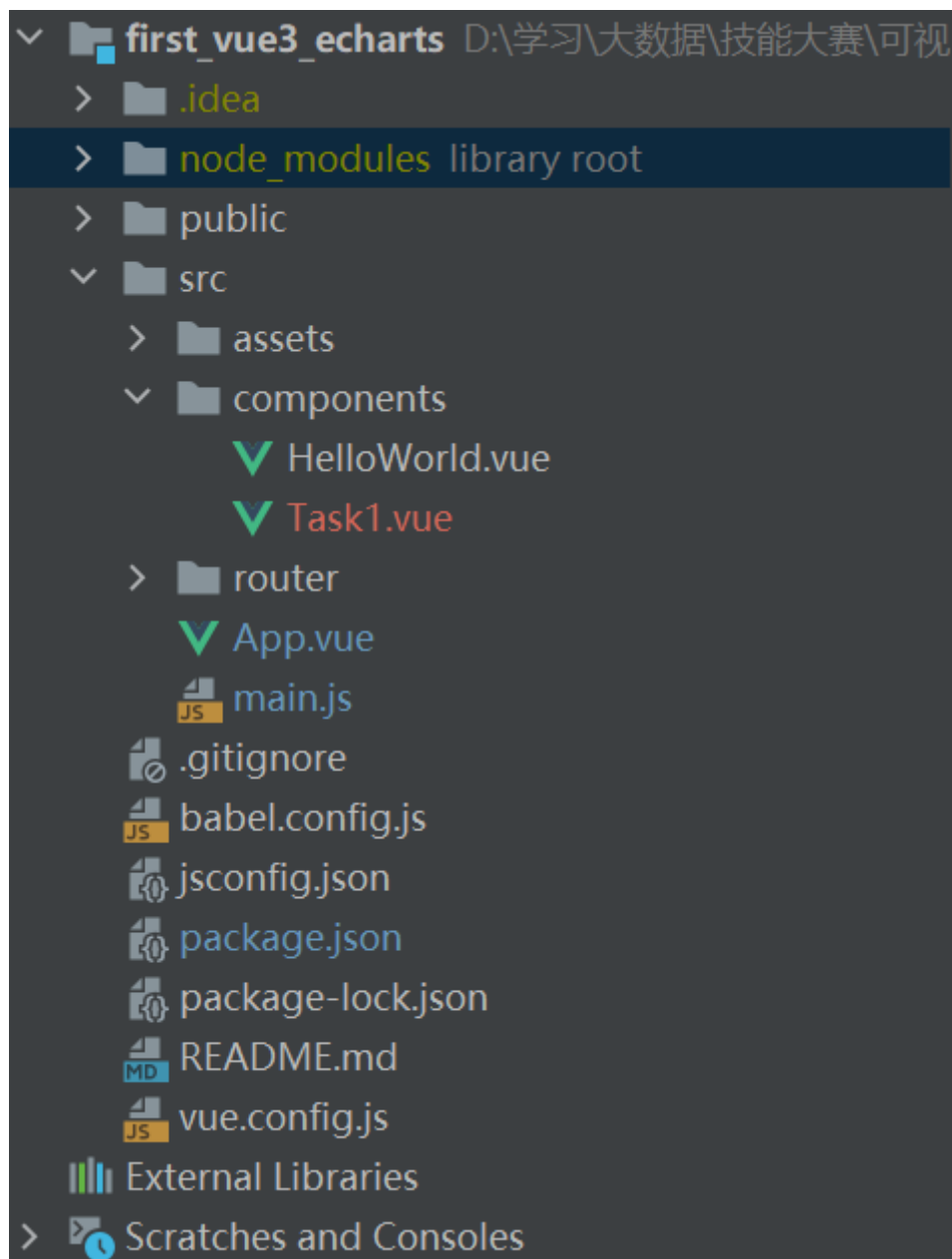
全局引入axios

```
import axios from 'axios'  
// 配置全局使用axios  
app.config.globalProperties.$axios = axios
```

main.js全部代码

```
import { createApp } from 'vue'  
import App from './App.vue'  
import * as echarts from 'echarts'  
import axios from 'axios'  
  
const app = createApp(App)  
// 配置全局使用  
app.config.globalProperties.$axios = axios  
app.config.globalProperties.$echarts = echarts  
app.mount('#app')
```

根据任务数量在项目路径下的src/components目录下创建component



注意：这里的router是后期创建的，新建的项目是没有的！！

新建组件源代码

```
<template>

</template>

<script>
export default {
  name: "Task2"
}
</script>

<style scoped>

</style>
```

在template中添加div标签作为echarts画布

```
<template>
  <div id="main" style="width:600px;height:600px;"></div>
</template>
```

在export default中添加mounted生命周期方法

```
<script>
export default {
  name: "Task2",
  mounted() {

  }
}
</script>
```

在mounted中添加Echarts图表相关配置

```
<script>
export default {
  // eslint-disable-next-line vue/multi-word-component-names
  name: "Task1",
  mounted() {
    var option = {
      title:{
        text:'测试'
      },
      xAxis:{
        type:'category',
        data:[]
      },
      yAxis:{
        type:'value'
      },
      series:[
        {
          type:'bar',
          data:[]
        }
      ]
    }
  }
}
</script>
```

注意：这里绘图相关数据置空是为了后续异步通信！！

在mounted中实现异步请求获取数据

```
<script>
export default {
  // eslint-disable-next-line vue/multi-word-component-names
  name: "Task2",
```

```

mounted() {
  var option = {
    title:{
      text:'测试'
    },
    xAxis:{
      type:'category',
      data:[]
    },
    yAxis:{
      type:'value'
    },
    series:[
      {
        type:'bar',
        data:[]
      }
    ]
  }
  this.$axios.get("请求的后端接口").then(response => {
    option.xAxis.data = response.data.data[0]
    option.series[0].data = response.data.data[1]
  })
}
}
</script>

```

在axios异步请求代码体中添加Echarts图表初始化相关操作

```

<script>
export default {
  // eslint-disable-next-line vue/multi-word-component-names
  name: "Task2",
  mounted() {
    var option = {
      title:{
        text:'测试'
      },
      xAxis:{
        type:'category',
        data:[]
      },
      yAxis:{
        type:'value'
      },
      series:[
        {
          type:'bar',
          data:[]
        }
      ]
    }
    this.$axios.get("请求的后端接口").then(response => {
      option.xAxis.data = response.data.data[0]
      option.series[0].data = response.data.data[1]
    })
  }
}

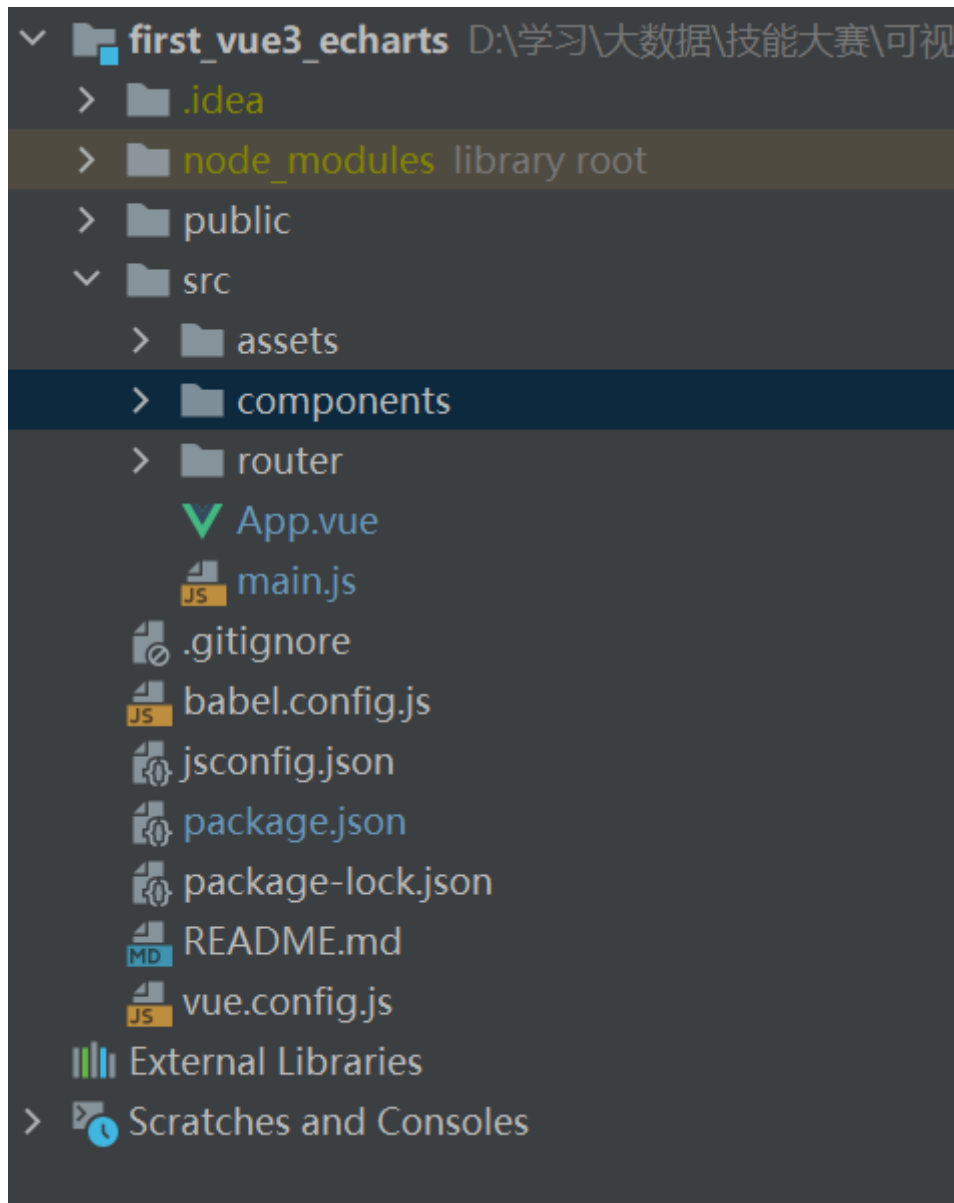
```

```
var mychart = this.$echarts.init(document.getElementById("main"))
mychart.setOption(option)
})
}
}
</script>
```

注意：这里为什么是在axios代码体中添加图表初始化操作，主要是因为axios异步通信会有延迟！！

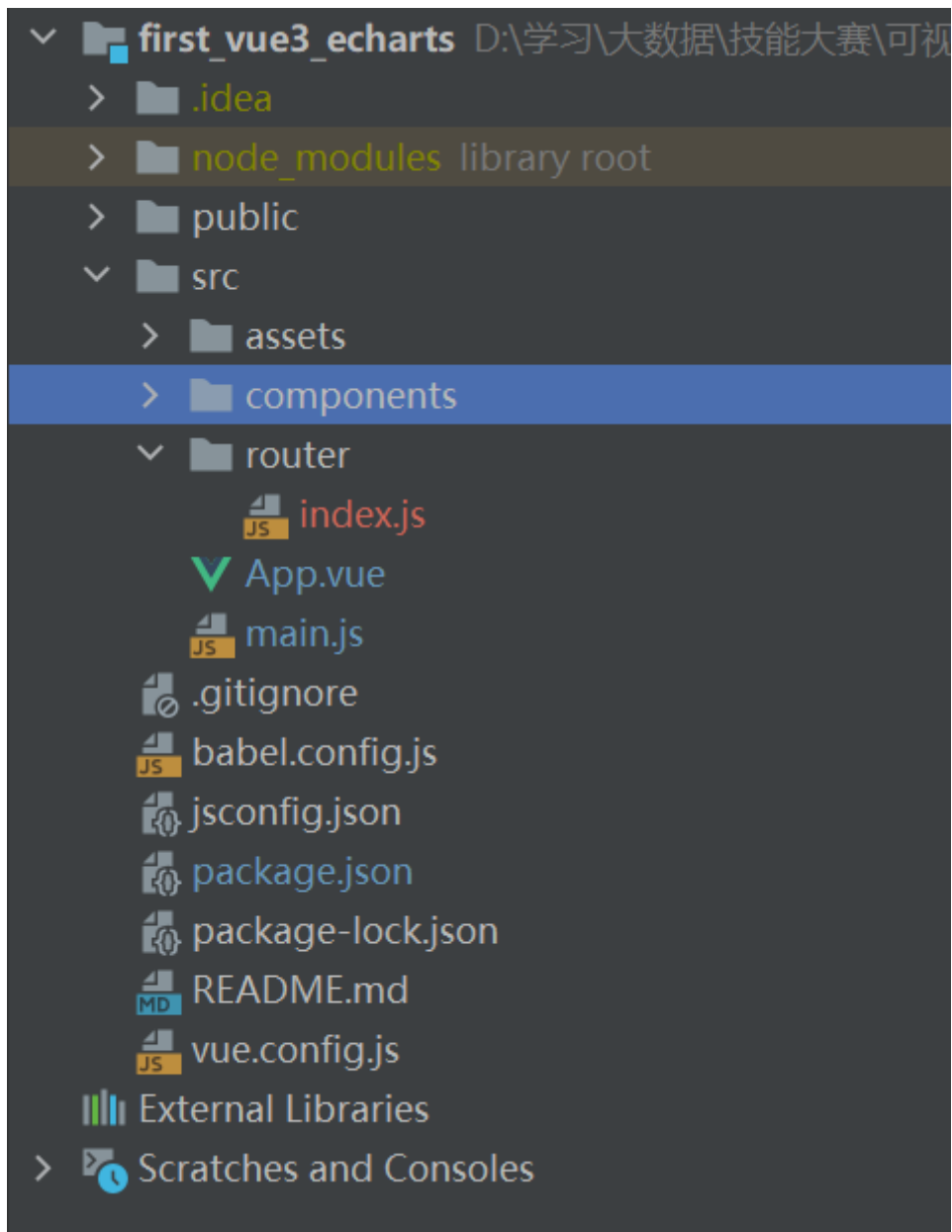
为创建好的component配置路由

在项目路径下src/目录下创建路由文件夹



注意：对于新建项目是没有router文件夹的！！

在router文件夹下新建一个空白index.javascript文件



从vue-router中引入配置路由必须的两个功能

```
import {createRouter, createWebHistory} from "vue-router";
```

引入要配置路由的component

```
import Task1 from "@/components/Task1";
```

为component配置路由

```
const routes = [  
  {path: '/task1', component: Task1}  
]
```

根据配置好的路由实例化路由器

```
const router = createRouter({  
  routes,  
  history: createWebHistory()  
})
```

导出路由器以供使用

```
export default router
```

index.js完整代码

```
import {createRouter, createWebHistory} from "vue-router";

import Task1 from "@/components/Task1";

const routes = [
  {path: '/task1', component: Task1}
]

const router = createRouter({
  routes,
  history: createWebHistory()
})

export default router
```

在main.js文件中配置使用路由

```
import { createApp } from 'vue'
import App from './App.vue'
import * as echarts from 'echarts'
import axios from 'axios'
import router from './router'

const app = createApp(App)
app.use(router)
// 配置全局使用
app.config.globalProperties.$axios = axios
app.config.globalProperties.$echarts = echarts
app.mount('#app')
```

修改app.vue代码

将template视图原内容修改为router-view标签对

```
<template>
  <router-view></router-view>
</template>
```

去除所有导入component, 删除导出components表中的component


```
<template>
  <router-view></router-view>
</template>
<script>

export default {
  name: 'App',
  components: {
  }
}
</script>
```

注意：这里为什么去除所有引入的component，主要是为了规避component注册不使用的問題！！

运行项目

在命令行输入：

```
npm run serve
```

注意：这里启动的是前端服务，而根据代码可以知道在启动前端服务前应该先启动后端服务！！