

前后端通信说明文档

by 2154286 郑伟丞

前后端采用API实现通信，前端发送和接收，后端监听和回复，这里涉及到两个端口：一个是启动前端网页运行的本地端口，一个是后端要监听的端口。以第一个发在群里的前端项目为例，启动网页地址为 `http://localhost:8080`，同时还有个 `target` 字段给出了另一个地址：
`target: 'http://localhost:5000'`，这个就是后端监听的端口，前端发送API请求时会往 `5000` 的端口发送

前端

前面提到的两个端口在 `config/index.js` 里面配置好了，所以在实际的发送API请求时不需要携带目标地址。但这只是针对那个项目而言，否则需要在API路由前面加上目标地址。

在每个页面下方会有各种处理API请求的函数，如这里的 `trackByImg()`

```
let formData = new FormData()
// let fileType = this.beforeTrackImg.raw.type.split('/')[0]
let suffix = this.beforeTrackImg.name.split('.')
formData.append('file', this.beforeTrackImg)
formData.append('type', this.chosenType)
// formData.append('fileType', fileType)
formData.append('suffix', suffix[suffix.length - 1])
console.log('开始检测')
this.refreshProgress(300)
trackByImg(formData).then(res => {
  this.trackingPercentage = 100
  console.log(res)
  this.afterTrackImgUrl = window.URL.createObjectURL(res.data)
  this.afterTrackImg = new File([res.data], 'result', {type: res.data.type})
  this.trackFileSize = this.convertFileSize(this.afterTrackImg.size)
  console.log(this.afterTrackImg)
  this.successTrack = true
  this.trackStatue = 0
  this.$message({
    message: '检测成功',
    type: 'success'
  })
})
})
```

这里将需要发送给后端的数据封装到 `formData` 里面，待后端接收后再根据关键词提取

`trackByImg()` 的具体实现在 `src/api/api.js`

```
export const trackByImg = (data) => {
  return request({
    method: 'post',
    url: '/api/Track/trackByImg',
    responseType: 'blob',
    data: data
  })
}
```

这里就可以看到API路由的设计了，即: `/api/Track/trackByImg`

如果没有配置好默认的目标地址，则可能是: `http://localhost:5000/api/Track/trackByImg`

后端

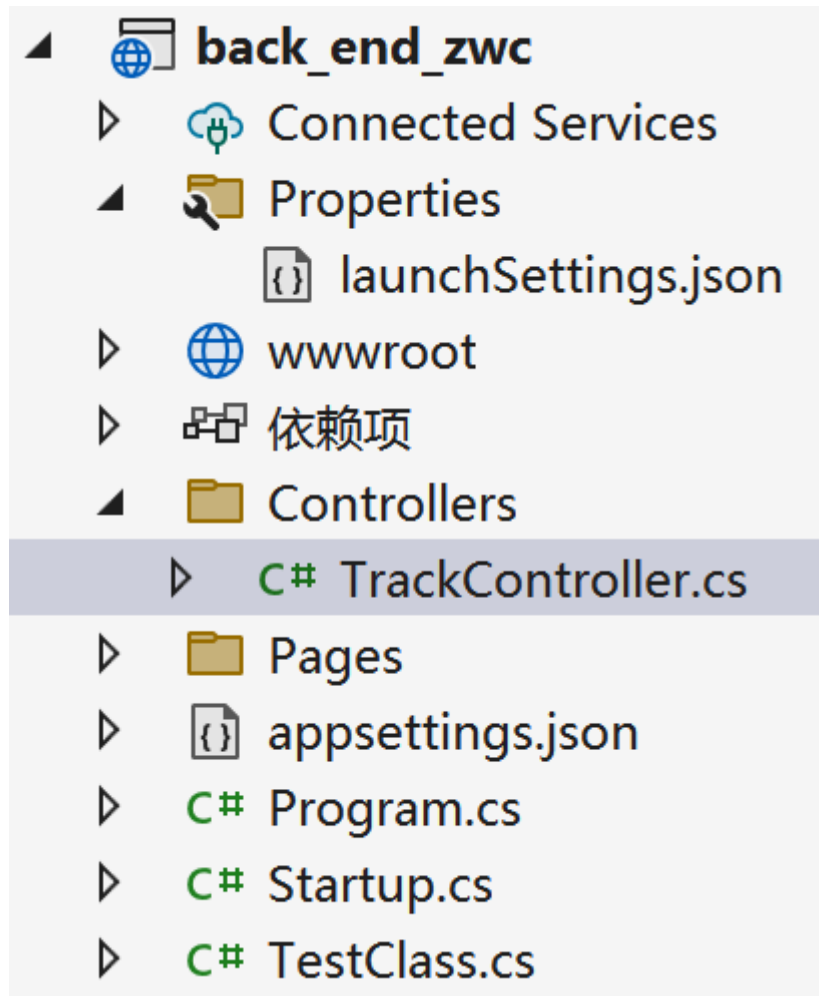
设置监听端口



`Startup.cs` 据说是会创建自动生成，没有的话自己创建一下，里面的配置应该会比较统一，这个询问AI即可

最重要的是API控制器，也就是说明了后端监听到什么样的API要做出什么样的反应

创建一个 `Controllers` 文件夹



TrackController.cs

```
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using System.IO;
using System.Threading.Tasks;

namespace back_end_zwc.Controllers
{
    [ApiController]
    [Route("api/[controller]")]
    public class TrackController : ControllerBase
    {
        private readonly ILogger<TrackController> _logger;

        public TrackController(ILogger<TrackController> logger)
        {
            _logger = logger;
        }

        [HttpGet]
        public ActionResult<string> Get()
        {
            return "Hello, World!VUE!!!";
        }

        [HttpPost("trackByImg")]
        public async Task<ActionResult> TrackByImg([FromForm] IFormFile file, [FromForm] string type, [FromForm] string suffix)
        {
            _logger.LogInformation($"Received file: {file.FileName}, Type: {type}, Suffix: {suffix}");
        }
    }
}
```

后端想正确监测指定路由要关注这三个红框，同样这里不用再关注监听的地址了

能监测到的API路由基本就是由这三个部门构成了, `api+Track+trackByImg`

也就是Post类型的 `api/Track/trackByImg` 请求

最值得一提的是, 这里的 `TrackController` 类的名字不是乱取的, 必须以 `Controller` 结尾, 且前半段的名称是API路由的一部分, 也就是 `Track`