

## 项目架构以及前后端开发

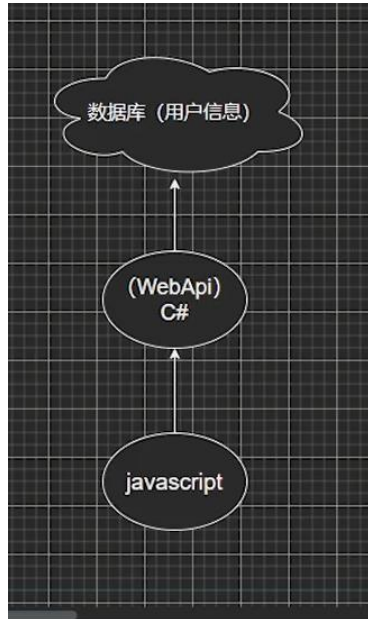
### 0. 参考视频:

[C#/.NET BBS 论坛项目开发全程实录 \(WebAPI、ASP.NET Core、EfCore、VUE、ElementUI、SqlServer、MVC\)](#) [哔哩哔哩 bilibili](#)

文档觉得抽象就看这个，讲的很细包教包会，看到前 12P 就能上手了

### 1. 项目架构

采用前后端分离的架构，前端基于 Vue 框架，后端基于 WebApi，通俗地说就是 WebApi 是前端界面与数据库本身的桥梁，负责在二者之间传递请求与数据、实现各种功能。



(1) **Vue**: 一种前端框架，可以方便地把前端各种元素集合在一起。

目录结构:

我们要编写的是 **src** 文件夹里的内容，各子文件夹存放内容如下

#### **src**

存放 **vue** 项目的源代码。其文件夹下的各个文件（文件夹）分别为：

**assets** : 资源文件，比如存放 **css**，图片等资源。

**component** : 组件文件夹，用来存放 **vue** 的公共组件（注册于全局，在整个项目中通过关键词便可直接输出）。

**router** : 用来存放 **index.js**，这个 **js** 用来配置路由

**tool** : 用来存放工具类 **js**，将 **js** 代码封装好放入这个文件夹可以全局调用（比如常见的 **api.js**，**http.js** 是对 **http** 方法和 **api** 方法的封装）。

**views** : 用来放主体页面，虽然和组件文件夹都是 **vue** 文件，但 **views** 下的 **vue** 文件是可以用来充当路由 **view** 的。

**main.js** : 是项目的入口文件，作用是初始化 **vue** 实例，并引入所需要的插件。

**app.vue** : 是项目的主组件，所有页面都是在该组件下进行切换的。

注:

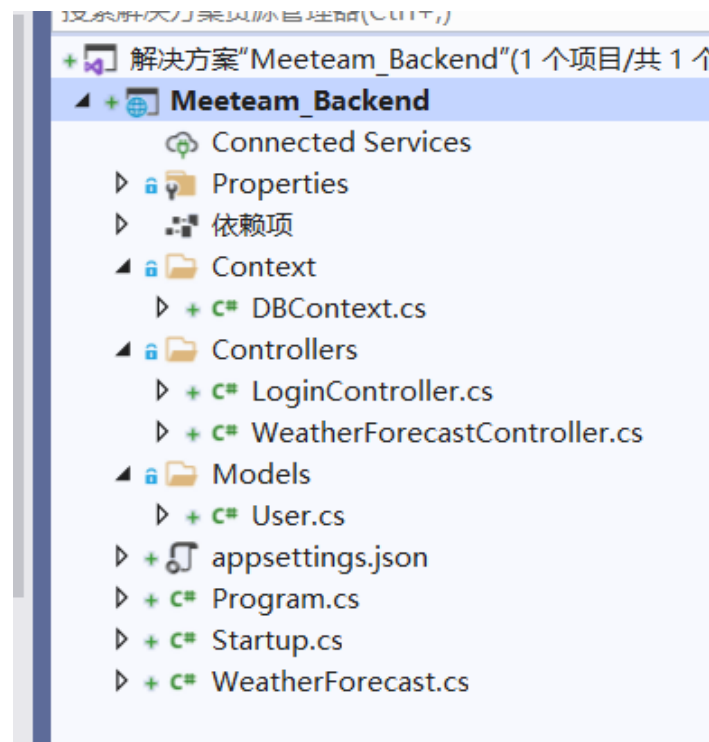
目前的代码里 View 里还没有东西，大家写前端代码时候别把东西全放 Components 里，感觉会复用比较多的组件（导航栏等等）放在 Components 里。

components是小组件  
containers是容器级组件  
views是页面级组件

也就是说，views是页面级组件，components是小组件，小组件可被引用在views中，一般views组件不被复用【containers是容器级组件（根据项目大小决定是否使用）】

## （2）WebApi: 后端

### 文件夹结构



我们主要需要编写的有以下三个文件夹：

**Context:** 存放与数据库数据交互的方法和属性（可以看一下学长的代码，感觉他们在这里使用了 **ORM** 框架，我们目前还是处于硬写 **SQL** 查询的状态）

**Controller:** 与前端交互的部分写在这里面

**Models:** 感觉应该是不与前端数据交流，但可能会被 Controller 调用的属性和方法写在这里

另外：

**Properties:** launchSettings.json 为本地调试时候 VS 的配置，可以自行修改

## 2. 前后端通信

前端：axios

(注意在本地调试时候 axios 后面写的 url 需要改成在后端-Properties 里面填写的地址，现在这个是服务器上的，我觉得这里可以封装一下，免得每次都要改)

```

axios.get("http://106.14.193.8:8090/Login/login?user_id="+this.user+"&password="+this.pass).then
    if(res.data===false){
        vm.$message.error("登录失败");
        vm.resetForm(formName);
    }else{
        vm.$message.success("登录成功");
        vm.$router.push({path: "/users/InforList"}); //接下来进入到哪个路由
    }
})

```

后端：对应的 Controller

```

//路由规则http://根目录/[Controller]/[action]
//返回的，开头2代表正确，最后4表示没有数据
[Route("/[Controller]/[action]")]
[ApiController]
[EnableCors("any")]
0 个引用
public class LoginController : ControllerBase
{
    //登陆
    [HttpGet]
    0 个引用
    public bool login(string user_id, string password)
    {
        UserInfo info = new UserInfo();
        string real_psw = info.get_Password(user_id);
        if (real_psw == null || real_psw != password)
            return false;
        else
            return true;
    }
}

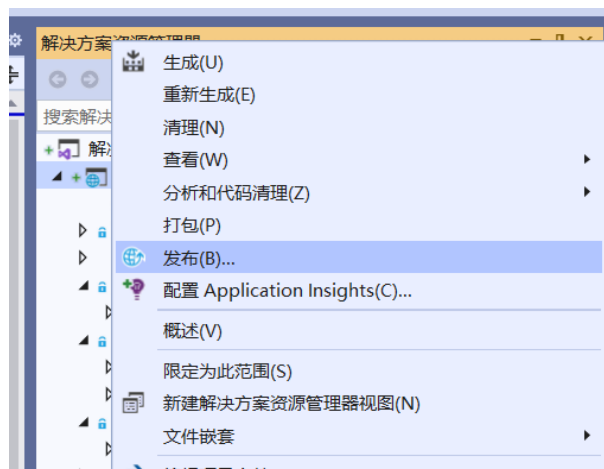
```

相关问题

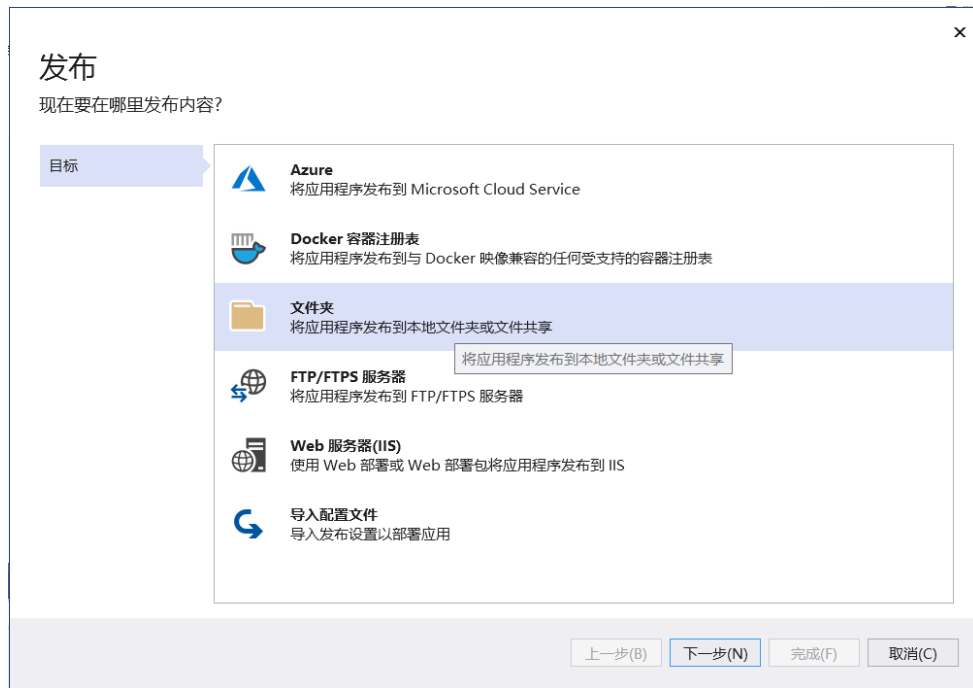
### 3. 把项目部署到服务器上

(1) WebApi 部分：

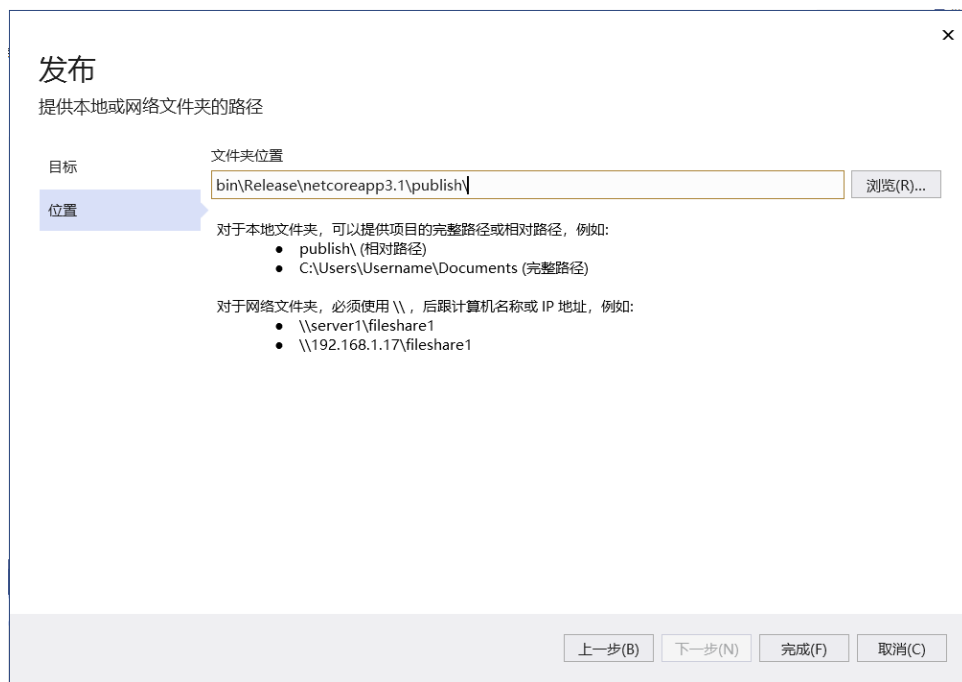
右键项目，点发布



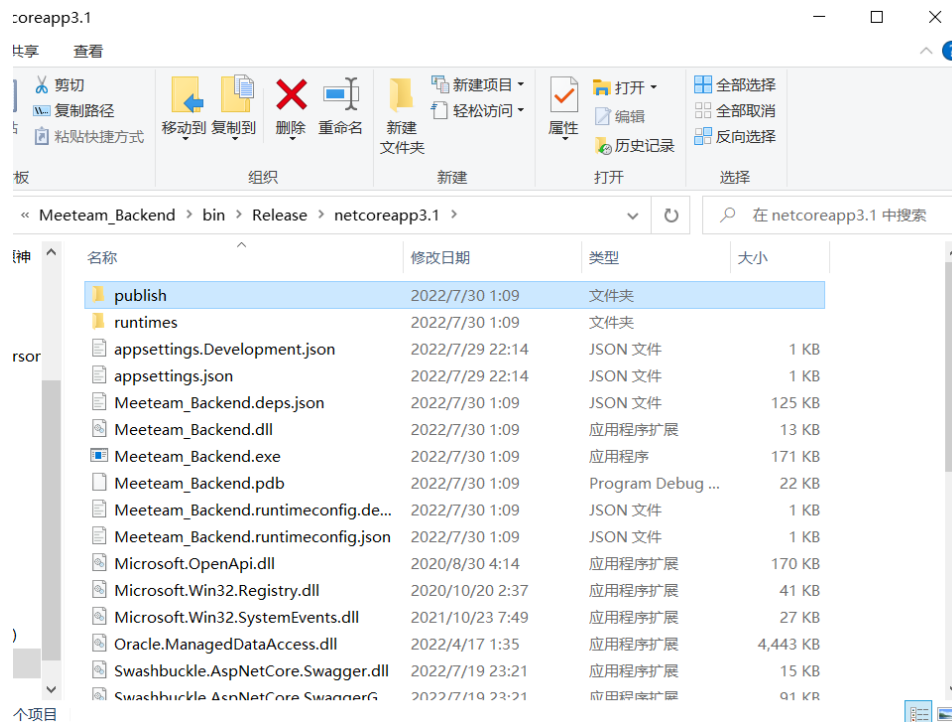
选择文件夹



点完成就行

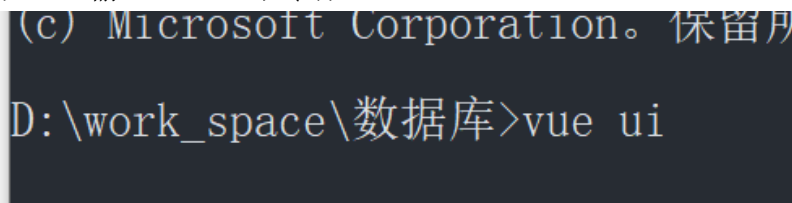


此时对应目录下出现了一个 **publish** 文件夹，这个就是我们需要部署的后端文件。



## (2) 前端

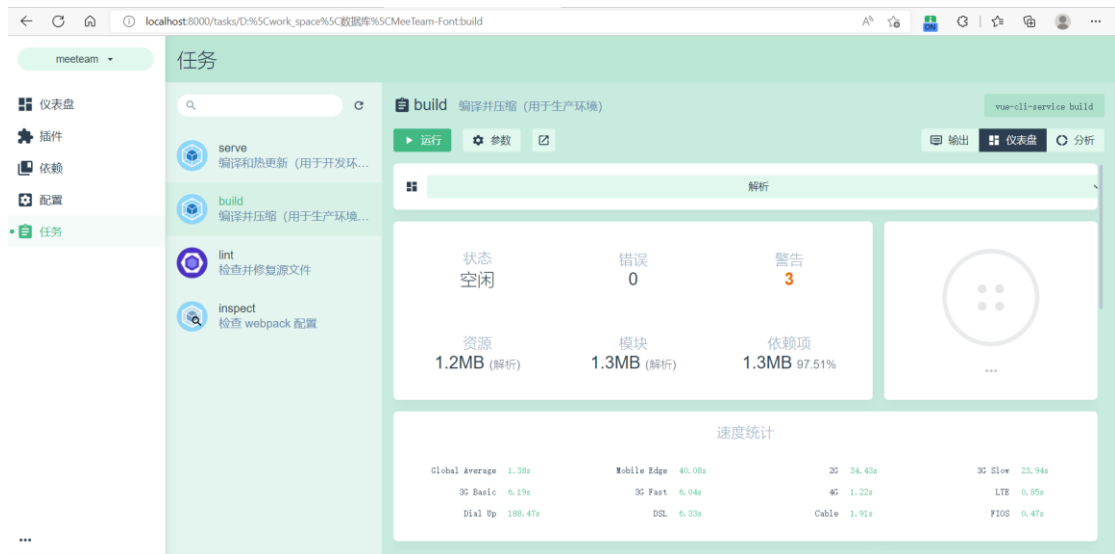
这里使用 vue ui 的方式，命令行方式也可以  
打开 cmd 输入 vue ui 命令行



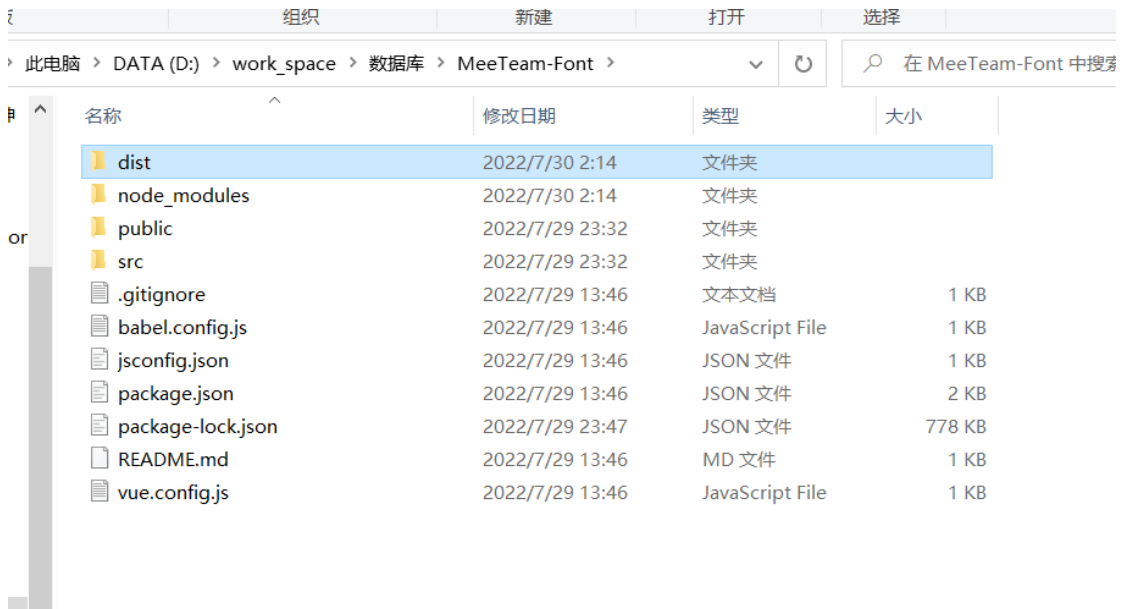
选择对应的 vue 项目



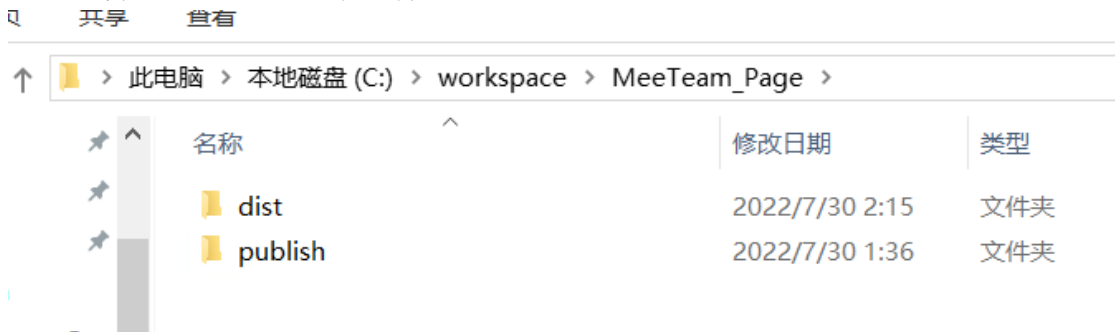
选择任务-build-运行



此时项目文件夹下出现了一个 **dist** 文件夹，即为我们要部署的前端文件



连接服务器，把上述两个文件夹复制到这里即可



### (3) 访问测试

在外部访问 <http://106.14.193.8:8081/>

能看到我们的网站，并且可以交互

登录

账号

密码

提交

重置

登录

账号normal

密码.....

提交

重置

🔍 不安全   106.14.193.8:8081/#/users/IntorList		
日期	姓名	项目名称
✔ 登录成功		
暂无数据		