# 环境准备

## 安装Node.js

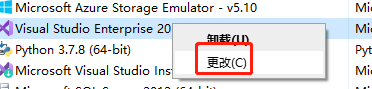
安装包位置：

https://106.12.252.166:8443/svn/SanjelDocuments/trunk/GuideLines/Lab Test Database开发环境搭建/node-v12.20.1-x64.msi

安装目录选择D:\Nodes\

## 安装.NetCore2.2SDK

控制面板->程序­->程序和功能



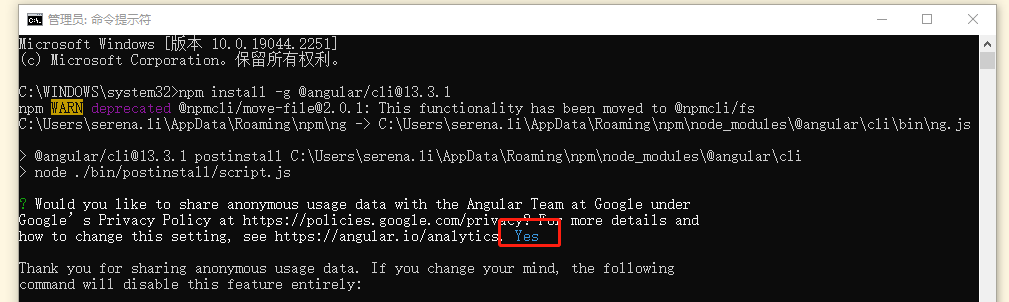


## 安装Angular

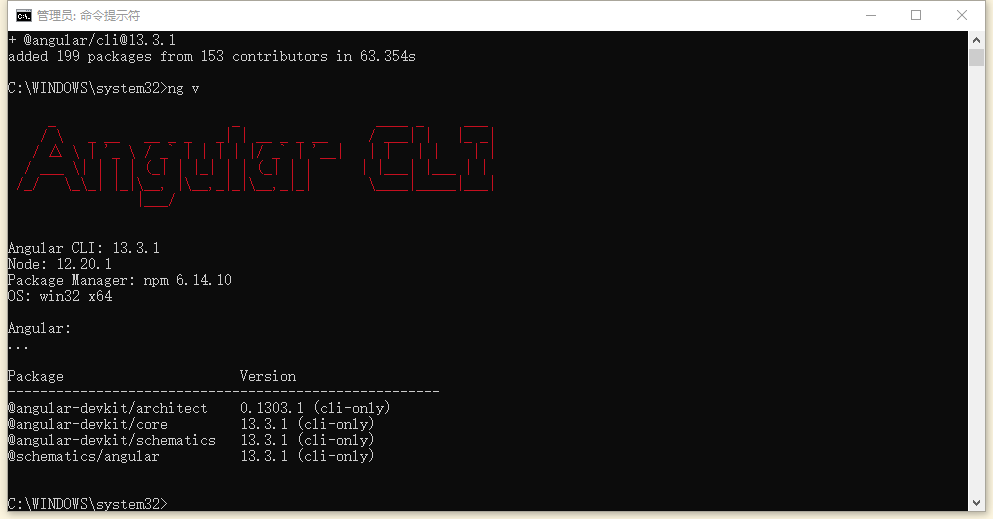
以管理员身份打开命令行，执行命令：

npm install -g @angular/cli@13.3.1

输入Y



安装完成后输入命令ng v显示如下：



安装成功。

# 代码运行

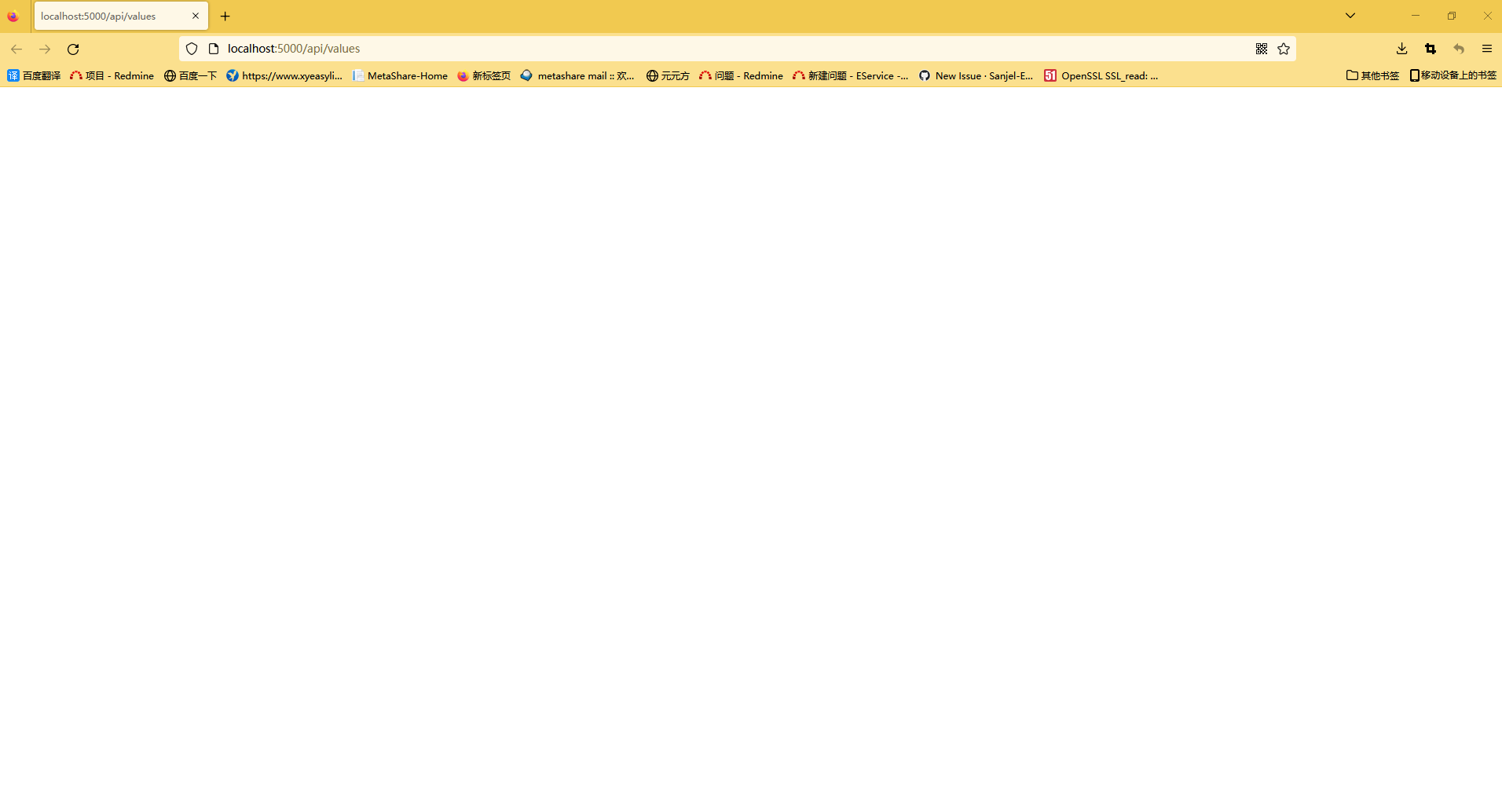
## 后台代码运行

修改配置文件：

Sanjel-Energy-Services\LabDatabase\backend\SldApi\appsettings.json

修改数据库连接字符串后保存。

安装完成后重新运行VS中的SldBackEnd.sln项目运行结果如下：



以上结果表示应用正常运行。

## 运行前台代码

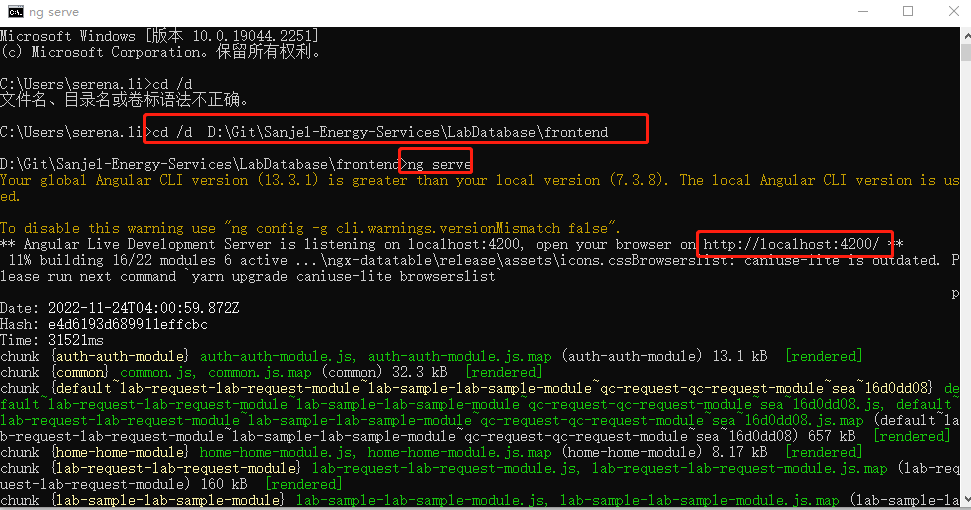
拷贝node\_modules.zip文件到Sanjel-Energy-Services\LabDatabase\frontend到目录下并解压。

此目录存放的是前台代码运行所需要的组件信息，安装时会有失败，node\_modules.zip是比较完整的插件包，直接解压使用即可。

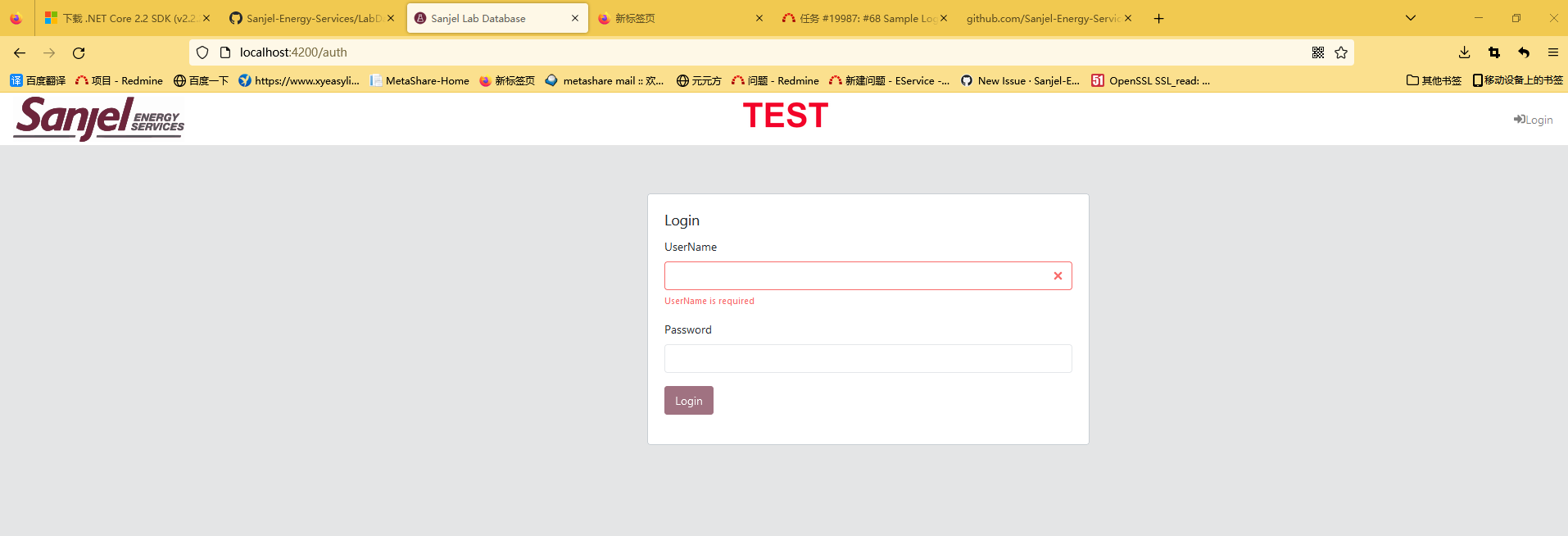
在命令行中，进入到前台代码路径，

cd /d D:\Git\Sanjel-Energy-Services\LabDatabase\frontend

ng serve



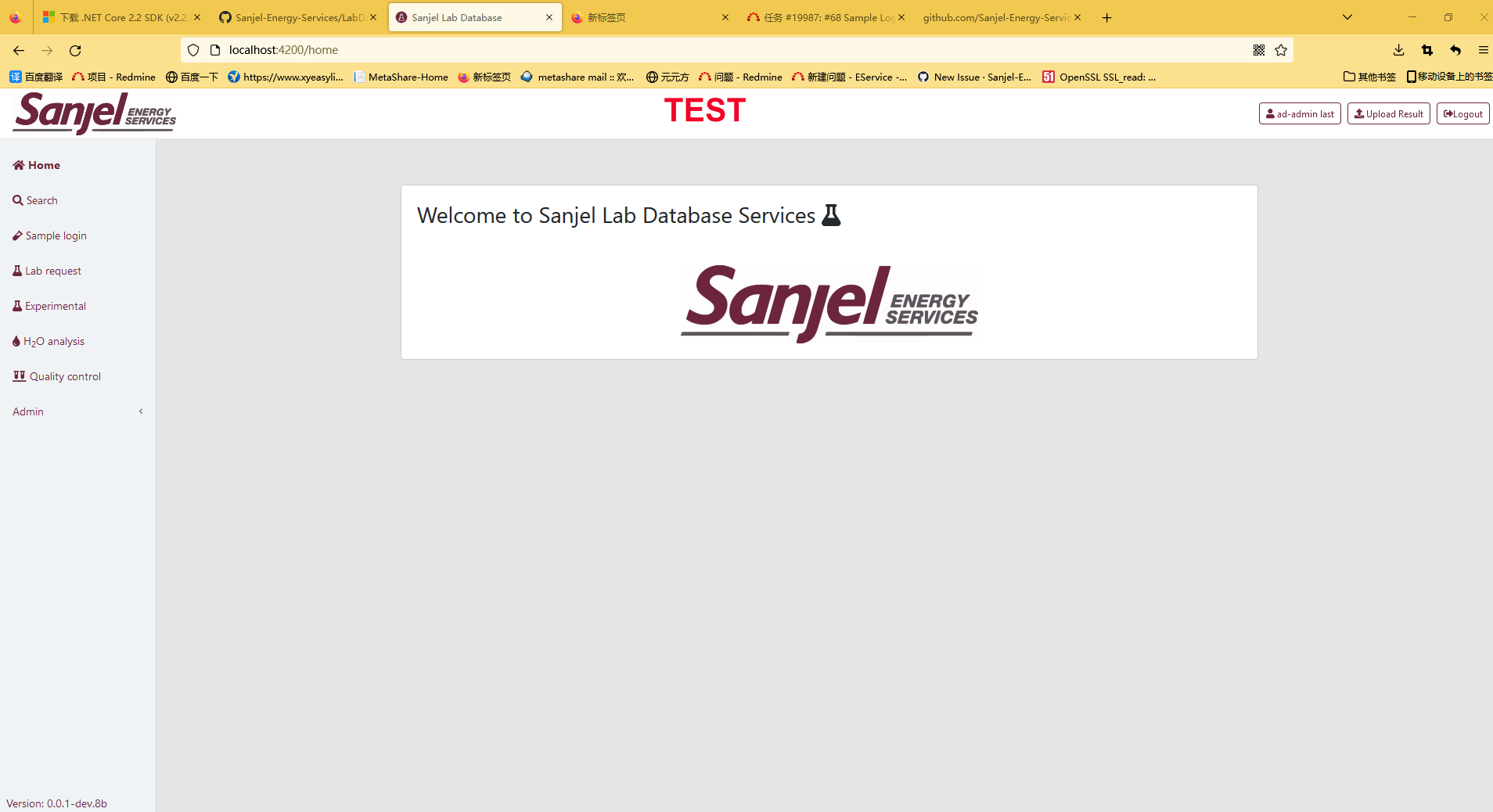
根据提示的URL访问



用户名：ad-admin

密码：pud

登录成功页面如下：



# 代码打包

## 前台代码打包

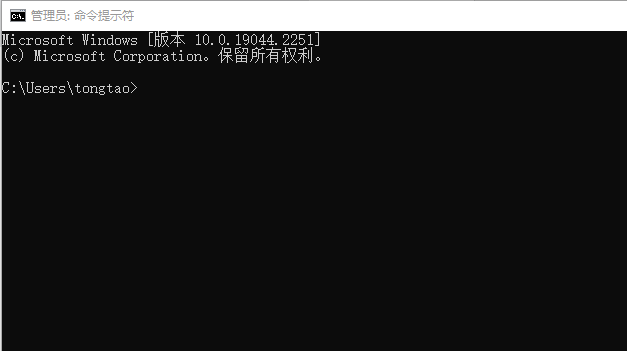
1. 更新代码到最新版本

配置文件说明：在代码目录\LabDatabase\frontend\src\environments 下的

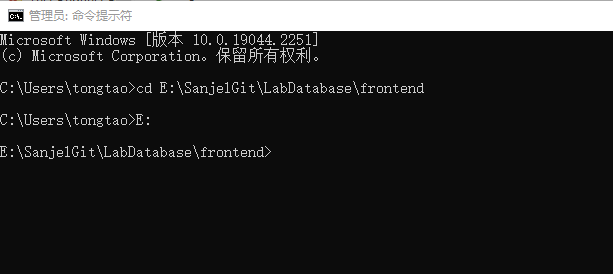
environment.prod.ts 文件是生产环境的打包配置文件，使用ng build –prod命令打包。

environment.ts 文件是全部打包的后台配置文件，使用ng build命令打包。

1. 以管理员身份打开命令窗口



2. 输入前台代码路径，进入其文件夹中



3. 输入打包指令

ng build –prod、 ng build2个打包命令：

* ng build --prod ：用于生产服务器部署，打包文件较小，运行速度快，默认后台服务器配置地址为：

<http://sanjel04.snj.local/sld-api/api/>

* ng build-打包文件大，但是没有压缩，默认后台服务配置地址为：

<http://localhost:5000/api/values、>

一般用于本地打包部署

* 打包测试环境发布

ng build --prod --configuration staging --deploy-url /sld.app/ --base-href /sld.app/

* 打包测试环境发布

ng build --prod --deploy-url /sld.app/ --base-href /sld.app/

Web.cofing

<?xml version="1.0" encoding="UTF-8"?>

<configuration>

<system.webServer>

<rewrite>

<rules>

<clear />

<rule name="AngularJS Routes" stopProcessing="true">

<match url=".\*" />

<conditions logicalGrouping="MatchAll">

<add input="{REQUEST\_FILENAME}" matchType="IsFile" negate="true" />

<add input="{REQUEST\_FILENAME}" matchType="IsDirectory" negate="true" />

</conditions>

<action type="Rewrite" url="/sld.app/" />

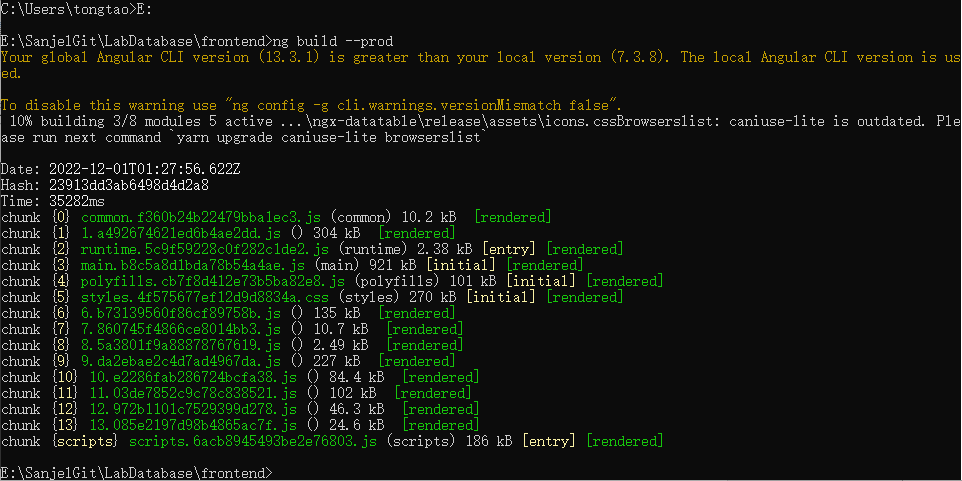
</rule>

</rules>

</rewrite>

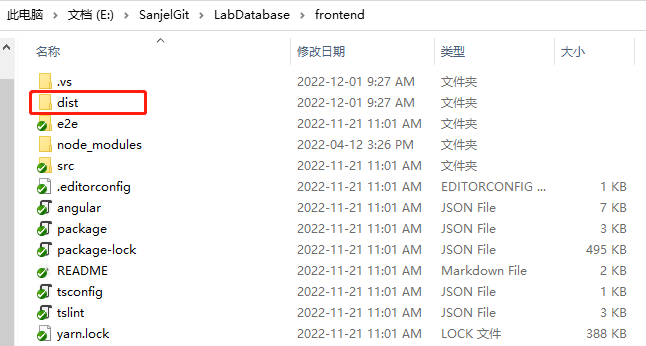
</system.webServer>

</configuration>



4. 打包完成，进入前台项目文件夹，进入dist文件夹，

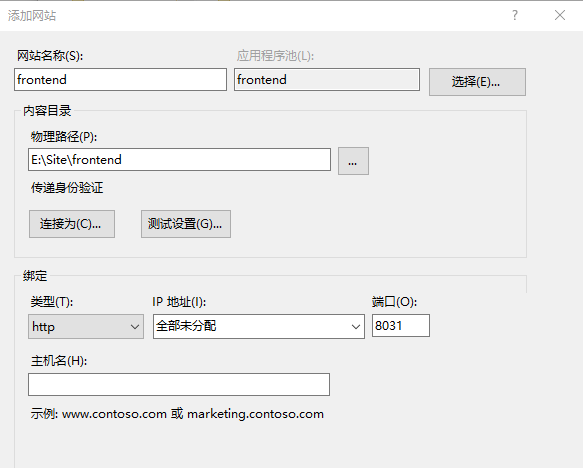
使用该文件夹中的文件包进行发布



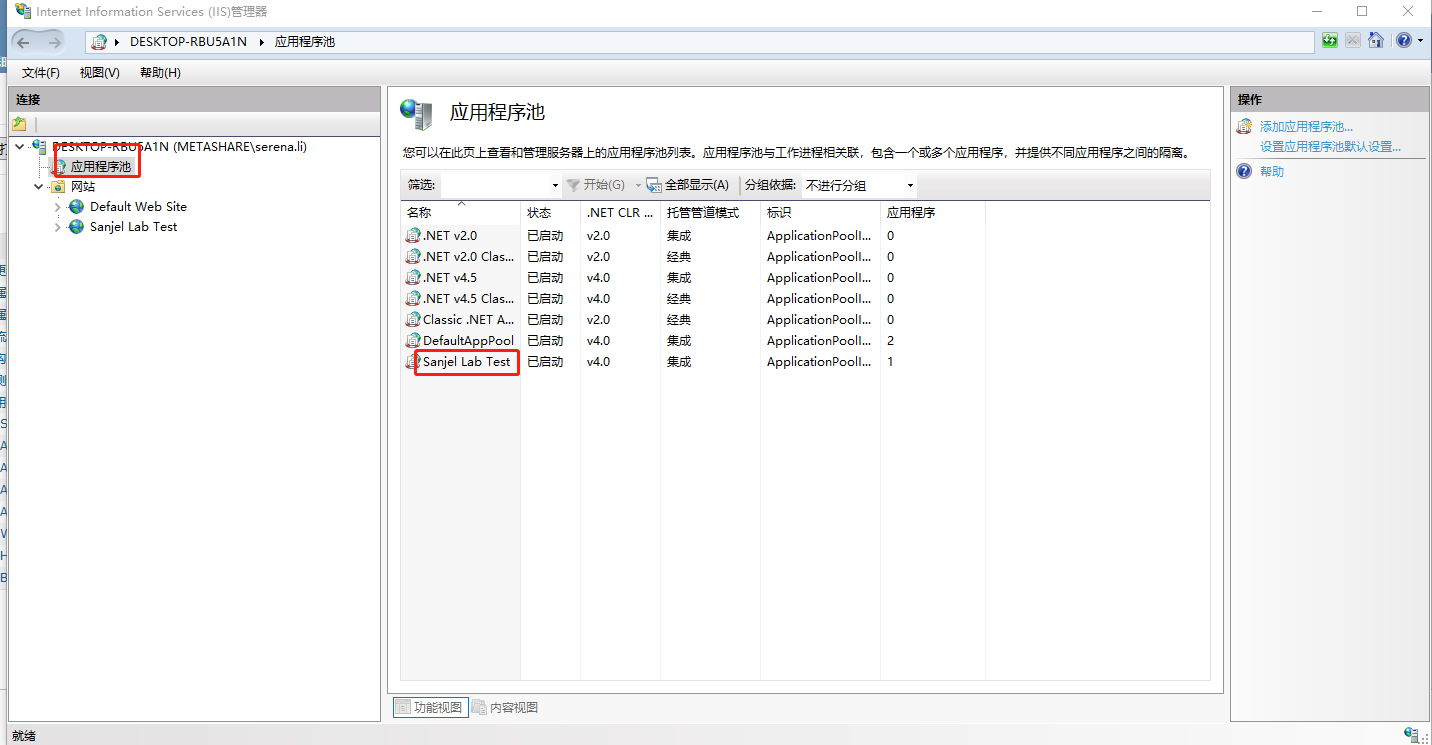


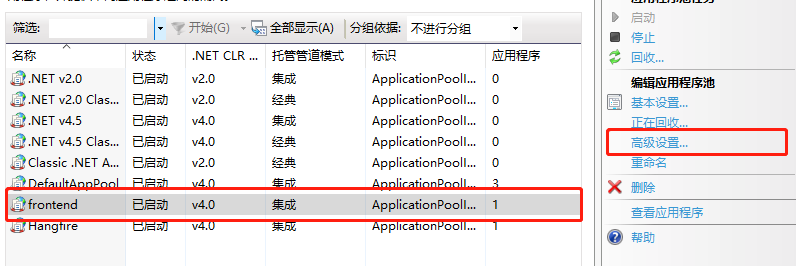
5.将发布文件文件包拷到发布文件夹下

6. 在IIS上发布该项目

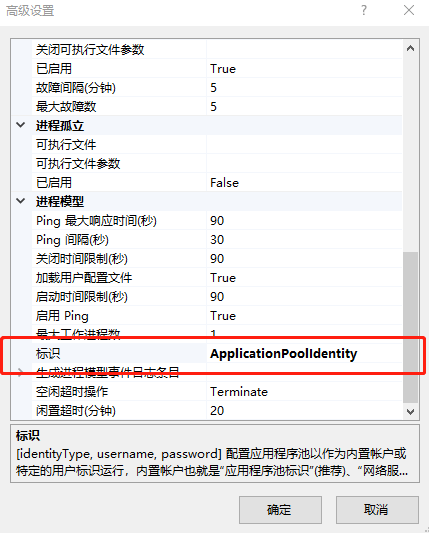


7. 对该项目的应用程序池进行高级设置

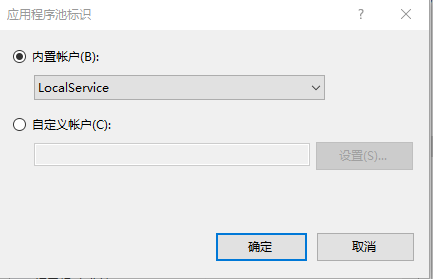




修改“高级设置”中的“进程模型”的标识



将“内置账户”修改为“LocalService”



发布配置完成

## 后台代码打包

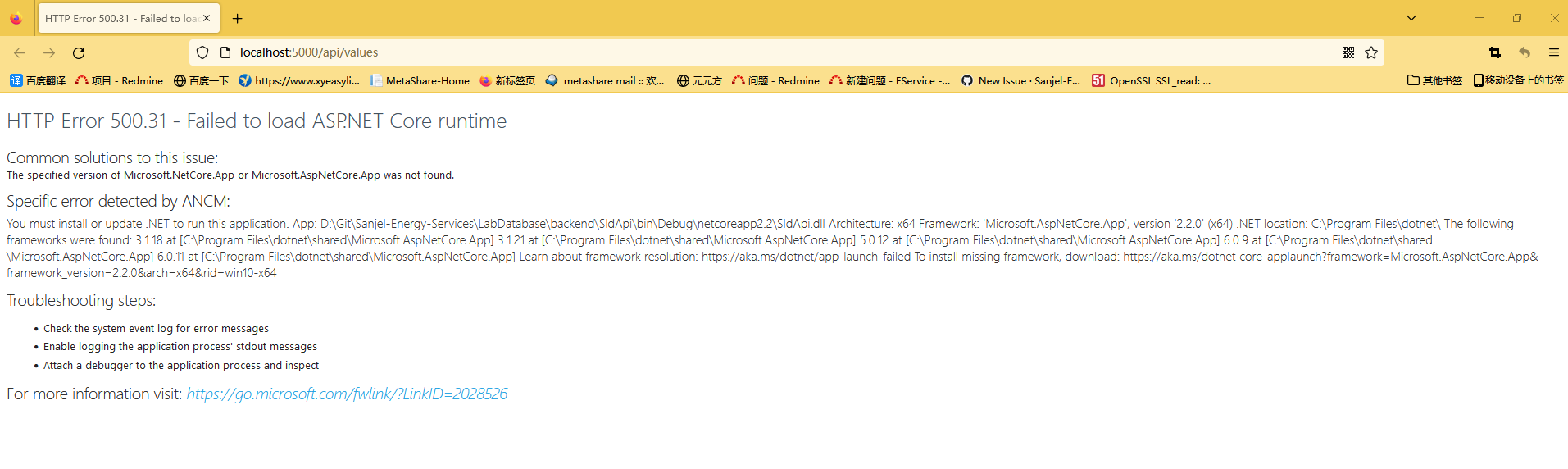
# 常见问题

## 后端运行

### HTTP500.31 Failed to load ASP.NET Core Runtime

原因.net core版本不正确，按照2章节安装.net core 2.2 sdk

VS2019 更改，



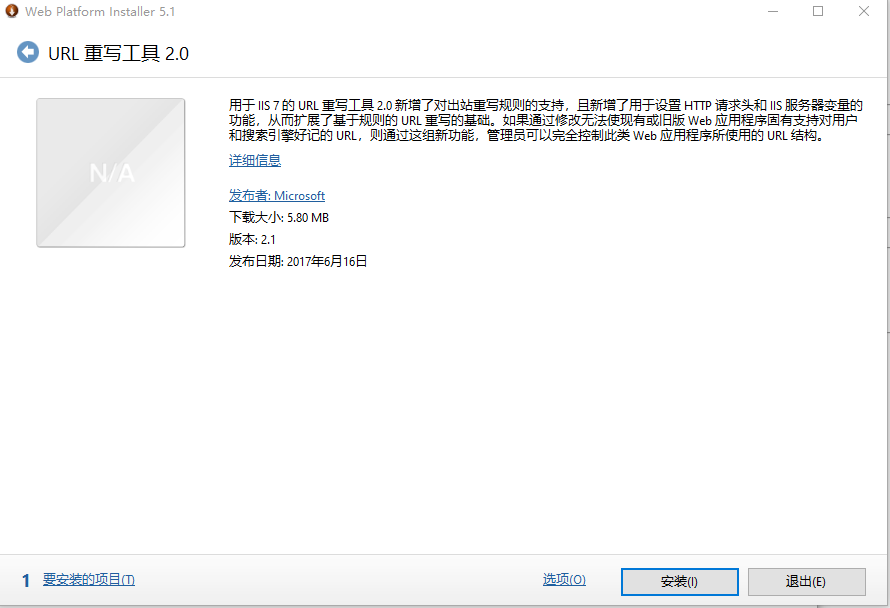
## 前端部署运行

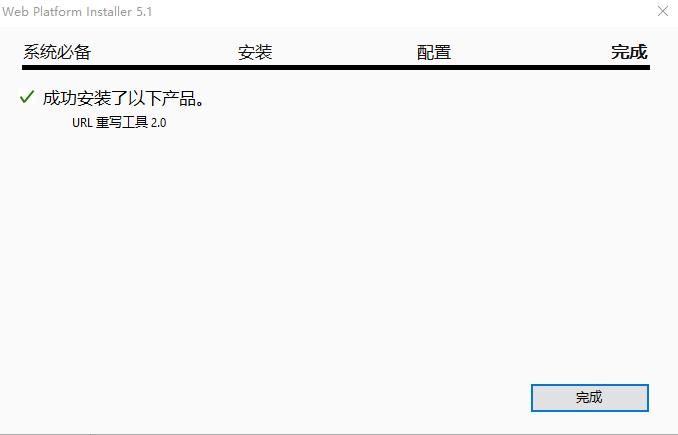
### HTTP 错误 500.19 - Internal Server Error

前台部署运行报错：

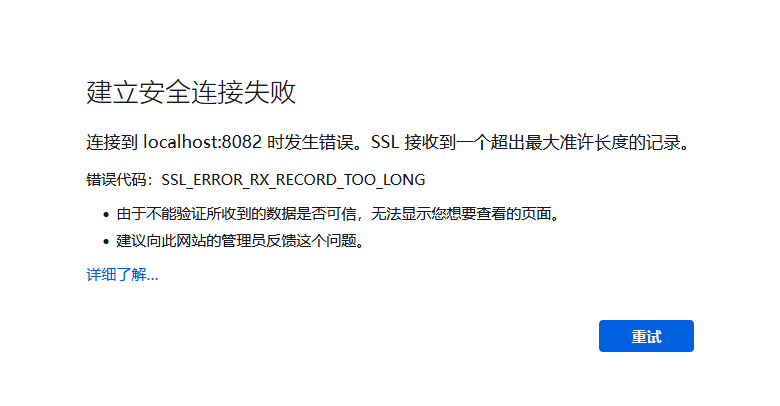


原因：需要安装URL重写工具，安装包目录：



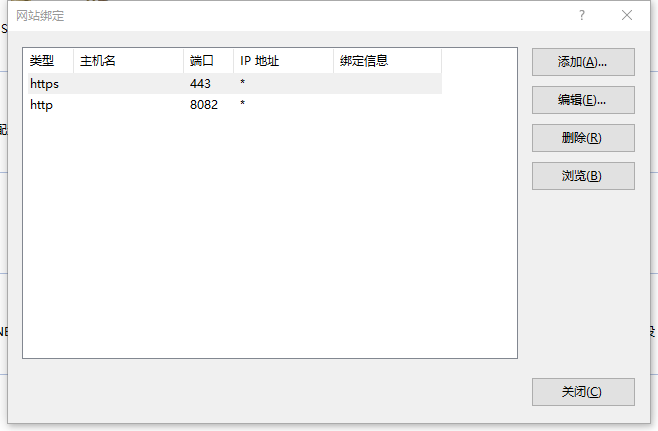


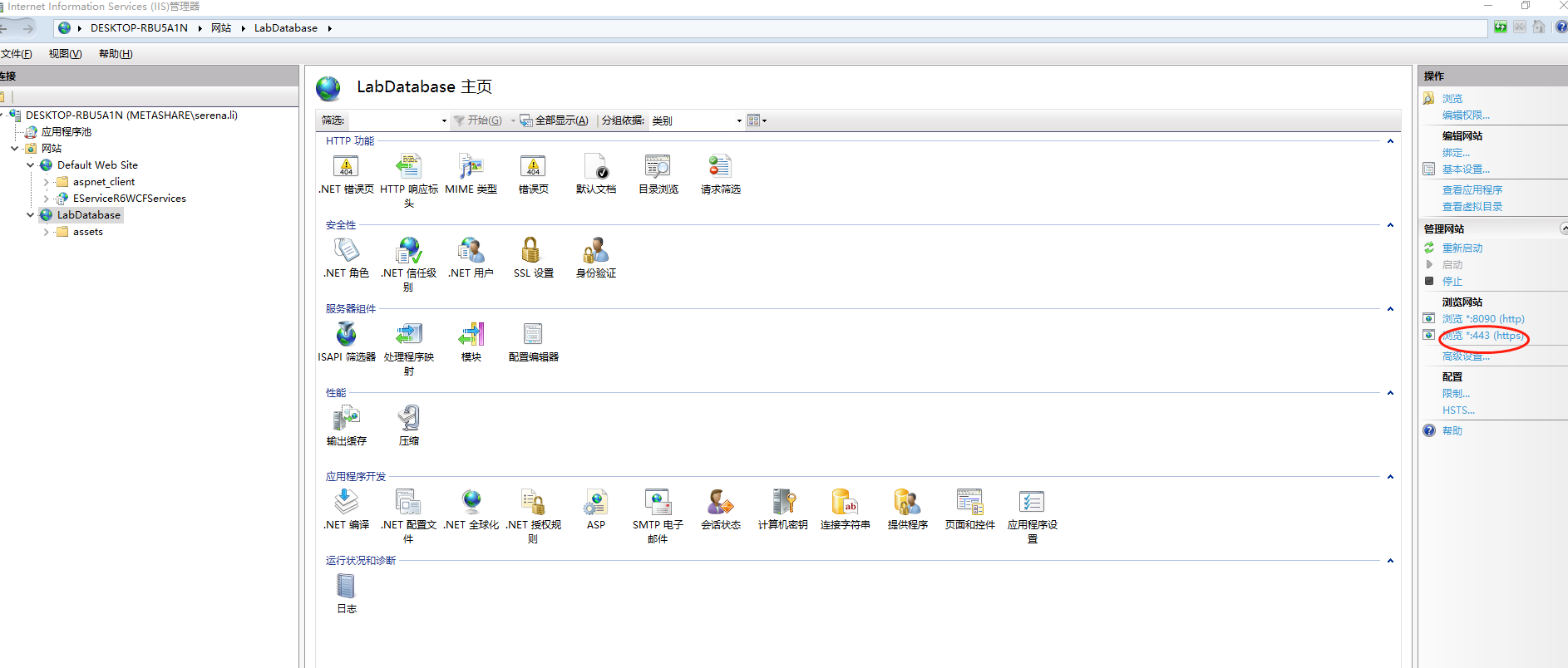
### SSL\_ERROR\_EX\_RECORD\_TOO\_LONG



官方解释

https://learn.microsoft.com/zh-CN/iis/manage/configuring-security/how-to-set-up-ssl-on-iis





此种情况下需要访问443端口的连接，确认继续访问：

