

西安煤机	文档编号	XAMJ_AZSQ_08001	版本	V1.0	密级	商密 A
	项目名称	煤矿掘进工作面网络监控与分析决策软件平台				
	项目来源	自研				

煤矿掘进工作面网络监控与分析决策软件平台

安装部署手册

编写人	赵良梁、张维金、巩瑞华	日期	2022-08-10
检查人	刘帅	日期	2022-08-12
审核人	陈建	日期	2022-08-13
批准人	李世辉	日期	2022-08-14

西安煤机

版权所有 不得复制

版本控制

[illegible]

目 录

1	引言	1
1.1	版权信息	1
1.2	编写目的	1
1.3	适用读者	1
1.4	术语定义	1
1.5	配置清单	2
2	系统构成	2
3	部署环境	3
3.1	硬件环境	3
3.2	软件环境	4
3.3	网络拓补	4
4	环境部署	5
4.1	JAVA 环境	5
4.1.1.	环境准备	5
4.1.2.	安装 JDK.....	5
4.1.3.	环境变量配置	8
4.1.4.	验证是否安装成功	15
4.2	三维动画运行环境安装	17
4.2.1.	软件安装	17
4.2.2.	配置中文显示	18
4.2.3.	取消进度显示	20
4.2.4.	配置鼠标显示	20
4.3	NGINX 安装	22
4.3.1.	安装包下载	22
4.3.2.	解压安装包	22
4.3.3.	启动 NGINX	23
4.3.4.	验证安装结果	23
4.3.5.	停止服务	23
5	系统部署	24
5.1	集控服务程序	24
5.1.1.	修改配置	24
5.1.2.	启动程序	26
5.1.3.	退出程序	26
5.2	接口服务启动	26
5.2.1.	打开压缩包	26
5.2.2.	编辑 YML 文件	27
5.2.3.	启动程序	27

5.2.4.	启动验证	28
5.3	三维动画	29
5.3.1.	修改端口地址	29
5.3.2.	打开软件	29
5.3.3.	界面操作	29
5.3.4.	启动关闭操控	31
5.3.5.	快捷键大全	31
5.3.6.	注意事项	31
5.4	WEB 部署	32
5.4.1.	解压程序包	32
5.4.2.	拷贝程序文件	32
5.4.3.	添加 conf/nginx.config 配置	33
5.4.4.	验证	33
6	系统验证	33
6.1	配置参数发送验证	33
6.2	控制命令发送验证	34
6.3	位姿测量数据接收验证	34
6.4	掘进机工况数据接收验证	34
6.5	自移机尾工况数据接收验证	34

1 引言

1.1 版权信息

本文档版权信息属于西安煤机，未经本单位书面许可，严禁任何形势的传播。

1.2 编写目的

本文面向实施工程师以及业主方系统管理人员，是实施工程师或业主方系统管理员搭建运行环境、安装部署、系统集成的主要依据和指导性文件。

1.3 适用读者

本文档适用于实施工程师以及业主方系统管理人员。

1.4 术语定义

术语	解释
WebSocket	WebSocket 协议是基于 TCP 的一种新的网络协议。它实现了浏览器与服务器全双工 (full-duplex) 通信 -- 允许服务器主动发送信息给客户端。 WebSocket通信协议于2011年被IETF定为标准 RFC 6455,并被RFC7936所补充规范。
Tcp/IP	TCP/IP 是基于 TCP 和 IP 这两个最初的协议之上的不同的通信协议的大集合,用于从应用程序到网络的数据传输控制。

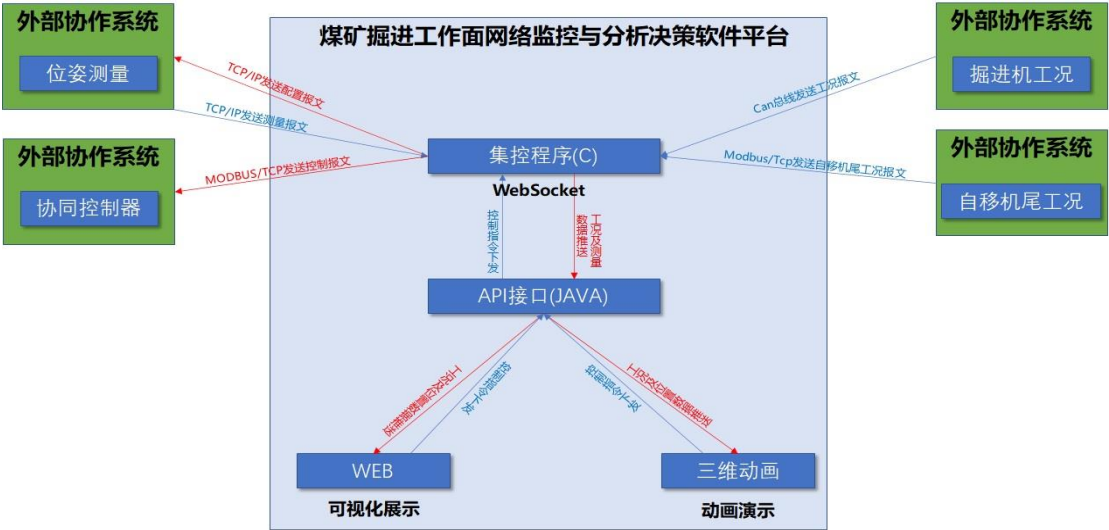
Modbus	Modbus 是一种串行通信协议，是 Modicon 公司(现在的施耐德电气 Schneider Electric)于 1979 年为使用可编程逻辑控制器(PLC)通信而发表。Modbus 已经成为工业领域通信协议的业界标准(De facto)，并且现在是工业电子设备之间常用的连接方式。
Can 总线	CAN 是控制器局域网(Controller Area Network, CAN)的简称，是由以研发和生产汽车电子产品著称的德国 BOSCH 公司开发的，并最终成为国际标准(ISO 11898)，是国际上应用最广泛的现场总线之一。

1.5 配置清单

清单名称	文件形式	文件名称
部署手册	WORD 文档	安装部署手册
接口程序包	Jar 包	tunneling-web-api.jar
三维程序安装包	Rar 压缩包	Ventuz.rar
Web 程序包	Rar 压缩包	DrivingFaceWisdom.rar
三维动画程序	Vpr 文件	掘进系统 0812S.vpr
集控服务程序	Rar 压缩包	IndControl.rar

2 系统构成

本系统分为集控程序（C 语言）、API 接口程序（JAVA）、WEB 可视化展示（VUE.JS+NODE.js）以及三维动画几个部份构成，系统各部份及相关协作系统工作流程如下：



3 部署环境

3.1 硬件环境

服务器：服务器包含数据库服务器、集控程序服务器、接口服务器、WEB 应用服务器，可以根据实际情况，配置一台或多台服务器，最低配置要求如下：

配置	硬件规格要求
处理器	Pentium(R) Dual-Core CPU E5300 @ 2.60GHz 或以上
内存	16GB 或以上
硬盘	1TB 以上 SSD 固态硬盘
网卡	1000 兆网卡

客户端：因为需要运维三维动画，因此客户对显示卡的要求比较特殊，必须是独立显示卡，最低配置要求如下：

配置	硬件规格要求
显示卡	GeForce GTX1050 Ti

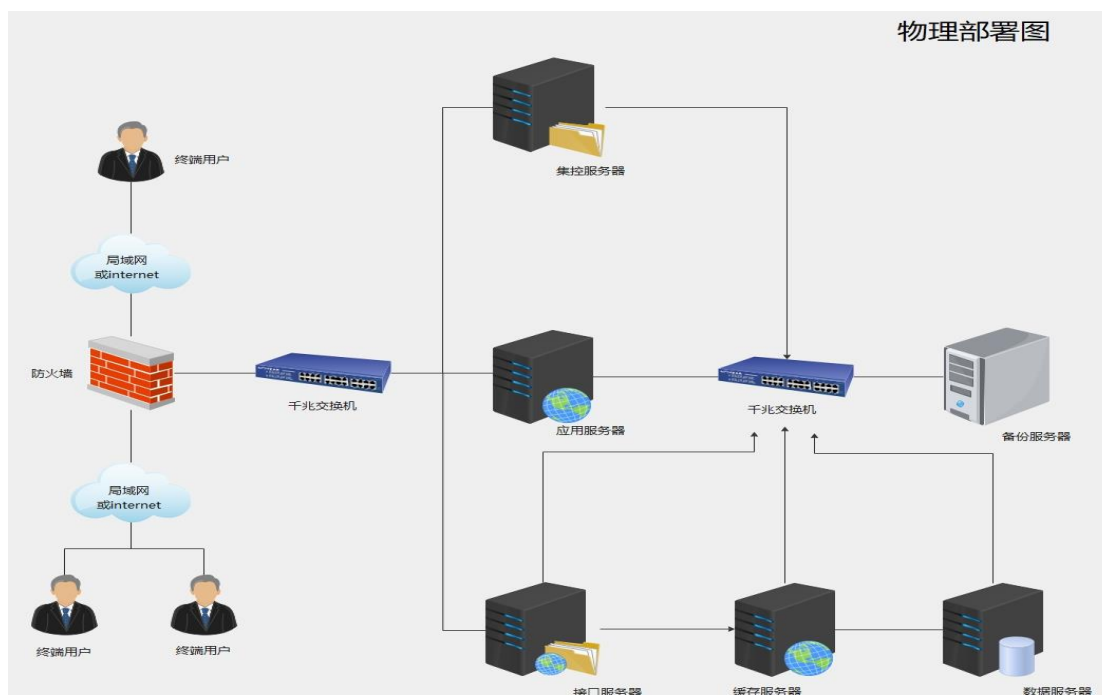
内存	8GB 或以上
显示器	1920*1080
浏览器	Chrome53 以上
硬盘	80GB 以上

3.2 软件环境

服务器：服务器建议使用 Windows Server 2012 及以上操作系统，也可以采用 Linux64 位操作系统，或用 Win10 64 位以上系统代替。此外系统需要数据库支持，数据库平台采用 PostgreSQL 11.2 (Debian 11.2-1.pgdg90+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 6.3.0-18+deb9u1) 6.3.0 20170516, 64-bit。

客户端：Win10 以上操作系统。

3.3 网络拓补



4 环境部署

4.1 JAVA 环境

4.1.1. 环境准备

1) Windows 系统或者 Linux 系统。

2) JDK1.8 安装包。

3) Windows 安装包地址:

<https://repo.huaweicloud.com/java/jdk/8u202-b08/jdk-8u202-windows-x64.exe>

Linux 安装包地址:

<https://repo.huaweicloud.com/java/jdk/8u202-b08/jdk-8u202-linux-x64.tar.gz>

4) tunneling-web-api.jar 接口程序包，随部署手册提供。

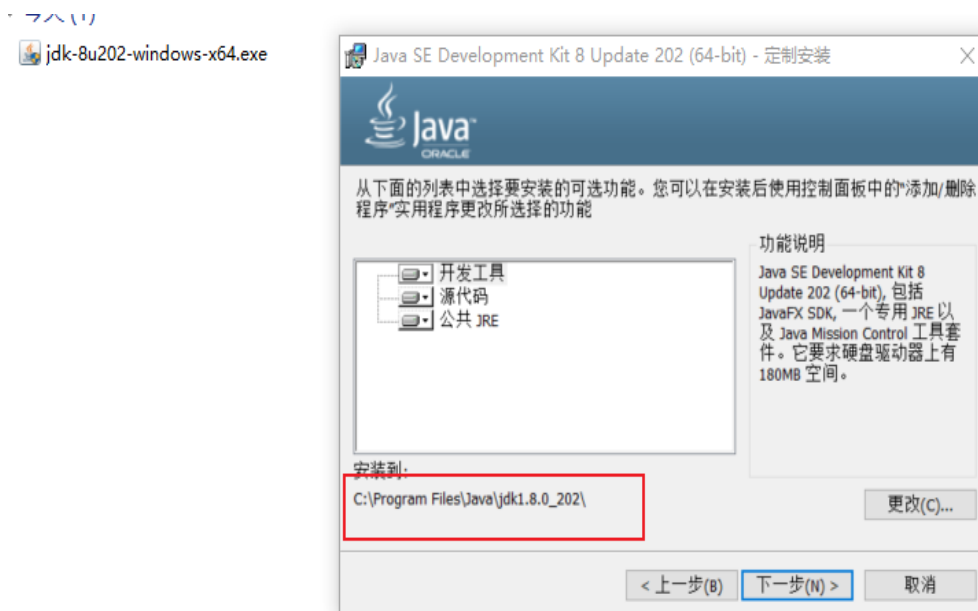
4.1.2. 安装 JDK

下面以 Window 环境为例，讲解 JDK 的安装环境，本手册中截图是在 Win10 操作系统环境下载的图片，可能会因操作系统版本不同而有所变化。

双击打开下载好的 jdk1.8 安装包（参考 4.1.1 中下载地址），然后点击【下一步】按钮。



选择 JDK 安装路径（建议复制出来, 后面要用到），然后点击【下一步】按钮继续。



安装完成 jdk 后会自动弹出安装 jre，这里就直接点击【关闭】，不需要单独安装 jre。

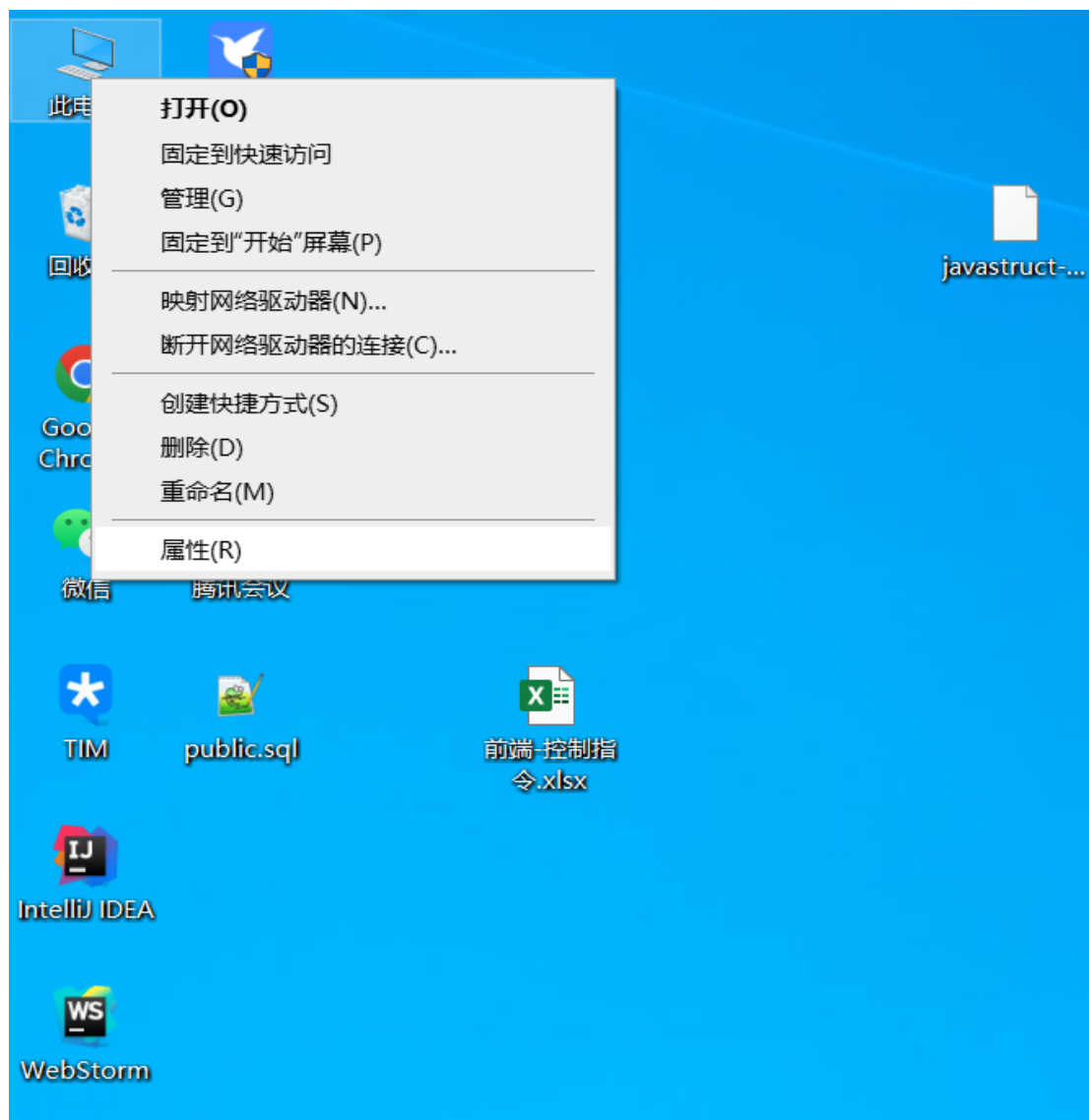


安装完毕后直接点击关闭即可。



4.1.3. 环境变量配置

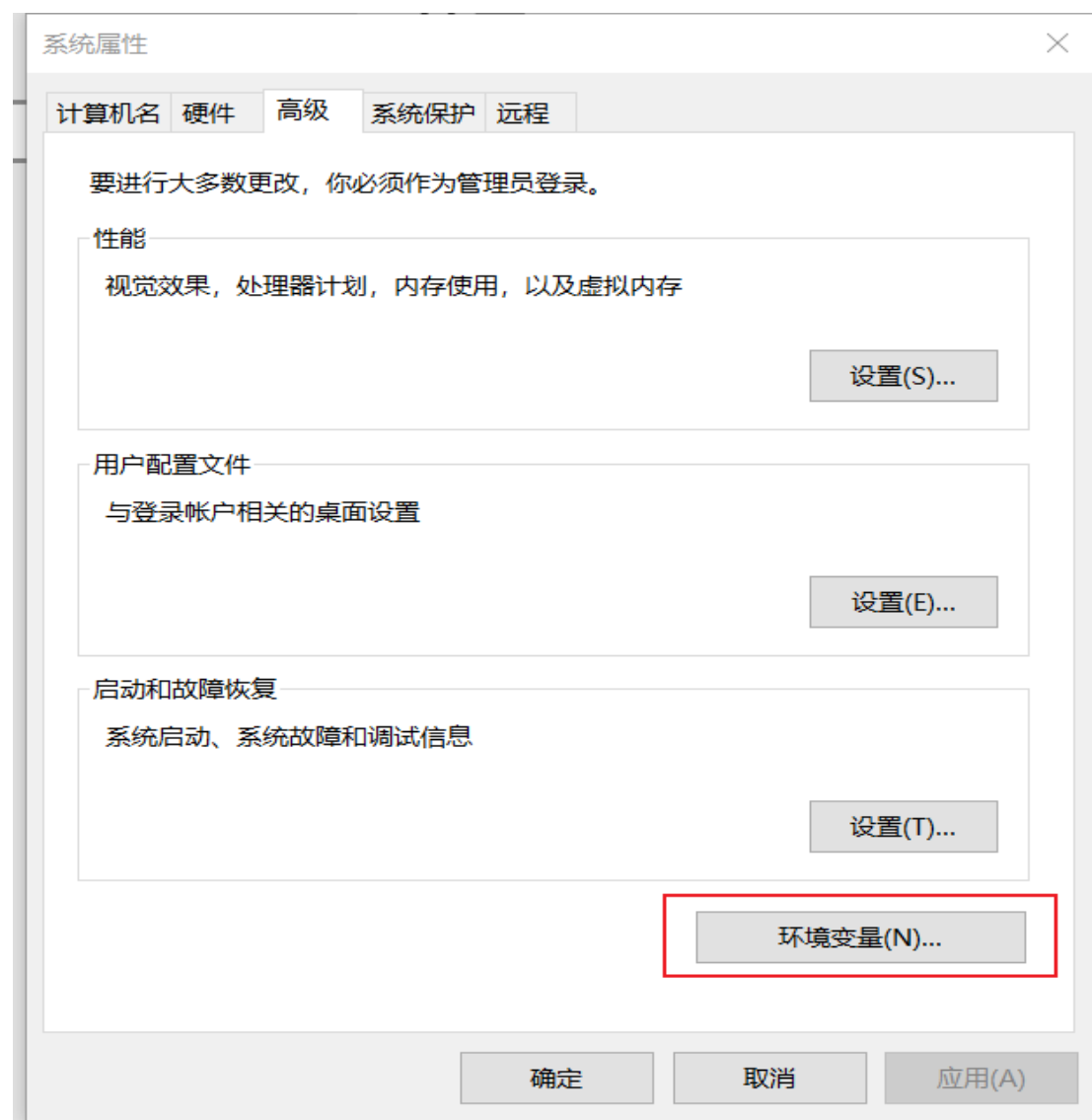
鼠标右击此电脑，然后点击属性。



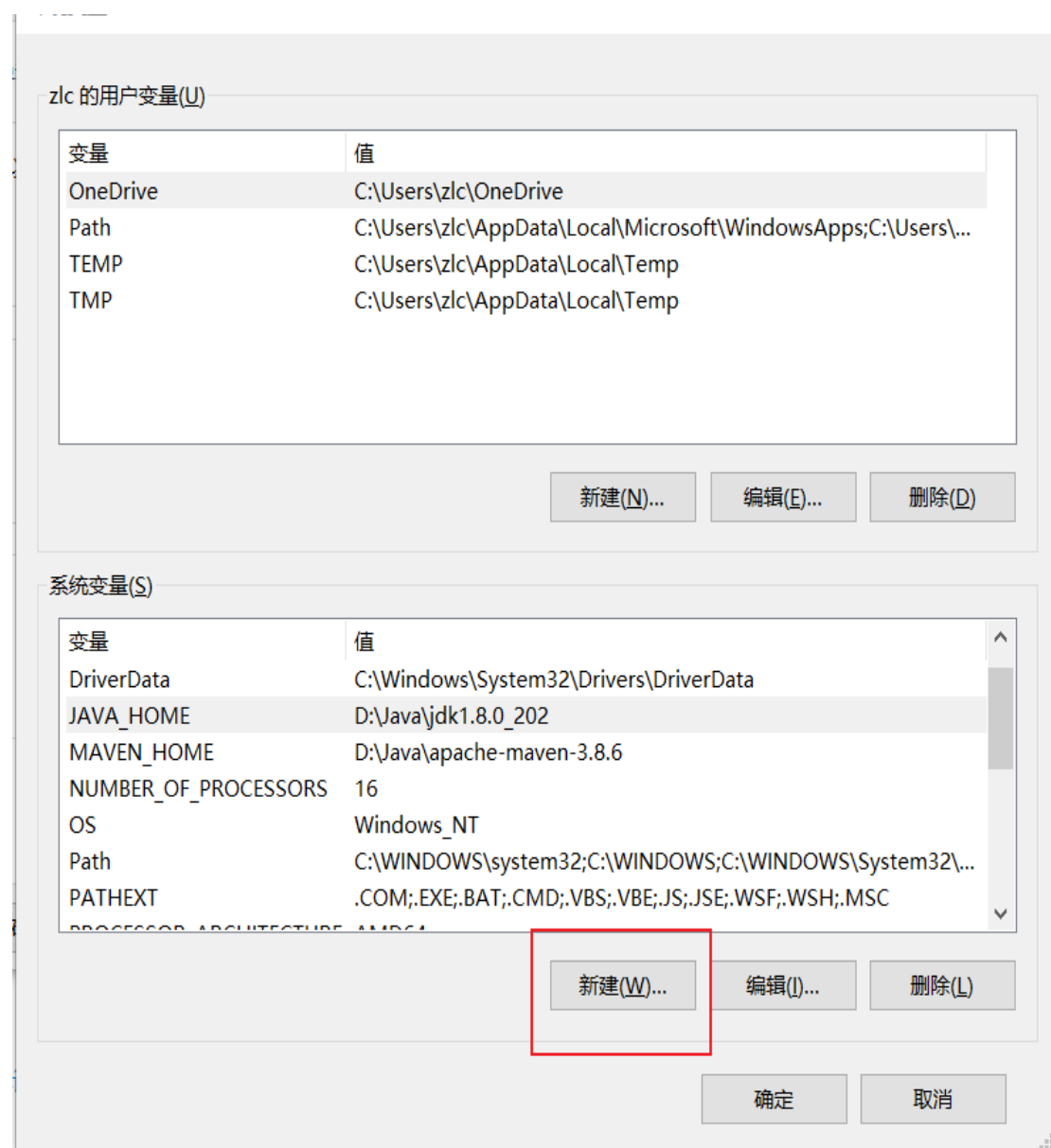
在弹出的对话框中，找到并点击高级系统设置选项。



在弹出来的窗口中找到并点击【环境变量】按钮，开始配置坏境变量。



在弹出来窗口中，在系统变量中点击【新建】按钮。

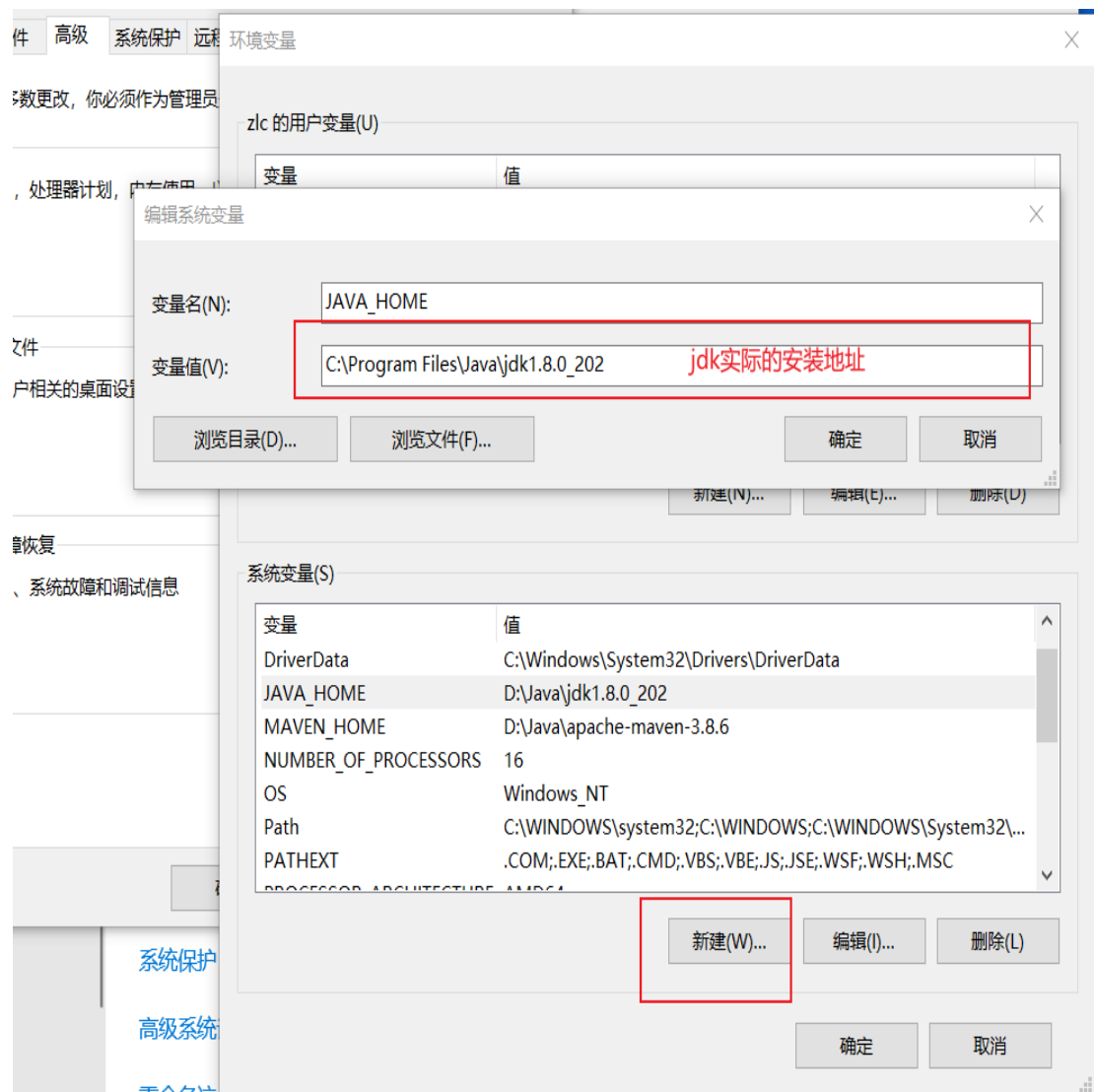


在弹出来的窗口中输入变量名和变量值。

变量名：JAVA_HOME （名字可以另外取）。

变量值：刚才 JDK1.8 安装的目录（复制出来的好处）。

然后点击【确定】按钮。

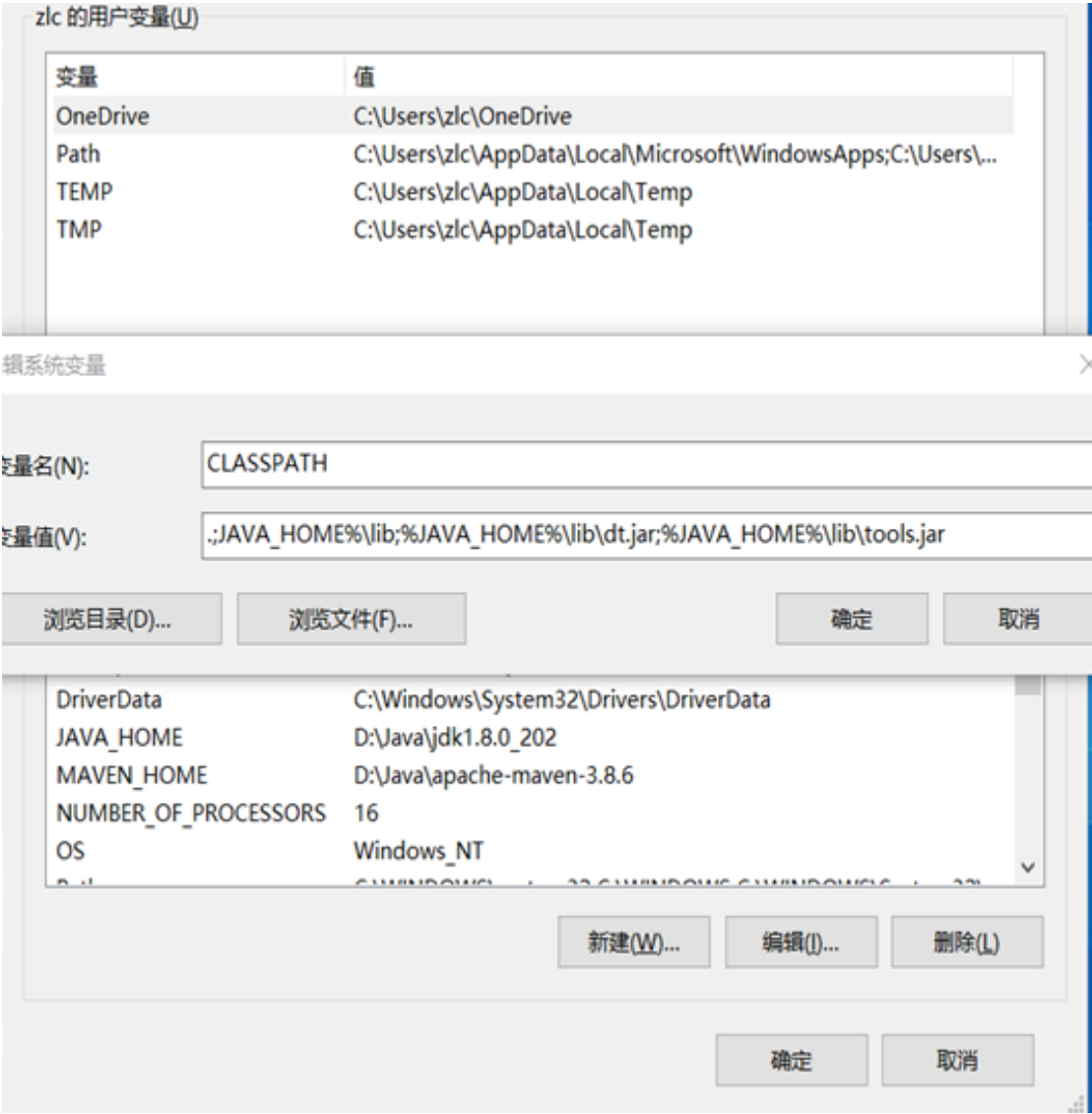


继续添加变量

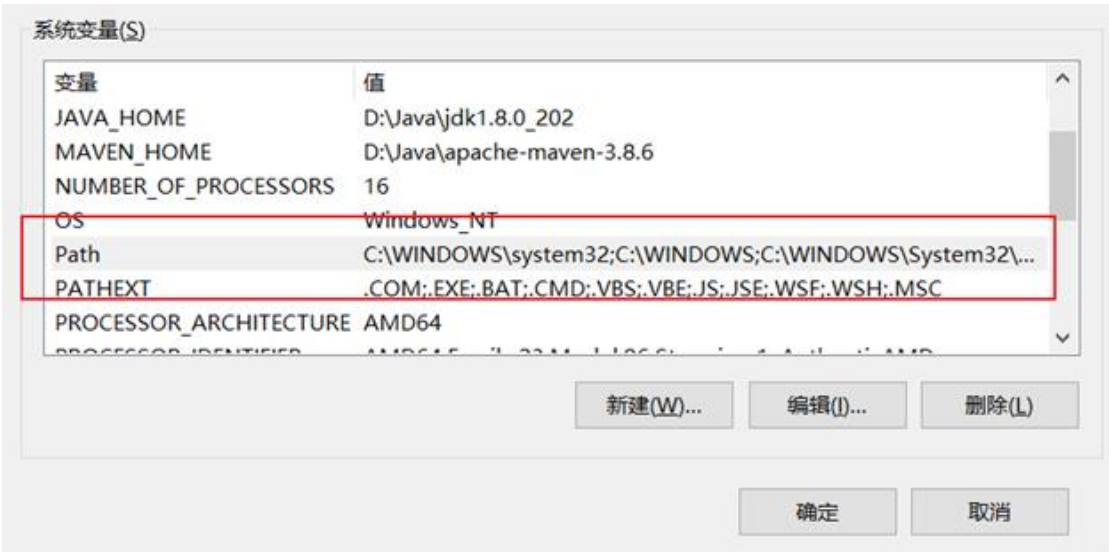
变量名：CLASSPATH

变量值：

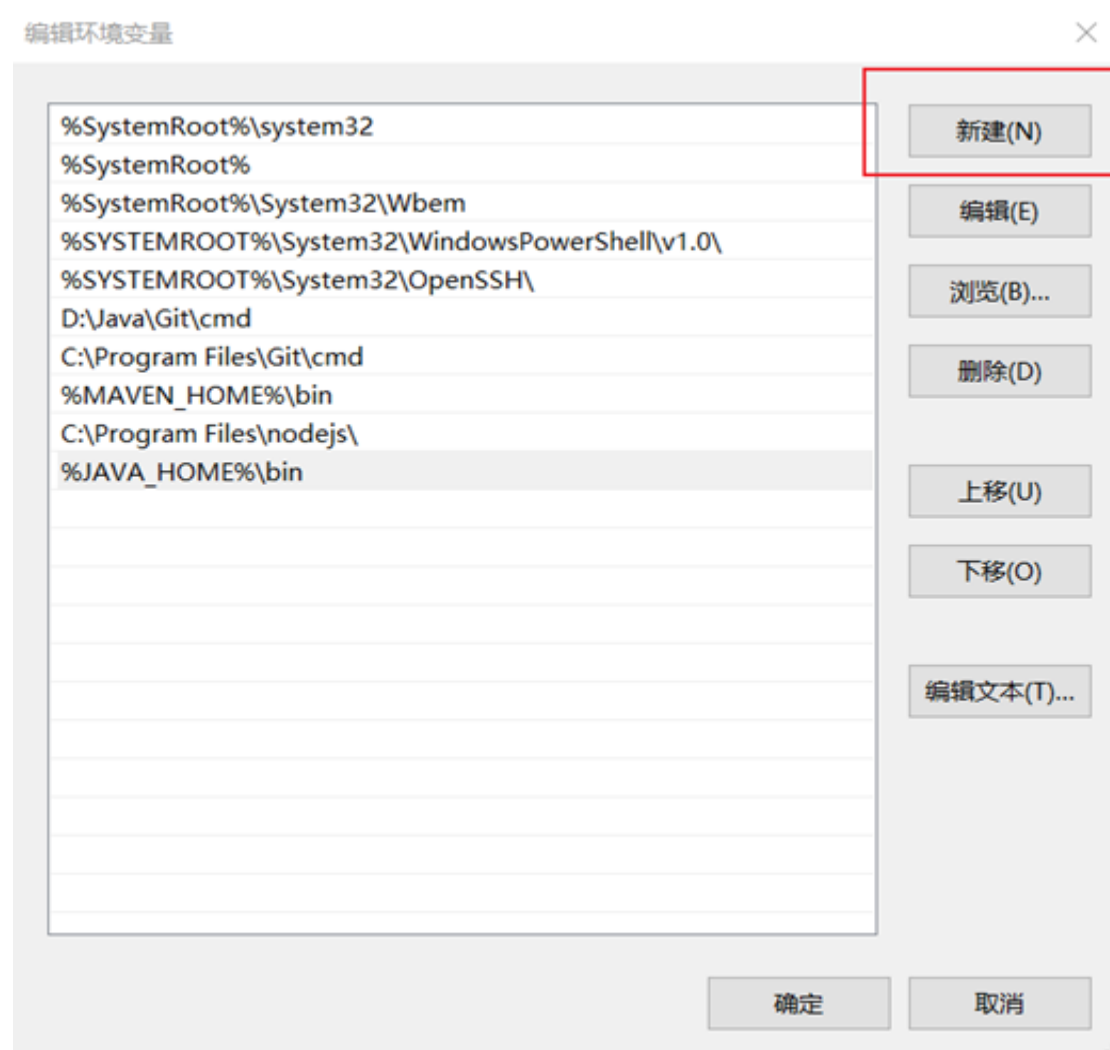
.;JAVA_HOME%\lib;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar （注意不要输错）。



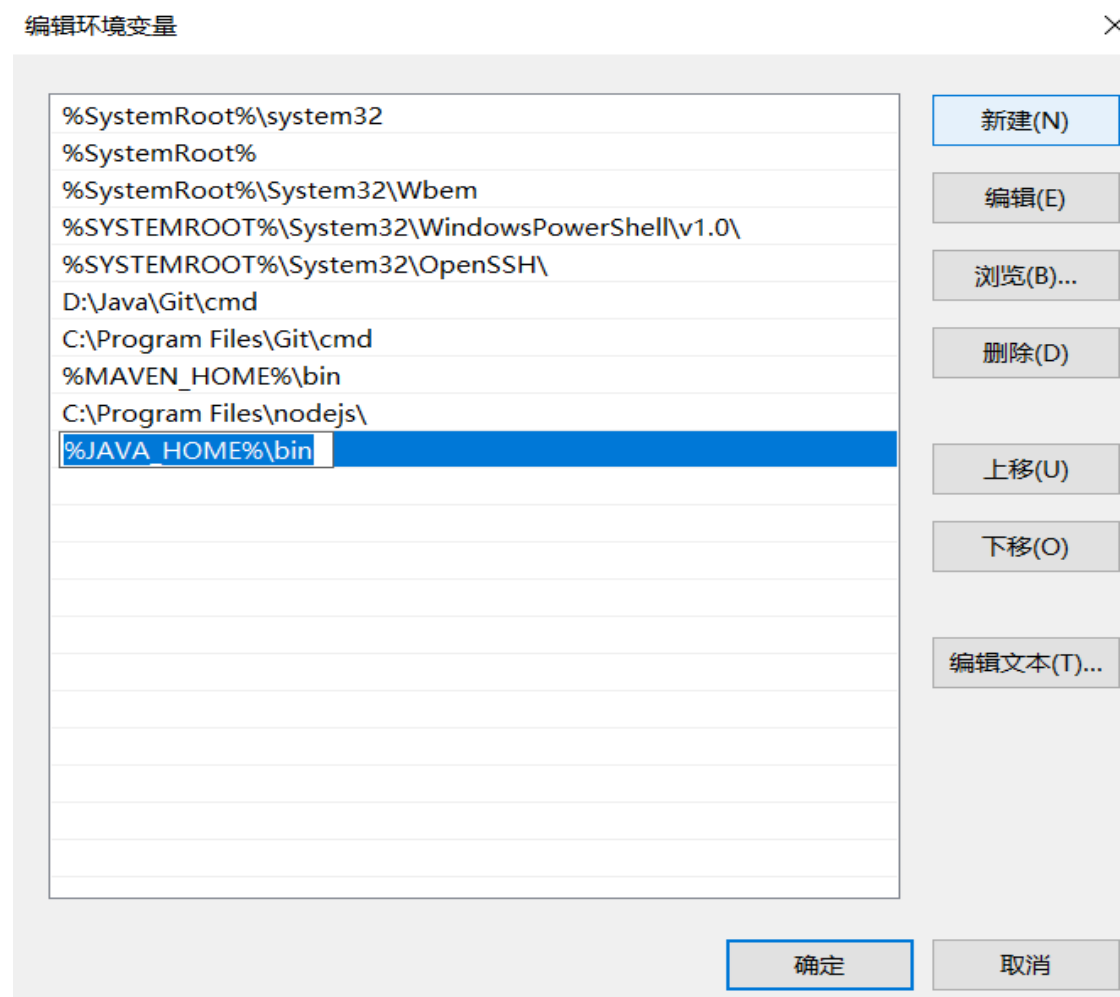
在系统变量列表中，找到 Path，点击【编辑】按钮。



在弹出来窗口中，点击【新建】按钮。



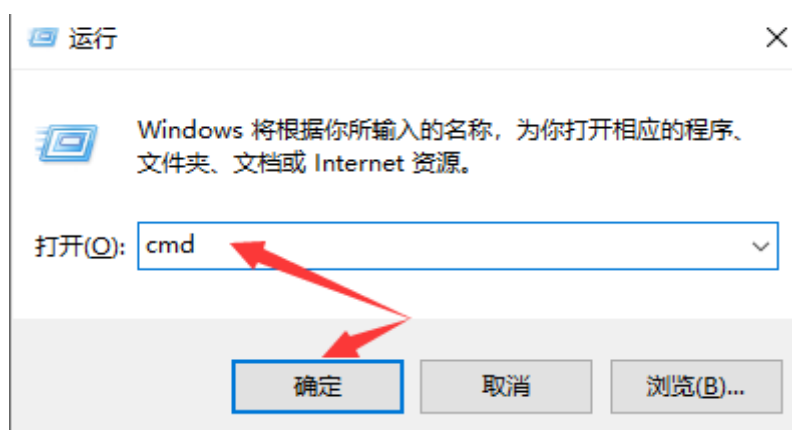
然后输入： %JAVA_HOME%\bin， 点击【确定】按钮。



至此 JAVA 环境已经安装配置完成。

4.1.4. 验证是否安装成功

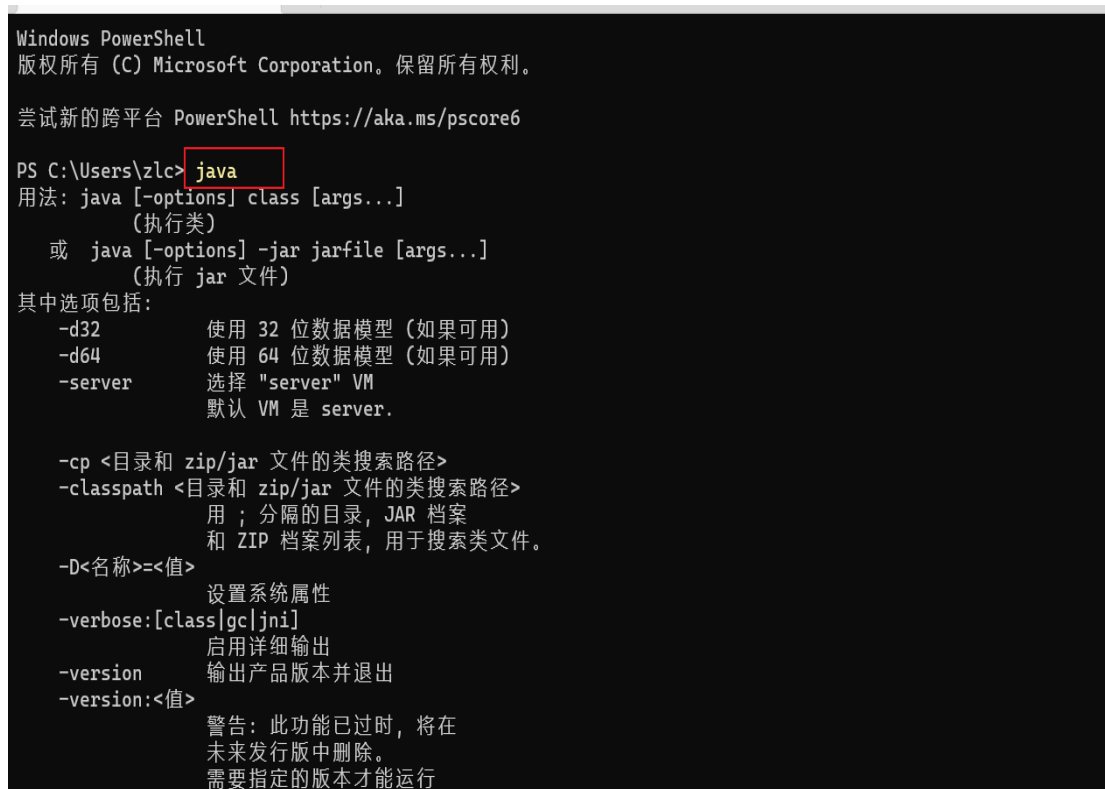
同时按下键盘上的 WINDOW 标志键 和 R 键，呼出运行窗口，输入 cmd，然后点击【确定】按钮。



系统会进入命令控制台。



在命令窗口中输入: `java`, 然后敲击回车键, 如果屏幕出下以下内容, 说明 JAVA 环境安装成功。



此外, 也可以输入: `javac`, 然后敲击回车键, 如果屏幕出下以下内容, 说明 JAVA 环境安装成功。

```

PS C:\Users\zlc> javac
用法: javac <options> <source files>
其中, 可能的选项包括:
-g                生成所有调试信息
-g:none          不生成任何调试信息
-g:{lines,vars,source} 只生成某些调试信息
-nowarn          不生成任何警告
-verbose         输出有关编译器正在执行的操作的消息
-deprecation     输出使用已过时的 API 的源位置
-classpath <路径> 指定查找用户类文件和注释处理程序的位置
-cp <路径>       指定查找用户类文件和注释处理程序的位置
-sourcepath <路径> 指定查找输入源文件的位置
-bootclasspath <路径> 覆盖引导类文件的位置
-extdirs <目录>   覆盖所安装扩展的位置
-endorseddirs <目录> 覆盖签名的标准路径的位置
-processor <class1>[,<class2>,<class3>...] 要运行的注释处理程序的名称; 绕过默认搜索进程
-processorpath <路径> 指定查找注释处理程序的位置
-parameters     生成元数据以用于方法参数的反射
-d <目录>       指定放置生成的类文件的位置
-s <目录>       指定放置生成的源文件的位置
-h <目录>       指定放置生成的本机标头文件的位置
-implicit:{none,class} 指定是否为隐式引用文件生成类文件
-encoding <编码> 指定源文件使用的字符编码

```

还可以输入 `java -version` 指令（注意 `java` 和 `version` 中间有个空格），然后敲击回车键，如果屏幕出下以内容，说明 `JAVA` 环境安装成功。

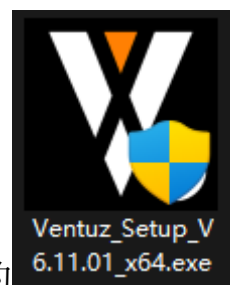
```

PS C:\Users\zlc> java -version
java version "1.8.0_202"
Java(TM) SE Runtime Environment (build 1.8.0_202-b08)
Java HotSpot(TM) 64-Bit Server VM (build 25.202-b08, mixed mode)
PS C:\Users\zlc> |

```

4.2 三维动画运行环境安装

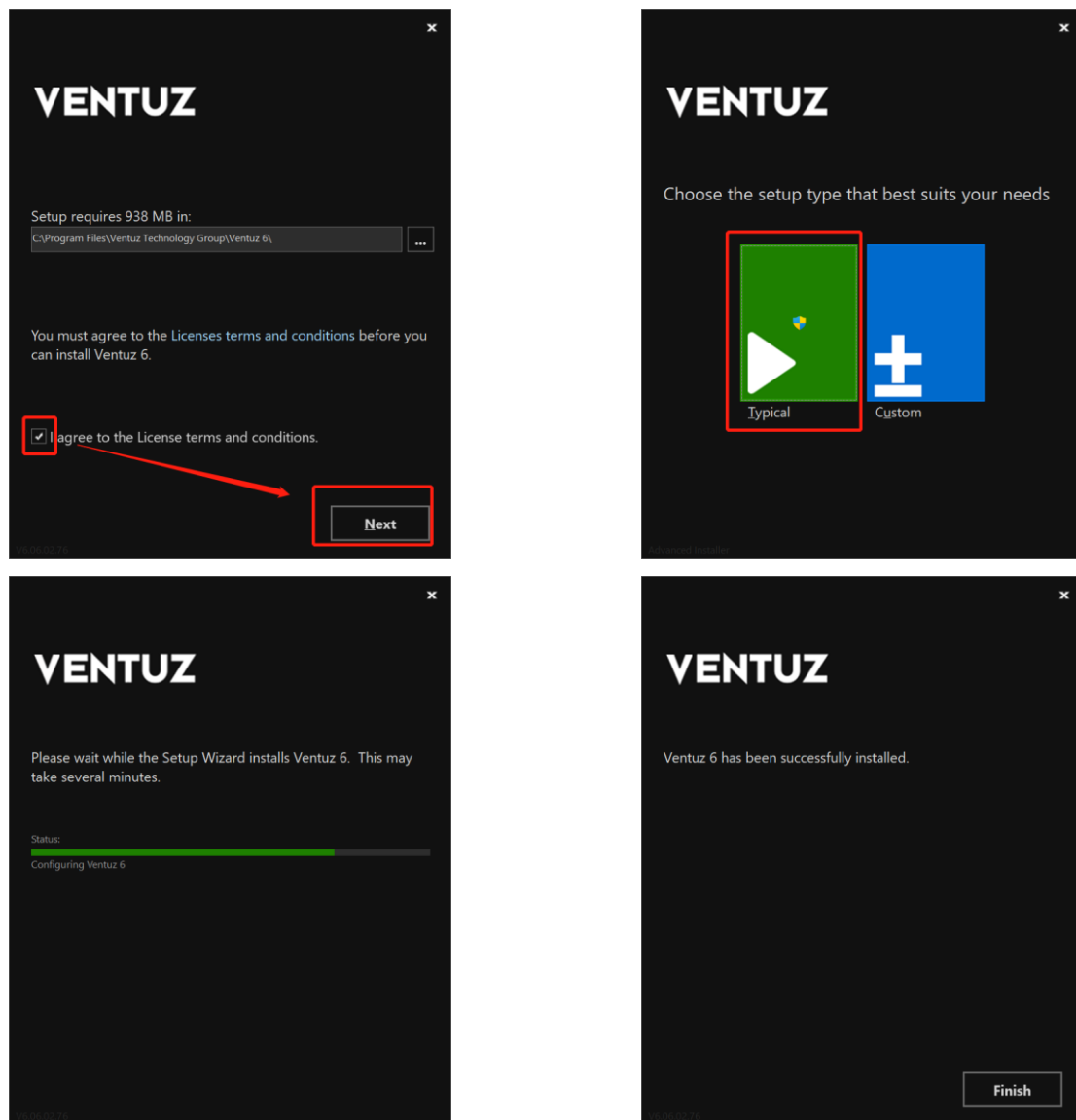
4.2.1. 软件安装



解压 Ventuz 压缩包，找到并双击运行其中的文件，

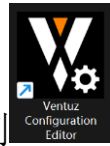
开始安装。

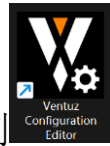
安装过程中会安装一些系统组件，请参照以下图示确保安装完毕。最后点击 **Finish** 按钮即可。

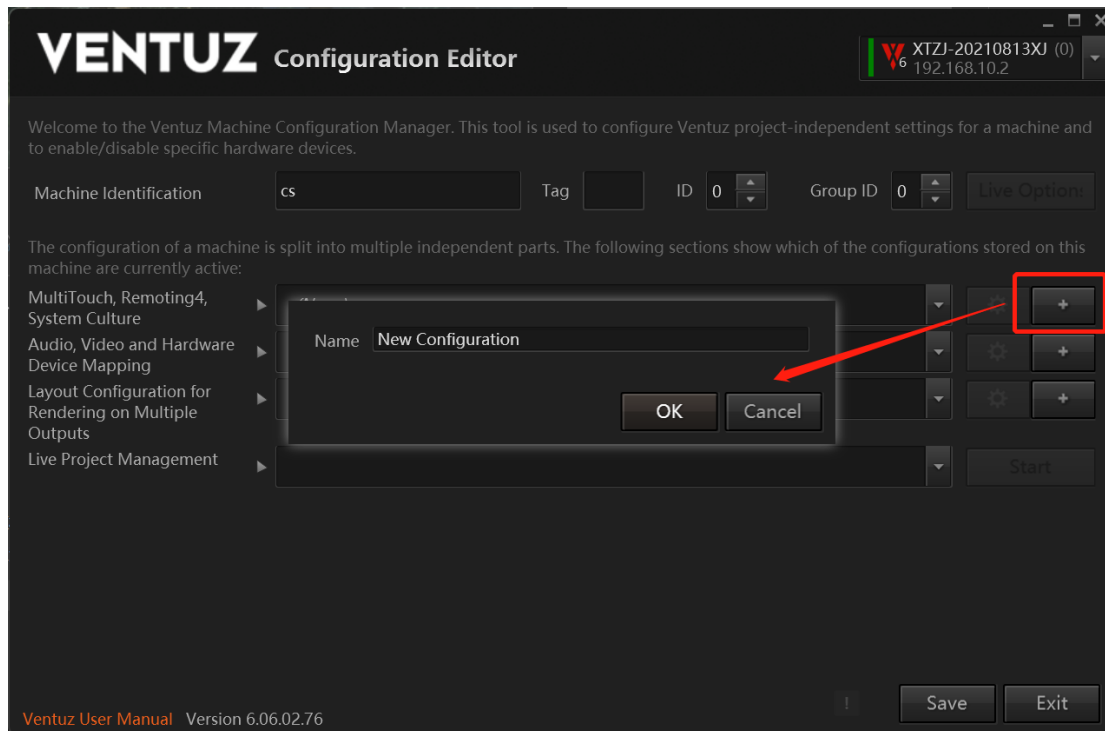


4.2.2. 配置中文显示

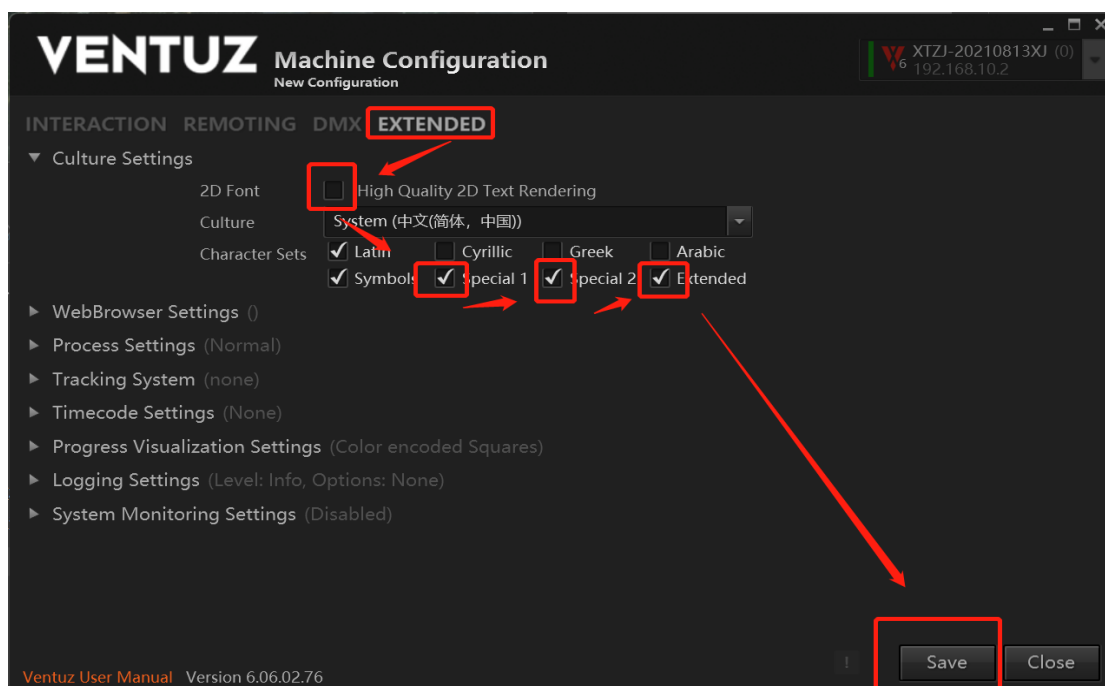
在 软 件 的 安 装 路 径 （ 默 认 路 径 为
C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Ventuz



Technology\Ventuz 6\Tools) 中, 找到 , 双击打开, 即进入配置界面。在红框位置点击加号新建配置, 点击然后 ok。

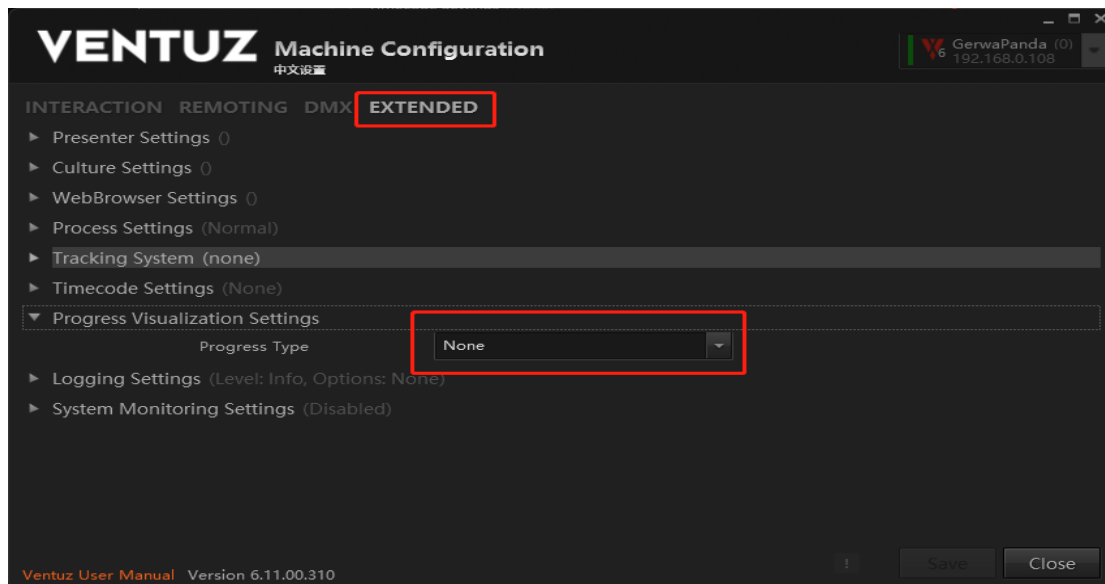


在 EXTENDED——Culture Settings 取消和勾选图片所示, 之后点击 Save。



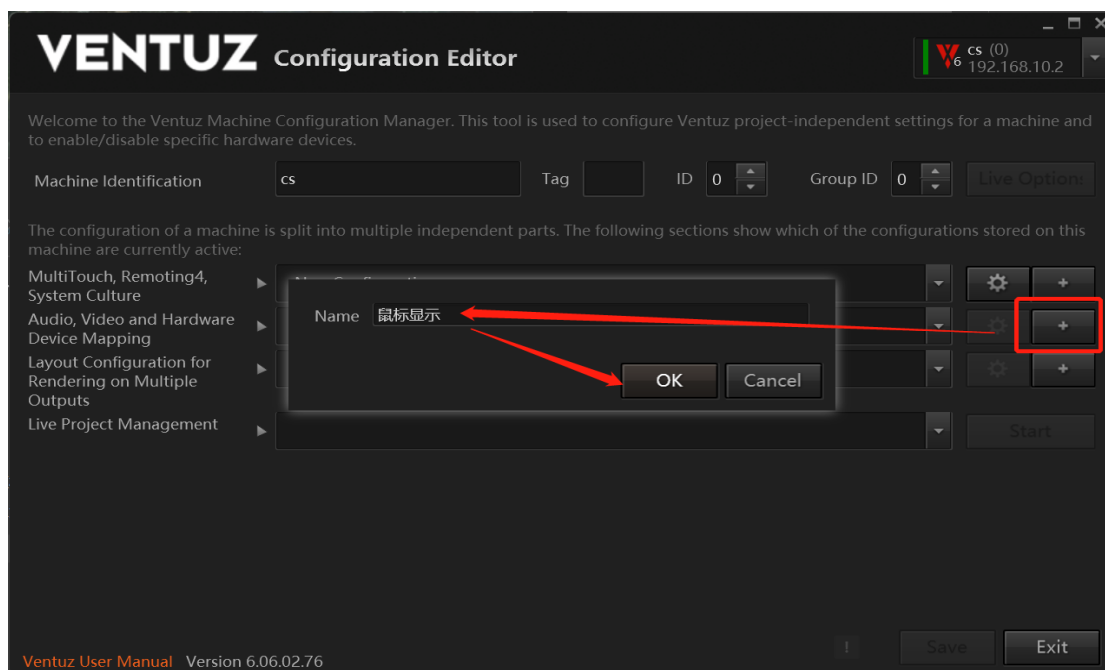
4.2.3. 取消进度显示

在 EXTENDED——Progress Visualization Setting, Progress Type 项改为 None, 之后点击 Save。



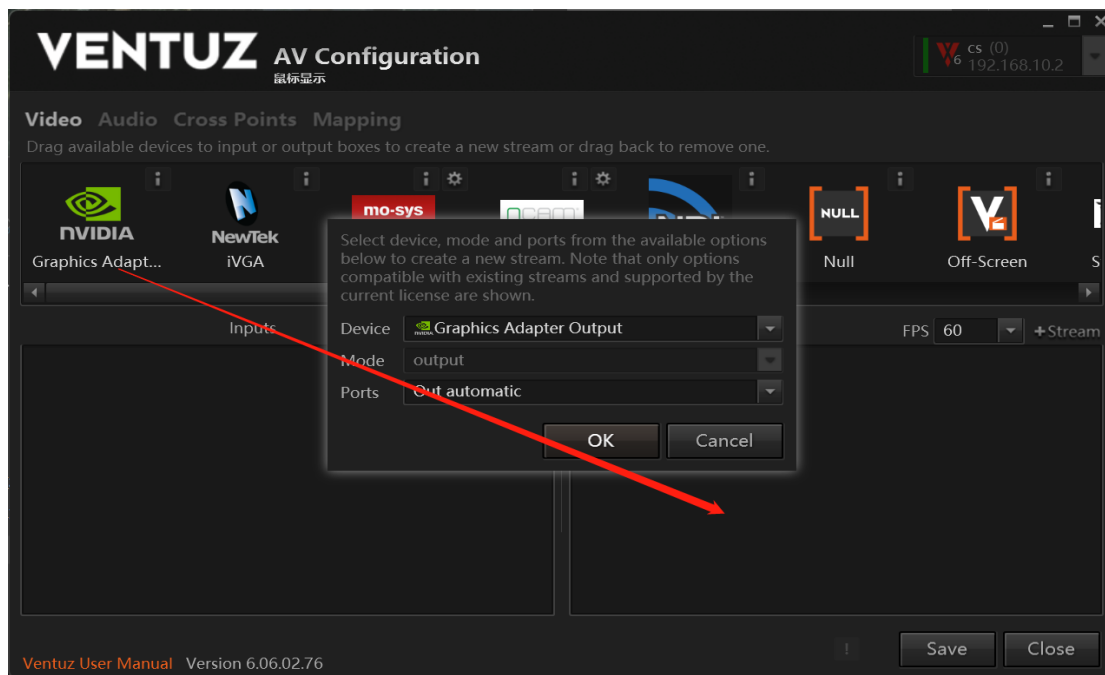
4.2.4. 配置鼠标显示

在红框位置点击加号新建配置（与上个配置名称一样，可以修改个配置名字），然后点击 ok。

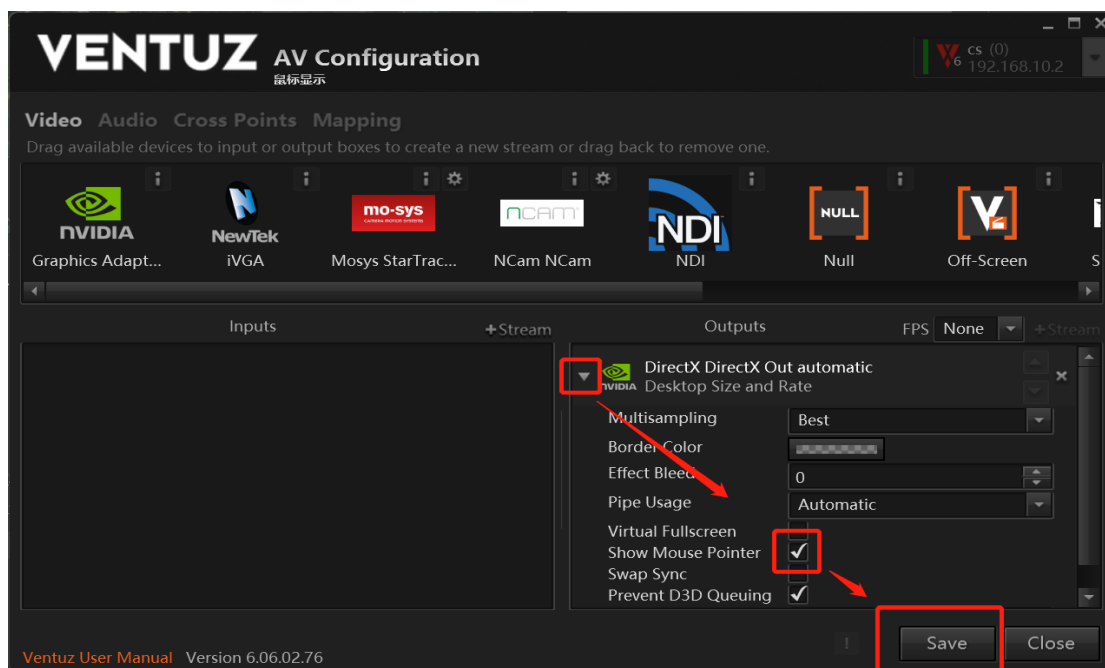


进入界面后，会自动弹出本机显卡输出，点击 ok。

注意：如果没有弹出本机显卡输出，手动拖动本机显卡图标到右下空白栏，会弹出本机显卡输出。



点击小三角展开选项，勾选“Show Mouse Pointer”选项，然后点击 Save。



然后点击界面右下角“Exit 按钮”退出配置。

4.3 NGINX 安装

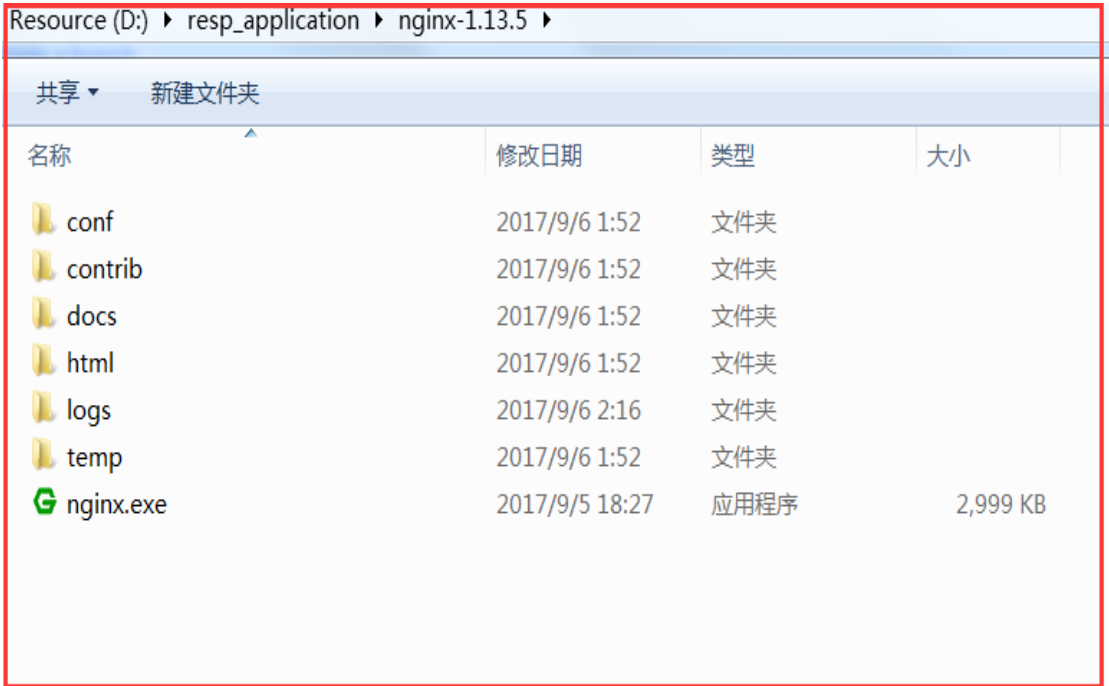
4.3.1. 安装包下载

<https://nginx.org/en/download.html>，找到需要的版本。



4.3.2. 解压安装包

解压下载的安装包，解压后的文件目录结构如下：

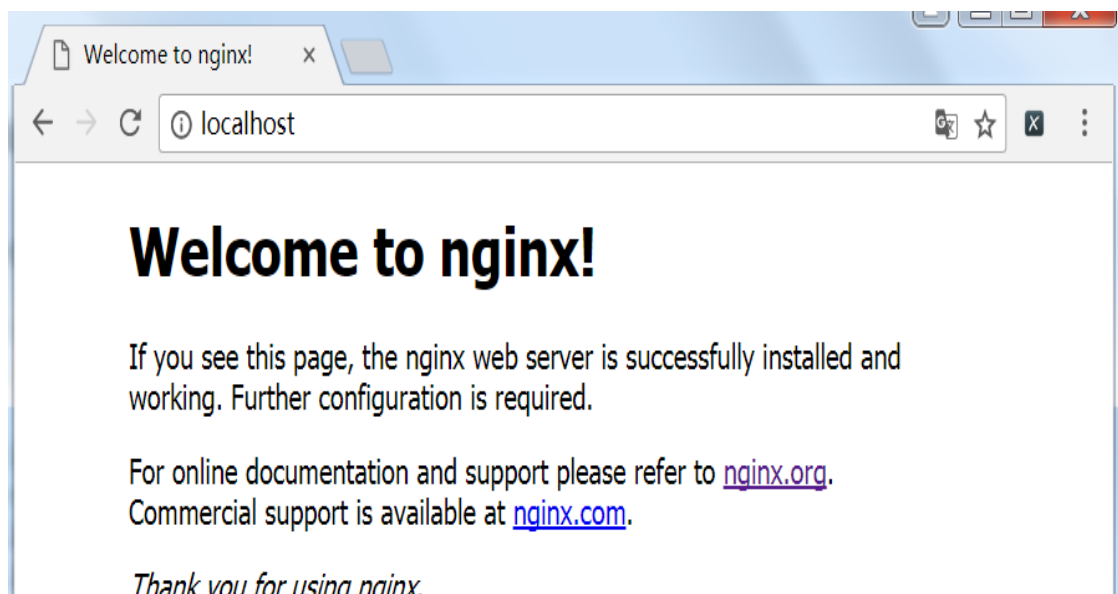


4.3.3. 启动 NGINX

- 1) 直接双击该目录下的 `nginx.exe`，即可启动 `nginx` 服务器。
- 2) 命令行进入该文件夹，执行 `nginx` 命令，也可以直接启动 `nginx` 服务器。

4.3.4. 验证安装结果

打开浏览器，输入地址：`http://localhost`，访问页面，出现如下页面表示访问成功。



4.3.5. 停止服务

命令行进入 `nginx` 根目录，执行如下命令，停止服务器：

- 1) 强制停止 `nginx` 服务器，如果有未处理的数据，则丢弃，**注**间以下路径要改成你实际的路径。

```
D:/nginx-1.13.5> nginx -s stop
```

2) 优雅的停止 nginx 服务器，如果有未处理的数据，等待处理完成之后停止。**注间以下路径要改成你实际的路径。**














```
D:/nginx-1.13.5> nginx -s quit
```

5 系统部署

5.1 集控服务程序

5.1.1. 修改配置

将 IndControl.rar 拷贝至需要安装的地方，用解压工具解压缩 IndControl 压缩包，解压后的目录结果如下：

 config.ini	2022/7/25 21:13	配置设置	1 KB
 ControlSetting.ini	2022/7/24 16:29	配置设置	4 KB
 IndControl.exe	2022/7/25 22:44	应用程序	14,987 KB
 indcontrol_plugin_import.cpp	2022/7/25 22:19	C++ 源文件	1 KB
 libcrypto-1_1-x64.dll	2022/3/23 5:12	应用程序扩展	2,787 KB
 libiconv-2.dll	2022/3/23 5:12	应用程序扩展	1,808 KB
 libintl-9.dll	2022/3/23 5:12	应用程序扩展	465 KB
 libpq.dll	2022/3/23 5:12	应用程序扩展	297 KB
 libpq.lib	2022/5/10 18:08	Object File Library	37 KB
 libssl-1_1-x64.dll	2022/3/23 5:12	应用程序扩展	669 KB
 libwinpthread-1.dll	2022/3/23 5:12	应用程序扩展	52 KB
 Makefile	2022/7/25 22:35	文件	415 KB
 msvcrt120.dll	2022/5/12 15:47	应用程序扩展	942 KB

双击打开 config.ini 文件，将里面的配置改成实际运行环境的配置即可，具体包含以下内容：

```
;位姿系统服务
```

```
[XueLang]
```

```
XueLangIp=127.0.0.1（改成实际运行环境的地址）
```

```
XueLangPort=21000（改成实际运行环境的端口）
```

;PostgreSQL 数据库配置（集控程序取消了数据库访问，此处可任意设置）

[Database]

PgHostIp=127.0.0.1（改成实际运行环境的地址）

PgHostPort=5432（改成实际运行环境的端口）

PgUser=postgres（改成实际运行环境的用户名）

PgPass=123456（改成实际运行环境的用户密码）

PgDatabase=tunneling

;Modbus 本机服务配置

[Modbus]

ModbusPort=502（改成实际运行环境的端口）

;canOpen 服务配置

[CanOpen]

CanOpenIp=127.0.0.1（改成实际运行环境的地址）

CanOpenPort=22004（改成实际运行环境的端口）

;WebSocket 本机服务配置

[WebSocket]

WebSocketPort=8888（改成实际运行环境的端口）


;Modbus 协同控制器

[ModbusSlave]


SlaveIp=127.0.0.1（改成实际运行环境的地址）

SlavePort=503（改成实际运行环境的端口）

5.1.2. 启动程序

双击其中的  IndControl.exe 文件，即可启动集控服务程序，程序自动启动成托盘程序，并开始在后台服务。

5.1.3. 退出程序

将鼠标移植托盘程序处，点击，会弹出 ，将鼠标移植红框所示处，会出现程序名称：协议服务器，在此图标上点击鼠标右键，弹出菜单，点击退出按钮即可结束程序。

5.2 接口服务启动

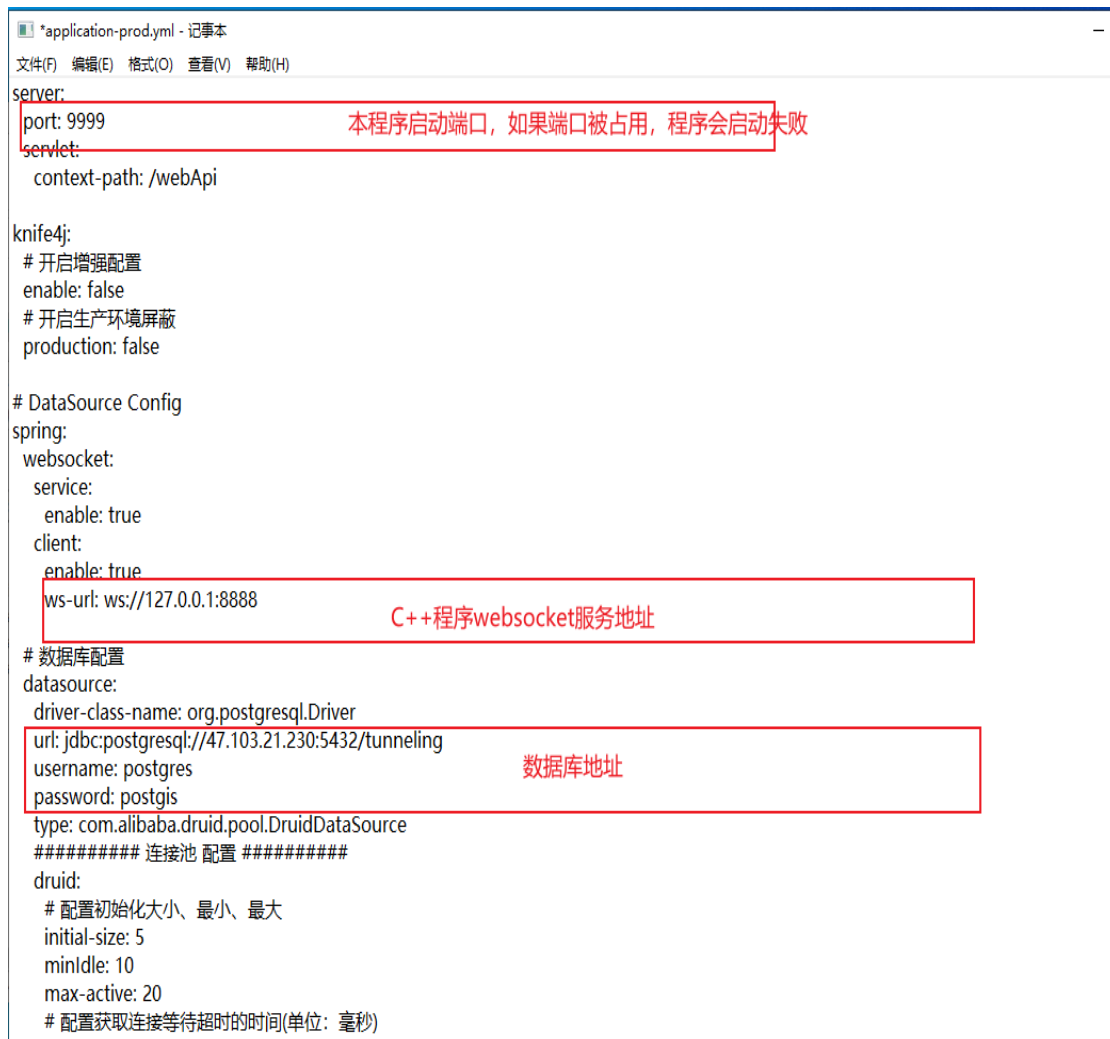
5.2.1. 打开压缩包

将 tunneling-web-api.jar 程序包拷贝至需要处，用解压工具打开压缩软件包，打开后的目录结果结构如下：

com	1 056 219	363 790	2022-08-1...	D drwxr-x...
mybatis.mapper	54 914	12 987	2022-08-1...	D drwxr-x...
application-dev.yml	4 534	2 074	2022-08-1...	-rw-r--r--
application-prod.yml	4 538	2 076	2022-08-1...	-rw-r--r--
application.yml	571	403	2022-08-1...	-rw-r--r--

5.2.2. 编辑 YML 文件

找到 application-prod.yml 文件，点击右键进行编辑，编辑完成保存该文件。



```
*application-prod.yml - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

server:
  port: 9999
  servlet:
    context-path: /webApi

knife4j:
  # 开启增强配置
  enable: false
  # 开启生产环境屏蔽
  production: false

# DataSource Config
spring:
  websocket:
    service:
      enable: true
    client:
      enable: true
  ws-url: ws://127.0.0.1:8888

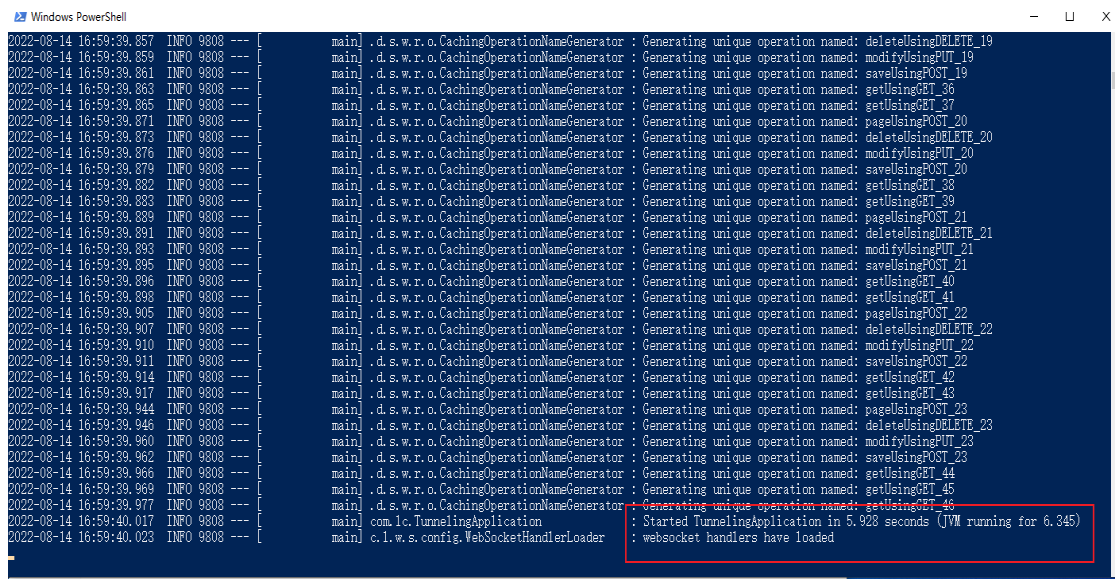
# 数据库配置
datasource:
  driver-class-name: org.postgresql.Driver
  url: jdbc:postgresql://47.103.21.230:5432/tunneling
  username: postgres
  password: postgis
  type: com.alibaba.druid.pool.DruidDataSource
  ##### 连接池 配置 #####
  druid:
    # 配置初始化大小、最小、最大
    initial-size: 5
    minIdle: 10
    max-active: 20
    # 配置获取连接等待超时的时间(单位: 毫秒)
```

5.2.3. 启动程序

首先进入 cmd 命令，然后切换至 tunneling-web-api.jar 所在目录，在命令窗口中，输入：`java -jar tunneling-web-api.jar --spring.profiles.active=prod`，即可启动接口服务程序。

注意，接口服务程序启动时会自动连接集控服务程序，启动接口

服务程序前，请先确保集控服务程序已经启动。

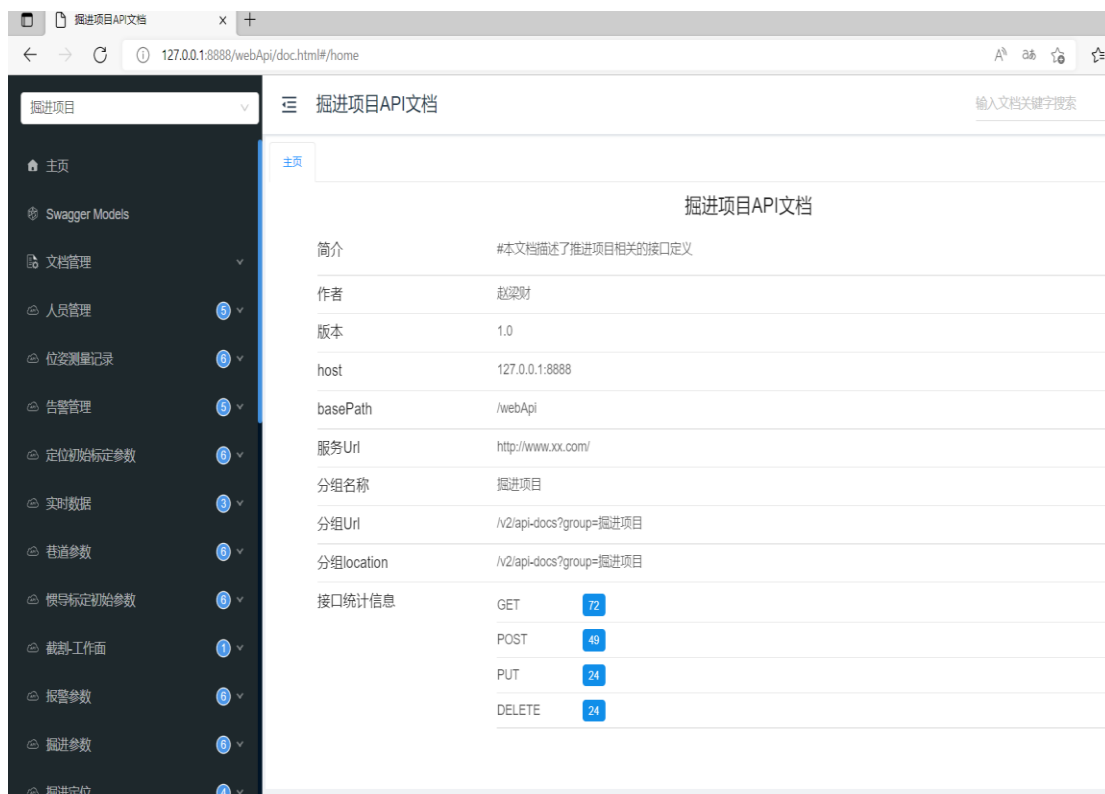


```
2022-03-14 16:59:39.857 INFO 9808 --- [main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: deleteUsingDELETE_19
2022-03-14 16:59:39.859 INFO 9808 --- [main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: modifyUsingPUT_19
2022-03-14 16:59:39.861 INFO 9808 --- [main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: saveUsingPOST_19
2022-03-14 16:59:39.863 INFO 9808 --- [main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: getUsingGET_36
2022-03-14 16:59:39.865 INFO 9808 --- [main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: getUsingGET_37
2022-03-14 16:59:39.871 INFO 9808 --- [main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: pageUsingPOST_20
2022-03-14 16:59:39.873 INFO 9808 --- [main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: deleteUsingDELETE_20
2022-03-14 16:59:39.876 INFO 9808 --- [main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: modifyUsingPUT_20
2022-03-14 16:59:39.879 INFO 9808 --- [main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: saveUsingPOST_20
2022-03-14 16:59:39.882 INFO 9808 --- [main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: getUsingGET_38
2022-03-14 16:59:39.883 INFO 9808 --- [main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: getUsingGET_39
2022-03-14 16:59:39.889 INFO 9808 --- [main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: pageUsingPOST_21
2022-03-14 16:59:39.891 INFO 9808 --- [main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: deleteUsingDELETE_21
2022-03-14 16:59:39.893 INFO 9808 --- [main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: modifyUsingPUT_21
2022-03-14 16:59:39.895 INFO 9808 --- [main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: saveUsingPOST_21
2022-03-14 16:59:39.896 INFO 9808 --- [main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: getUsingGET_40
2022-03-14 16:59:39.898 INFO 9808 --- [main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: getUsingGET_41
2022-03-14 16:59:39.905 INFO 9808 --- [main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: pageUsingPOST_22
2022-03-14 16:59:39.907 INFO 9808 --- [main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: deleteUsingDELETE_22
2022-03-14 16:59:39.910 INFO 9808 --- [main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: modifyUsingPUT_22
2022-03-14 16:59:39.911 INFO 9808 --- [main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: saveUsingPOST_22
2022-03-14 16:59:39.914 INFO 9808 --- [main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: getUsingGET_42
2022-03-14 16:59:39.917 INFO 9808 --- [main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: getUsingGET_43
2022-03-14 16:59:39.944 INFO 9808 --- [main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: pageUsingPOST_23
2022-03-14 16:59:39.946 INFO 9808 --- [main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: deleteUsingDELETE_23
2022-03-14 16:59:39.960 INFO 9808 --- [main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: modifyUsingPUT_23
2022-03-14 16:59:39.962 INFO 9808 --- [main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: saveUsingPOST_23
2022-03-14 16:59:39.966 INFO 9808 --- [main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: getUsingGET_44
2022-03-14 16:59:39.969 INFO 9808 --- [main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: getUsingGET_45
2022-03-14 16:59:39.977 INFO 9808 --- [main] .d.s.w.r.o.CachingOperationNameGenerator : Generating unique operation named: getUsingGET_46
2022-03-14 16:59:40.017 INFO 9808 --- [main] com.lc.TunnelingApplication : Started TunnelingApplication in 5.928 seconds (JVM running for 6.345)
2022-03-14 16:59:40.023 INFO 9808 --- [main] c.l.w.s.config.WebSocketHandlerLoader : websocket handlers have loaded
```

5.2.4. 启动验证

1) 出现上图中红框部份内容即为启动成功。

2) 此外也可以在浏览器地址栏中输入：**http://ip: 端口/项目名称/doc.html**，访问，出现如下界面也表示成功。



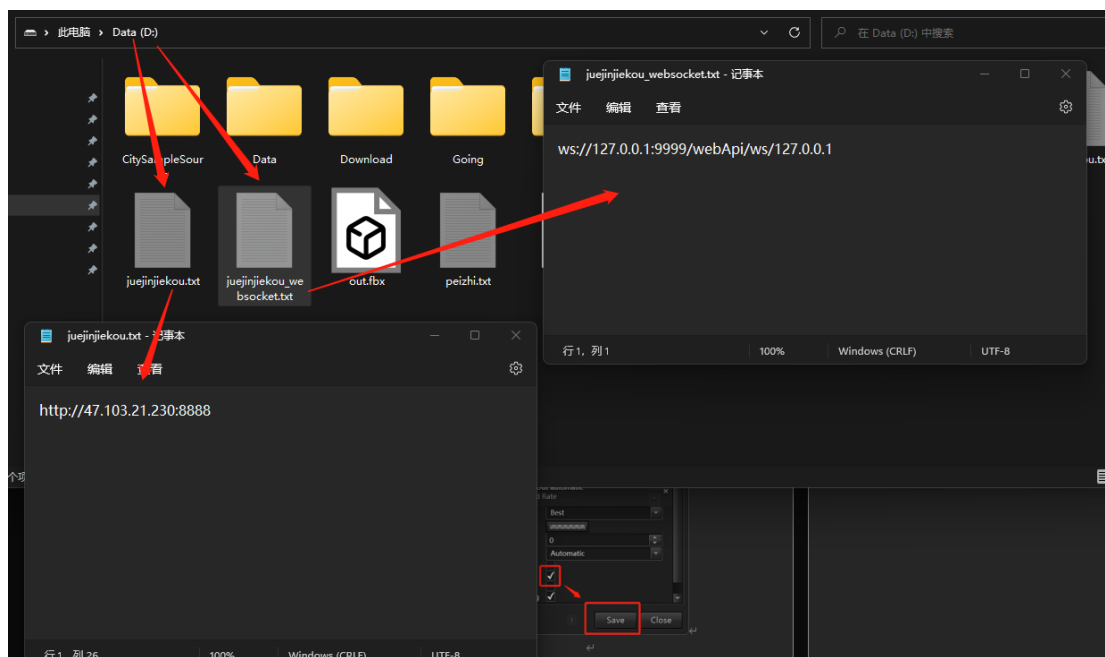
5.3 三维动画

5.3.1. 修改端口地址

1) 打开程序目录下的 juejinjiekou.txt 修改 api 端口地址，保存即可。

2) 打开程序目录下的 juejinjiekou_websocket.txt 修改 websocket 端口地址，保存即可。

注：txt 默认编码为 UTF-8，不要修改。



5.3.2. 打开软件

右键“Open”打开交付的程序文件，进入加载页面，等候加载，

注：若部署为开机自启动则不需要操作此步骤。

5.3.3. 界面操作

鼠标左键上下左右拖动控制场景旋转

鼠标右键上下左右拖动控制场景平移

鼠标中键滚动控制场景缩放

按下键盘空格键复位视角

键盘 1-9 键切换不同视角（30s 无操控回复主视角），0 键切换到主视角

注：0-9 键非小键盘数字键，为字母键上方数字



鼠标移至界面右上角会出现 关闭按钮，点击即可关闭系统



5.3.4. 启动关闭操控

点击启动按钮会弹出是否需要启动，点击确定，发送指定指令到对应接口。



5.3.5. 快捷键大全

Alt 键+F4 键：强制关闭软件

Alt 键+ENTER 键：切换窗口和全屏

5.3.6. 注意事项

启动可视化程序，最好右键“Open”打开，单机或双击会误操作，打开两个及以上可视化程序，会造成交互和接受数据混乱。

Alt+F4 退出软件会弹出错误窗口，关闭即可，对软件本身没影响。

Ventuz 软件安装后，会在桌面右下角任务栏开启 ventuz 服务。安装 360 等管理及杀毒软件会优化掉开机启动的 ventuz 服务，会造成接收不到数据。请确保此服务开启。



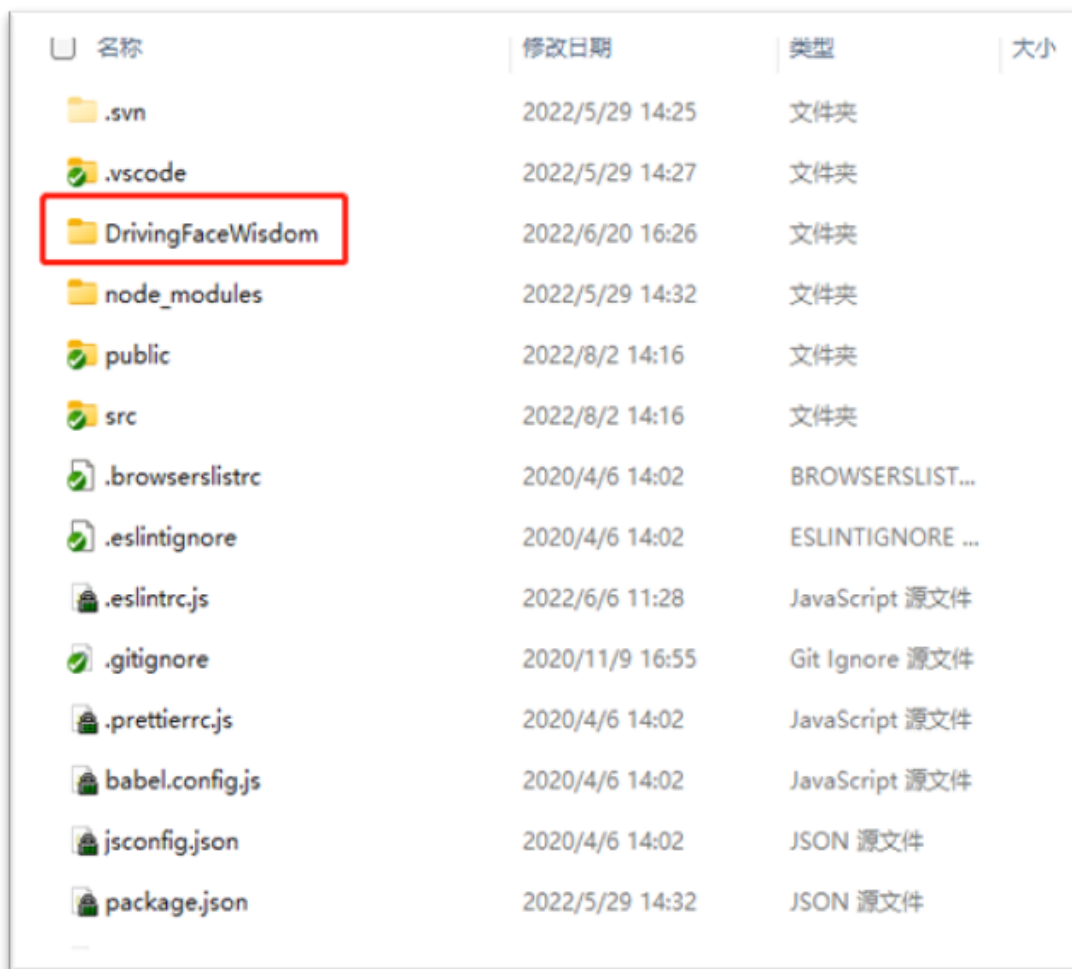
5.4 WEB 部署

5.4.1. 解压程序包

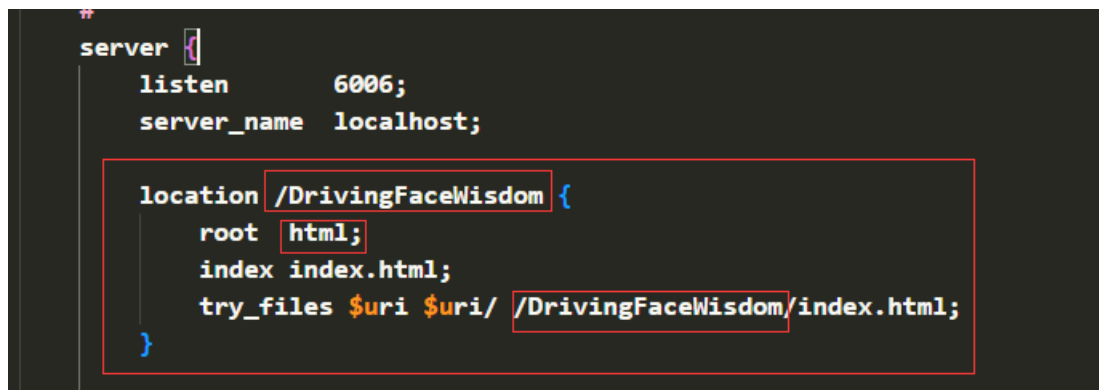
用解压软件解压程序包：DrivingFaceWisdom.rar，得到解压后的文件夹 DrivingFaceWisdom。

5.4.2. 拷贝程序文件

将解压出来的文件夹，拷贝到 nginx/html 文件夹下。



5.4.3. 添加 conf/nginx.config 配置



```
location /DrivingFaceWisdom {  
    root    html;  
    index  index.html;  
    try_files $uri $uri/ /DrivingFaceWisdom/index.html;  
}
```

5.4.4. 验证

打开浏览器，在地址栏中输入访问地址，页面打开即为部署成功。

注意：WEB 需要 API 接口支持，访问前请先确保 5.2 小节中所介绍服务是启动成功的。

6 系统验证

系统验证需确保，所有的网络都是连通的，所有的程序都正常启动好了。

6.1 配置参数发送验证

在 WEB 页面中进行参数配置，点击保存后，检查位姿系统是否成功收到了报文，并且解析后的数据是否与 WEB 页面配置数据一样。

6.2 控制命令发送验证

在 WEB 页面中点击相关控制按钮，检查 CAN 是否接收到了报文，并且控制命令是否正确。

6.3 位姿测量数据接收验证

检查接收到的位姿测量数据是否与位姿系统测量到的数据吻合。

6.4 掘进机工况数据接收验证

检查接收到的掘进机工况数据与实际测量数据是否吻合。

6.5 自移机尾工况数据接收验证

检查接收到的自移机尾工况数据与实际测量数据是否吻合。