

1.激活系统 Windows Server 2016

1、在桌面上选择我的电脑，然后点击鼠标右键，在鼠标右键菜单中单击“属性”，单击更改产品密钥。

PS: 如果桌面上没有显示我的电脑

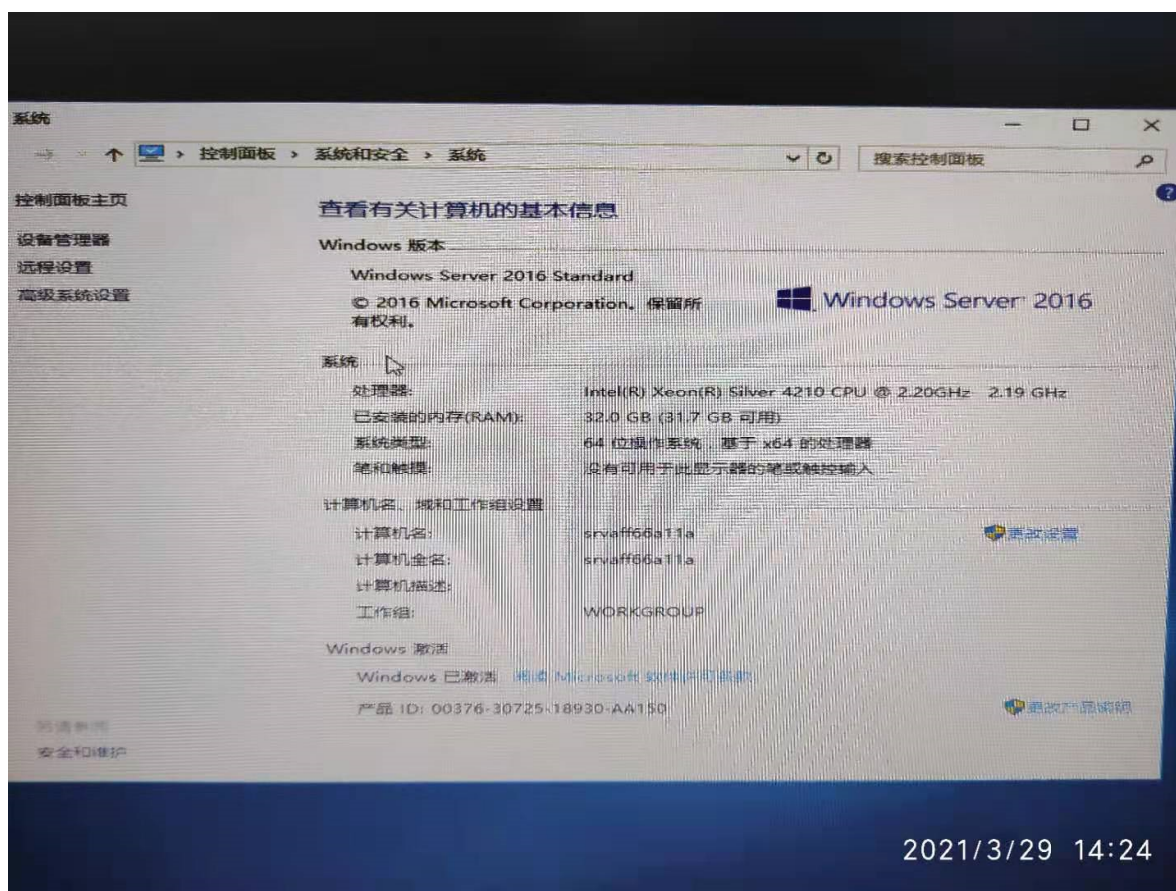
1.点击鼠标右键，在鼠标右键菜单中单击个性化，单击主题，单击桌面图标设置。

2.由于没有激活，这个选项无法选择，可选择激活系统，更改产品密钥。

2、输入Windows系统密钥。

3、输入产品密钥后，点击next完成系统激活。

4、激活后，你可以在我的电脑页面上查看激活状态。



2.配置路由器

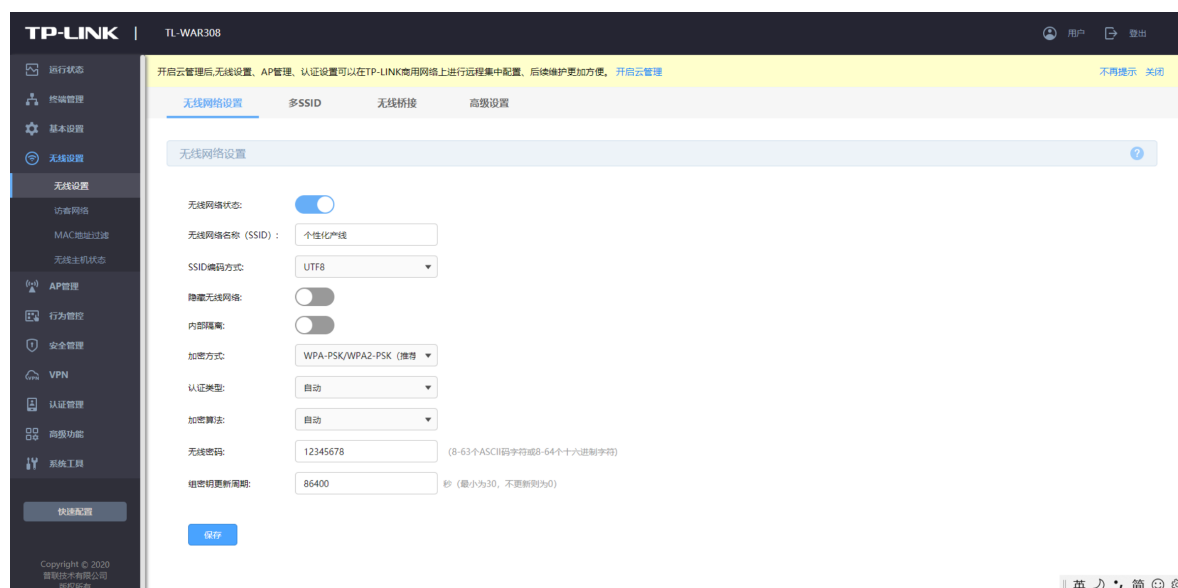
1、连接无线路由器，先把电源接通，然后插上网线，进线插在WAN口（一般是蓝色口），然后跟电脑连接的网线就随便插哪一个LAN口。

2、在浏览器中输入路由器后面的地址，一般是192.168.1.1，进入路由器设置界面，设置路由器的密码。

3、点击界面左侧选项中的“设置向导”，点击下一步，进入上网方式设置，有三种上网方式的选择。选择动态IP，一般电脑直接插上网络就可以用的，上层有DHCP服务器的。

4、下一步后进入到的是无线设置，SSID表示WIFI名称，无线安全选项我们要选择 wpa-psk/wpa2-psk，这样安全，免得轻易让人家破解而蹭网。

5、完成设置，就可以使用路由器了，现在服务器是使用网线连接路由器实现有线连接，用户使用无线连接访问服务。



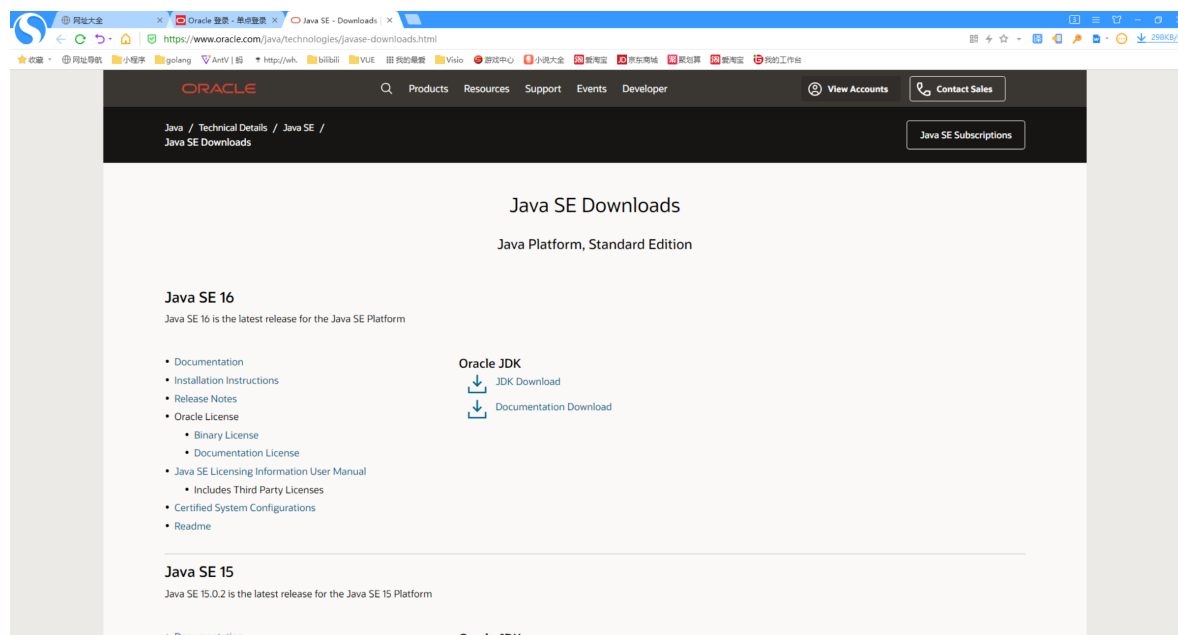
3.部署MES系统

3.1、安装JAVA

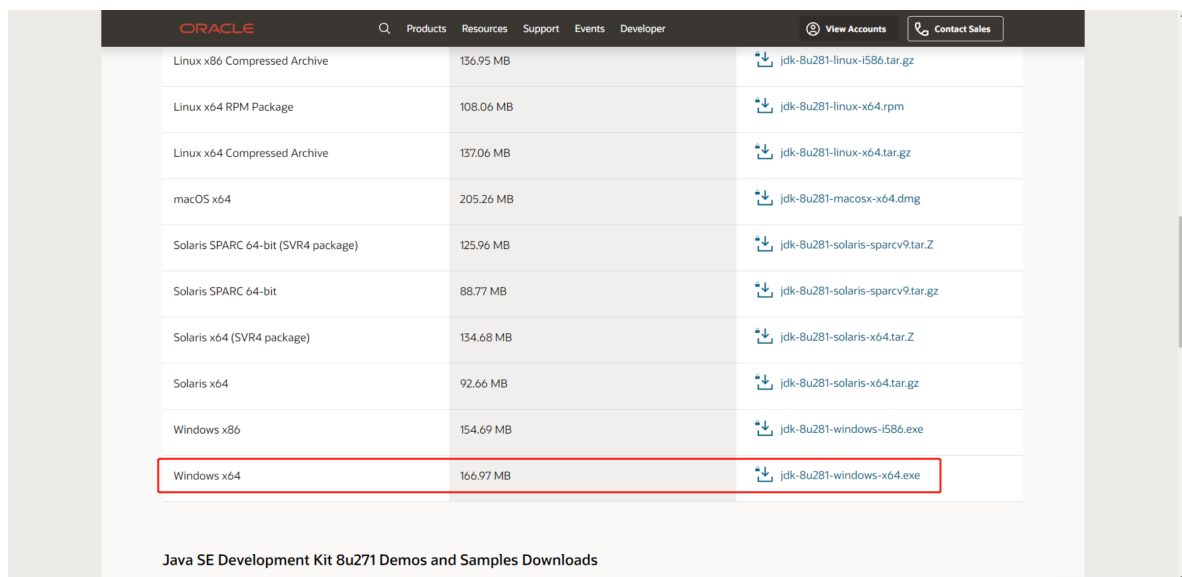
MES系统基于java开发的，如果需要运行MES系统，则需要安装java。

下载JDK

首先我们需要下载java开发工具包JDK，下载地址：<http://www.oracle.com/technetwork/java/javase/downloads/index.html>，点击 JDK Download 按钮：



在下载页面中你需要选择接受许可，并根据自己的系统选择对应的版本，本文以 Window 64位系统为例：



下载后JDK的安装根据提示进行，还有安装JDK的时候也会安装JRE，一并安装就可以了。

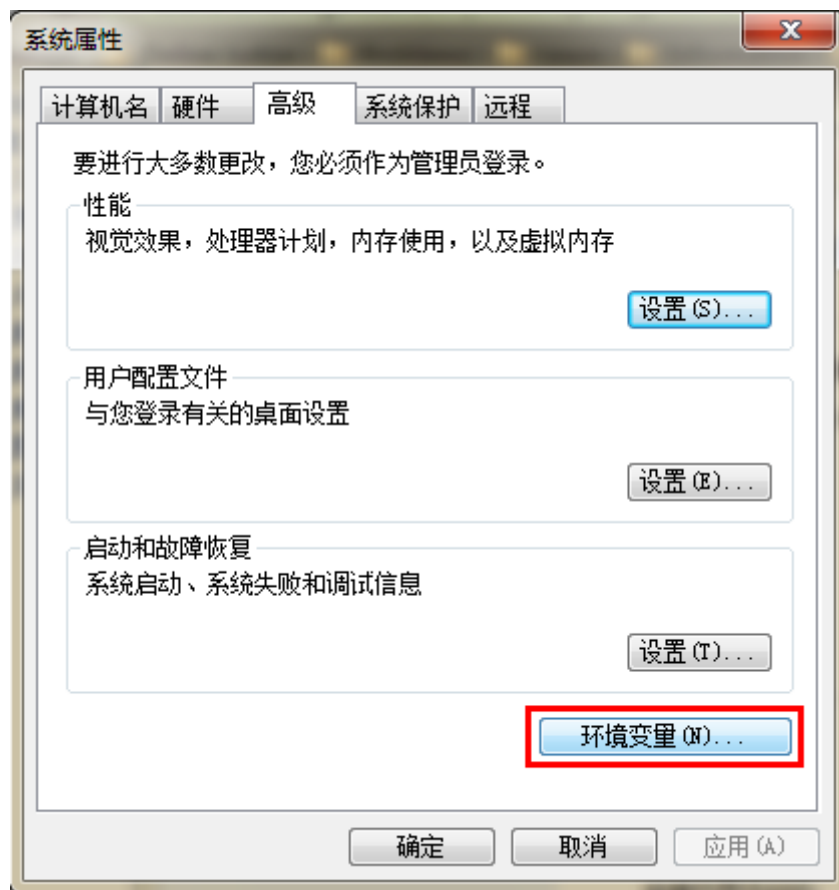
安装JDK，安装过程中可以自定义安装目录等信息，例如我们选择安装目录为 **C:\Program Files (x86)\Java\jdk1.8.0_91**。

配置环境变量

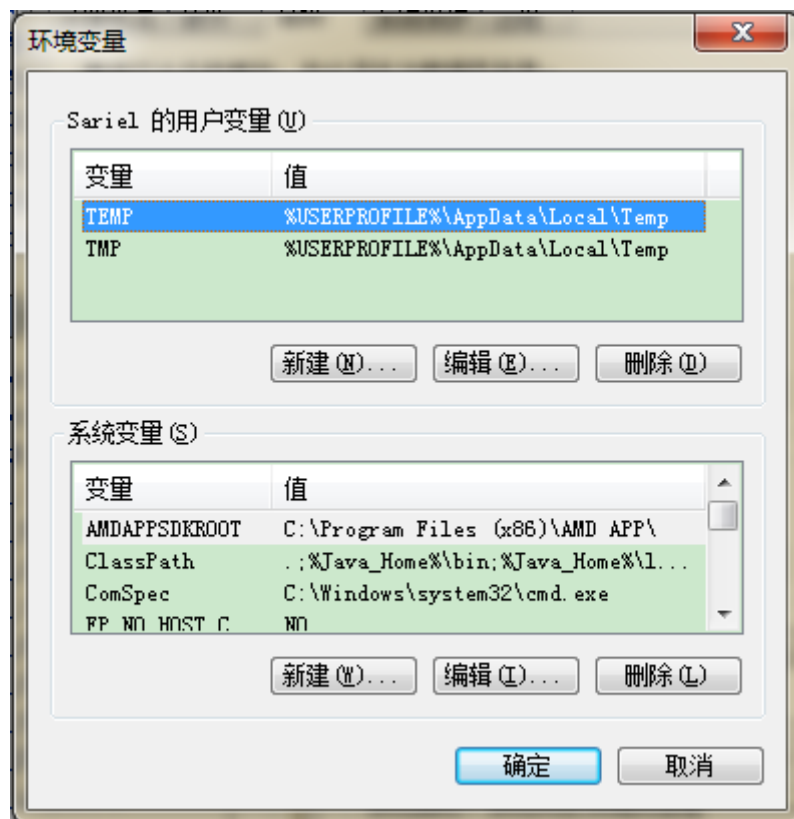
1.安装完成后，右击"我的电脑"，点击"属性"，选择"高级系统设置"；



2.选择"高级"选项卡，点击"环境变量"；



然后就会出现如下图所示的画面：



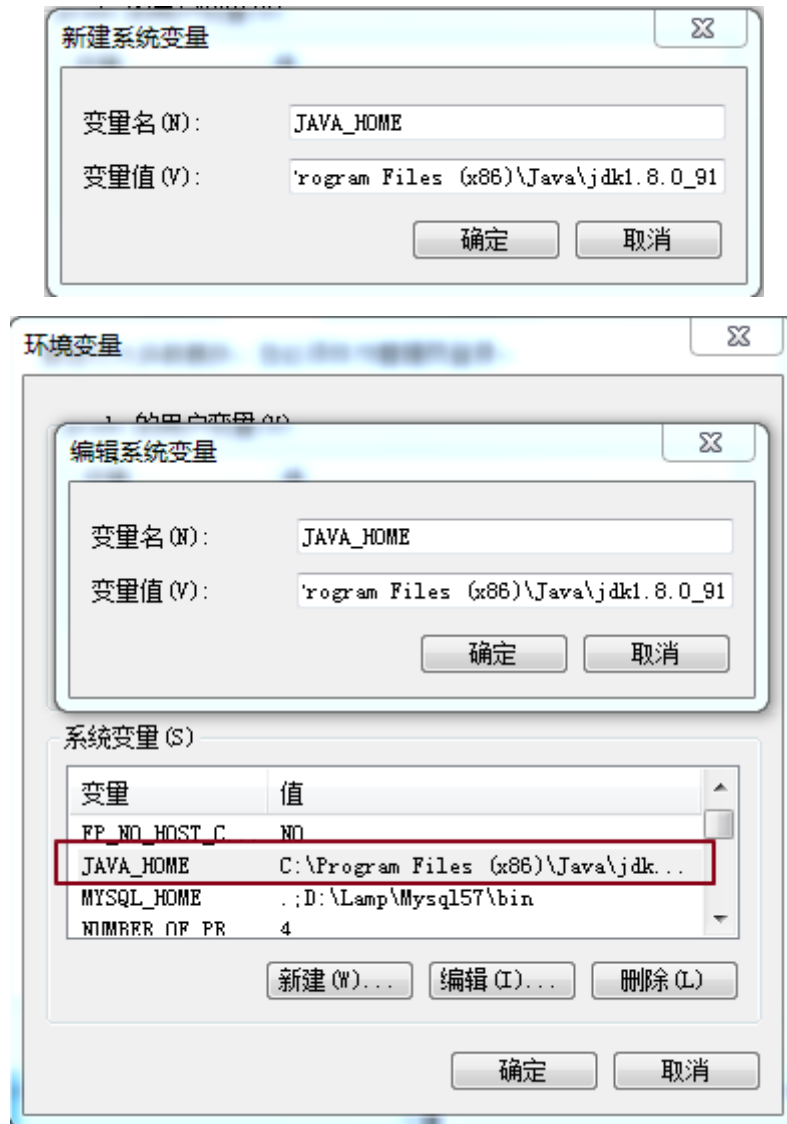
在 "系统变量" 中设置 3 项属性, JAVA_HOME、PATH、CLASSPATH(大小写无所谓),若已存在则点击"编辑", 不存在则点击"新建".

注意：如果使用 1.5 以上版本的 JDK, 不用设置 CLASSPATH 环境变量, 也可以正常编译和运行 Java 程序。

变量设置参数如下：

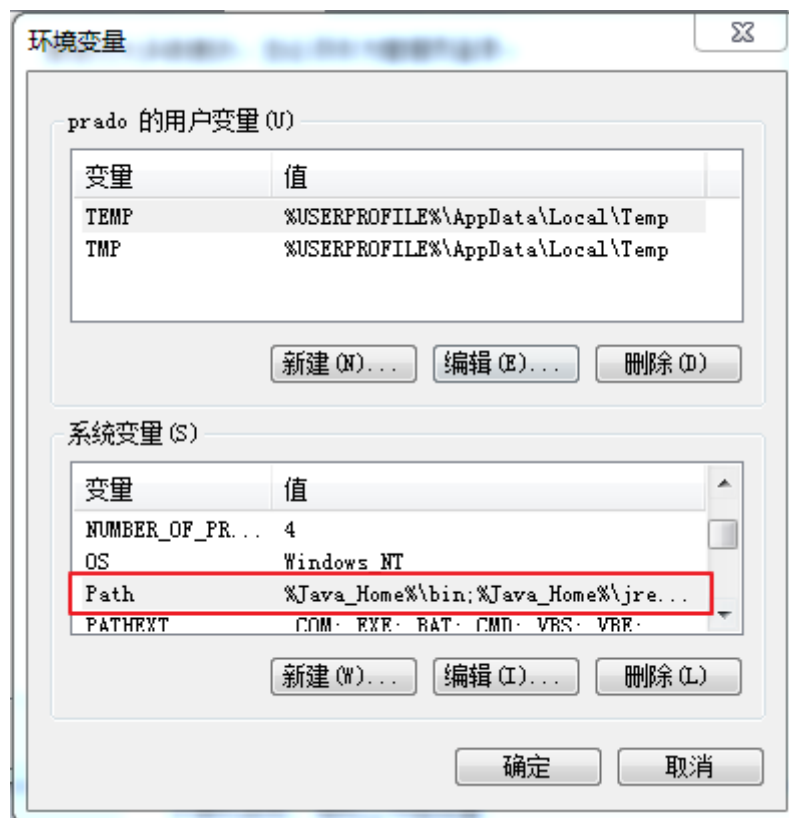
- 变量名: JAVA_HOME
- 变量值: C:\Program Files (x86)\Java\jdk1.8.0_91 // 要根据自己的实际路径配置
- 变量名: CLASSPATH
- 变量值: .;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar; //记得前面有个"."
- 变量名: Path
- 变量值: %JAVA_HOME%\bin;%JAVA_HOME%\jre\bin;

JAVA_HOME 设置



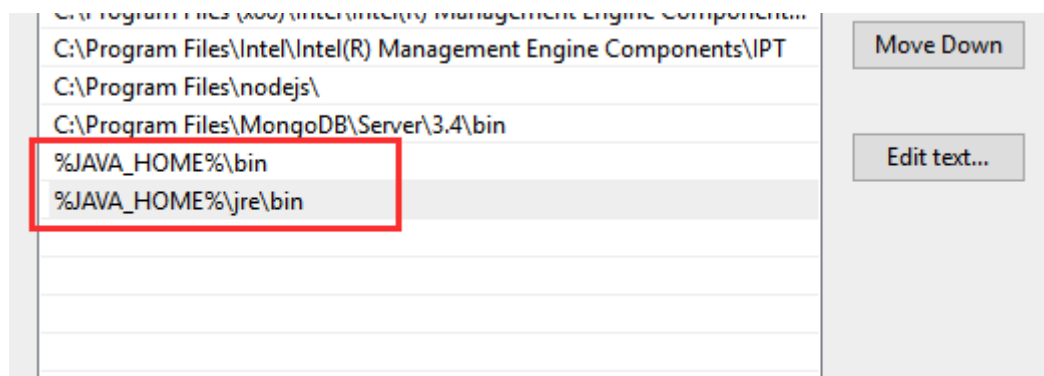
PATH设置





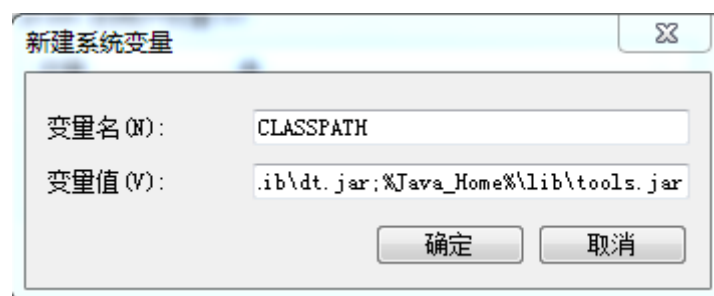
注意：在 Windows10 中，Path 变量里是分条显示的，我们需要将
%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin; 分开添加，否则无法识别：

```
%JAVA_HOME%\bin;
%JAVA_HOME%\jre\bin;
```



更多内容可参考：[Windows 10 配置Java 环境变量](#)

CLASSPATH 设置



这是 Java 的环境配置，配置完成后，你可以启动 Eclipse 来编写代码，它会自动完成java环境的配置。

测试JDK是否安装成功

- 1、"开始"->"运行"，键入"cmd";
- 2、键入命令: `java -version`、`java`、`javac` 几个命令，出现以下信息，说明环境变量配置成功;

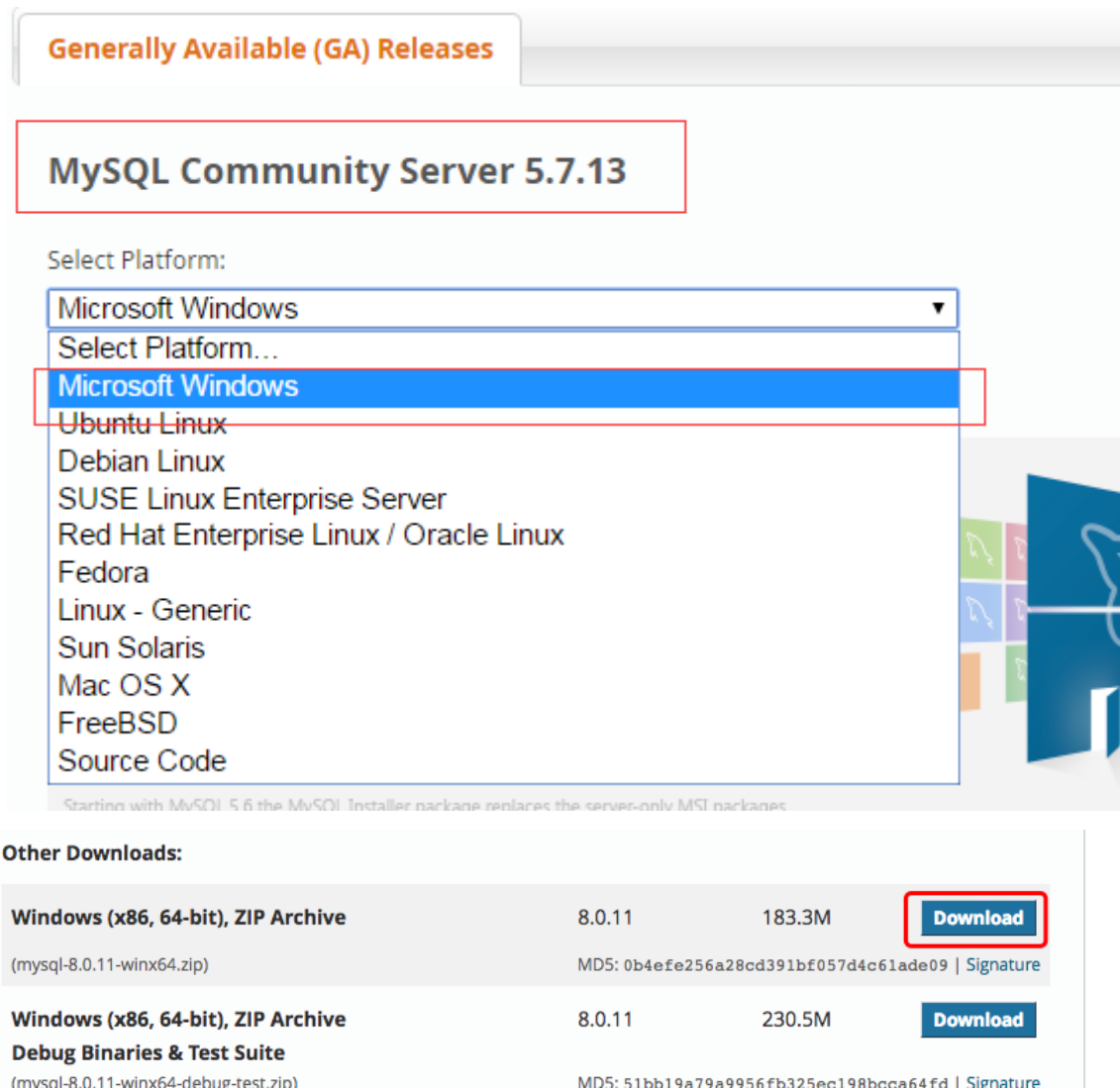
```
C:\Users\prado>java -version
java version "1.8.0_91"
Java(TM) SE Runtime Environment (build 1.8.0_91-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.91-b14, mixed mode)
```

3.2、MYSQL

MES系统数据存储与MYSQL中，如果需要持久化数据，则需要安装MYSQL。

安装MYSQL

Windows 上安装 MySQL 相对来说会较为简单，最新版本可以在 [MySQL下载](#) 中下载中查看



Generally Available (GA) Releases

MySQL Community Server 5.7.13

Select Platform:

Microsoft Windows

Select Platform...

Microsoft Windows

Ubuntu Linux

Debian Linux

SUSE Linux Enterprise Server

Red Hat Enterprise Linux / Oracle Linux

Fedora

Linux - Generic

Sun Solaris

Mac OS X

FreeBSD

Source Code

Starting with MySQL 5.6 the MySQL Installer package replaces the server-only MSI packages

Other Downloads:

Windows (x86, 64-bit), ZIP Archive (mysql-8.0.11-winx64.zip)	8.0.11	183.3M	Download
MD5: 0b4efe256a28cd391bf057d4c61ade09 Signature			
Windows (x86, 64-bit), ZIP Archive Debug Binaries & Test Suite (mysql-8.0.11-winx64-debug-test.zip)	8.0.11	230.5M	Download
MD5: 51bb19a79a9956fb325ec198bcca64fd Signature			

点击 **Download** 按钮进入下载页面，点击下图中的 **No thanks, just start my download.** 就可立即下载：

Login »
using my Oracle Web account

Sign Up »
for an Oracle Web account

MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can signup for a free account following the instructions.

No thanks, just start my download.

下载完后，我们将 zip 包解压到相应的目录，这里我将解压后的文件夹放在 **C:\web\mysql-8.0.11** 下。

接下来我们需要配置下 MySQL 的配置文件

打开刚刚解压的文件夹 **C:\web\mysql-8.0.11**，在该文件夹下创建 **my.ini** 配置文件，编辑 **my.ini** 配置以下基本信息：

```
[client]
# 设置mysql客户端默认字符集
default-character-set=utf8

[mysqld]
# 设置3306端口
port = 3306
# 设置mysql的安装目录
basedir=C:\\web\\mysql-8.0.11
# 设置 mysql数据库的数据的存放目录，MySQL 8+ 不需要以下配置，系统自己生成即可，否则有可能报错
# datadir=C:\\web\\sqldata
# 允许最大连接数
max_connections=20
# 服务端使用的字符集默认为8比特编码的latin1字符集
character-set-server=utf8
# 创建新表时将使用的默认存储引擎
default-storage-engine=INNODB
```

接下来我们来启动下 MySQL 数据库：

以管理员身份打开 cmd 命令行工具，切换目录：

```
cd C:\web\mysql-8.0.11\bin
```

初始化数据库：

```
mysqld --initialize --console
```

执行完成后，会输出 root 用户的初始默认密码，如：

```
...
2018-04-20T02:35:05.464644Z 5 [Note] [MY-010454] [Server] A temporary password
is generated for root@localhost: APWCY5ws&hjQ
...
```

APWCY5ws&hjQ 就是初始密码，后续登录需要用到，你也可以在登陆后修改密码。

输入以下安装命令：

```
mysqld install
```

启动输入以下命令即可：

```
net start mysql
```

注意: 在 5.7 需要初始化 data 目录:

```
cd C:\web\mysql-8.0.11\bin  
mysqld --initialize-insecure
```

初始化后再运行 net start mysql 即可启动 mysql。

登录 MySQL

当 MySQL 服务已经运行时, 我们可以通过 MySQL 自带的客户端工具登录到 MySQL 数据库中, 首先打开命令提示符, 输入以下格式的命名:

```
mysql -h 主机名 -u 用户名 -p
```

参数说明:

- **-h**: 指定客户端所要登录的 MySQL 主机名, 登录本机(localhost 或 127.0.0.1)该参数可以省略;
- **-u**: 登录的用户名;
- **-p**: 告诉服务器将会使用一个密码来登录, 如果所要登录的用户名密码为空, 可以忽略此选项。

如果我们要登录本机的 MySQL 数据库, 只需要输入以下命令即可:

```
mysql -u root -p
```

按回车确认, 如果安装正确且 MySQL 正在运行, 会得到以下响应:

```
Enter password:
```

若密码存在, 输入密码登录, 不存在则直接按回车登录。登录成功后你将会看到 Welcome to the MySQL monitor... 的提示语。

然后命令提示符会一直以 **mysql>** 加一个闪烁的光标等待命令的输入, 输入 **exit** 或 **quit** 退出登录。

遇到的问题

1.MySQL plugin 'caching_sha2_password' cannot be loaded

连接数据库时报错, 是因为新版本的mysql (8.0) root用户使用caching_sha2_password这种新的更安全的验证方式, 因此想要连接的成功的最简单方法是把验证方式修改为之前的方式:

mysql_native_password, 即登陆后输入

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '你的密码';
```

然后查看是否改变 `select user,host,plugin from mysql.user;` 可以看出验证方式已改。

user	host	plugin
mysql.infoschema	localhost	caching_sha2_password
mysql.session	localhost	caching_sha2_password
mysql.sys	localhost	caching_sha2_password
root	localhost	mysql_native_password

2.MySQL 8.0 only full group by 问题处理

出现这个问题是因为MySQL8.0默认的sql_mode设置了ONLY_FULL_GROUP_BY

解决方法：去掉ONLY_FULL_GROUP_BY，设置新的sql_mode值

1.临时生效

a.查看当前sql_mode的值

```
select @@sql_mode;
```

将查询结果复制出来

```
ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION
```

去掉ONLY_FULL_GROUP_BY, (逗号也一并去掉) , 再将剩余的值设置回去

b. 设置新的sql_mode

```
set @@sql_mode
='STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO
_AUTO_CREATE_USER
,NO_ENGINE_SUBSTITUTION';
```

2.永久生效

永久生效需要修改mysql的配置文件，在 `/etc/mysql/my.cnf` 中 `[mysqld]` 下面加上一行。设置sql_mode，设置值与临时设置的值一样。

```
# 设置sql_mode
sql_mode=STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_
ZERO,NO_ENGINE_SUBSTITUTION
```

备注：

Linux下：mysql配置文件在 `/etc/my.cnf`

Windows下：mysql配置文件为 `my.ini`

Docker版本的MySQL，配置文件在docker里`/etc/mysql/my.cnf`

3.3、SVN

1、准备svn的安装文件,下载地址: <https://sourceforge.net/projects/win32svn/>

sourceforge

Search

Browse Enterprise Blog Deals Help

SOLUTION CENTERS Go Parallel Resources Newsletters Cloud Storage Providers Business VoIP Providers Call Cent

DDoS高防IP 300G无限防

2,300,000 用户正在享受阿里云“稳定、安全、易用、低成本”的产品服务!

Report a problem with ad content

Home / Browse / Subversion for Windows

Subversion for Windows

Command line tools, Language bindings, and Apache httpd modules

Brought to you by: [alagazam](#)

Summary Files Reviews Support Wiki Mailing Lists Code News Donate

★ 4.7 Stars (89)

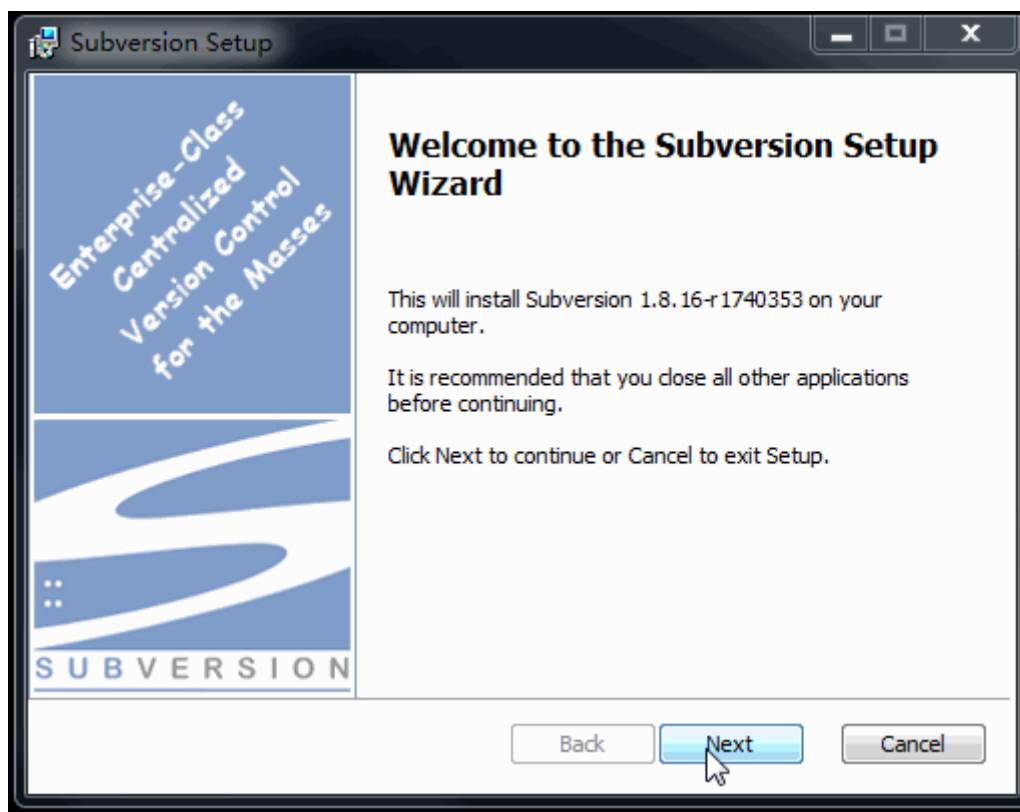
↓ 3,947 Downloads (This Week)

📅 Last Update: 2016-05-04

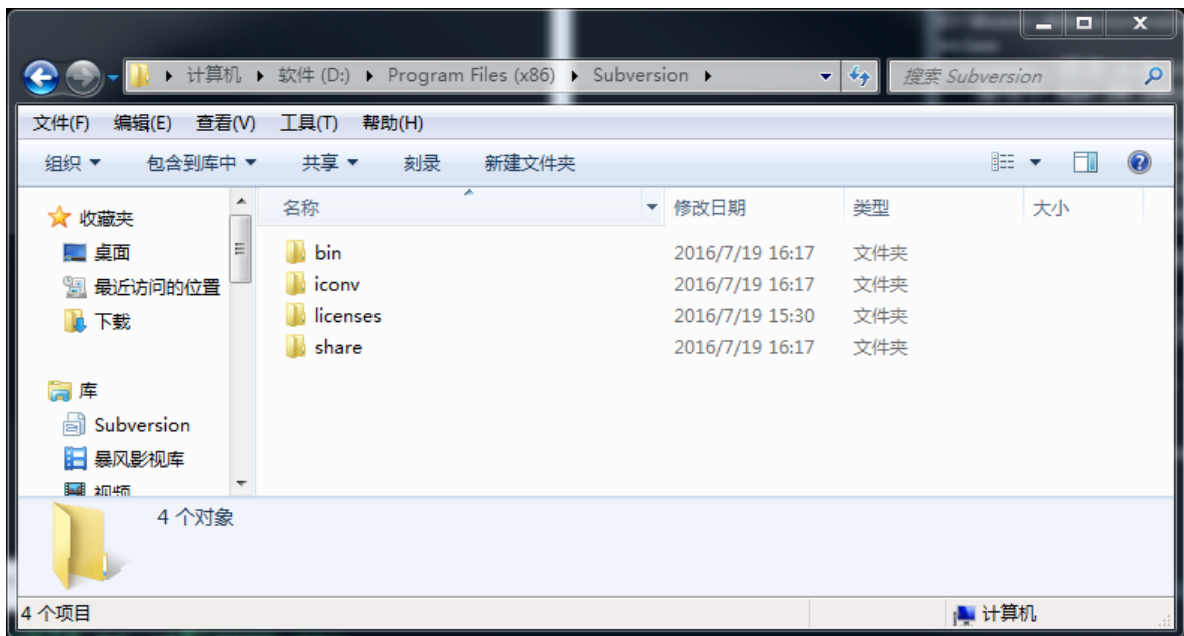
Download
Setup-Subversion-1.8.16.msi

[Browse All Files](#)

2、下载完成后，在相应的盘符中会有一个Setup-Subversion-1.8.16.msi的文件，目前最新的版本是1.8.16，这里就使用这个版本。然后双击安装文件进行安装。我们指定安装在D:\Program Files (x86)\Subversion目录里。



3、查看目录结构



把svn安装目录里的bin目录添加到path路径中，在命令行窗口中输入 `svnserve --help` ,查看安装正常与否。

```
C:\Windows\system32\cmd.exe

C:\Users\Administrator>"D:\Program Files (x86)\Subversion\bin\svnserve.exe" --version
svnserve, 版本 1.8.16 (r1740329)
  编译于 Apr 28 2016, 16:13:44 在 x86-microsoft-windows

Copyright (C) 2016 The Apache Software Foundation.
This software consists of contributions made by many people;
see the NOTICE file for more information.
Subversion is open source software, see http://subversion.apache.org/

下列版本库后端(FS) 模块可用:

* fs_fs : 模块与文本文件(FSFS)版本库一起工作。
* fs_base : 模块只能操作BDB版本库。

Cyrus SASL 认证可用。

C:\Users\Administrator>
```

至此，windows下的SVN安装完成

3.4、Nginx

```
查看Nginx的版本号: nginx -v
启动Nginx: start nginx
快速停止或关闭Nginx: nginx -s stop
正常停止或关闭Nginx: nginx -s quit
配置文件修改重载命令: nginx -s reload
查看windows任务管理器下Nginx的进程命令: tasklist /fi "imagename eq nginx.exe"
```

配置 `conf/nginx.conf`

```

worker_processes 1;

events {
    worker_connections 1024;
}

http {
    include mime.types;
    default_type application/octet-stream;
    sendfile on;
    keepalive_timeout 65;

    server {
        listen 8099;
        server_name localhost;

        error_page 500 502 503 504 /50x.html;

        location = /50x.html {
            root html;
        }

        location /demo {
            proxy_pass_header Server;
            proxy_set_header Host $http_host;
            proxy_redirect off;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Scheme $scheme;
            proxy_pass http://localhost:10083/;
            proxy_http_version 1.1;
            proxy_set_header Upgrade "websocket";
            proxy_set_header Connection "Upgrade";
            proxy_read_timeout 600s;
        }

        location / {
            proxy_pass http://localhost:10082/;
        }
    }
}

```

3.5、Tomcat

安装Tomcat

首先Tomcat8官网下载地址: <http://tomcat.apache.org/download-801>

Tomcat 8 Software Downloads

Welcome to the Apache Tomcat® 8.x software download page. This page provides download links for obtaining the latest versions of Tomcat 8.x software. Users of Tomcat 8.0.x should be aware that the Tomcat team have announced the [end of life date for Tomcat 8.0.x](#). Users of Tomcat 8.0.x should plan accordingly. Note: End of life has been announced for 8.0.x only. 8.5.x is not affected by this announcement.

Quick Navigation

[define v]8.5.34[end] [define w]8.0.53[end] [KEYS](#) | [\[v\]](#) | [\[w\]](#) | [Browse](#) | [Archives](#)

Release Integrity

You **must** [verify](#) the integrity of the downloaded files. We provide OpenPGP signatures for every release file. This signature should be matched against checksums for every release file. After you download the file, you should calculate a checksum for your download, and make sure it is the same as our

Mirrors

You are currently using **[preferred]**. If you encounter a problem with this mirror, please select another mirror. If all mirrors are failing, there are [backups](#) (if any logo) [Logo](#) [end]

Other mirrors:

[v]

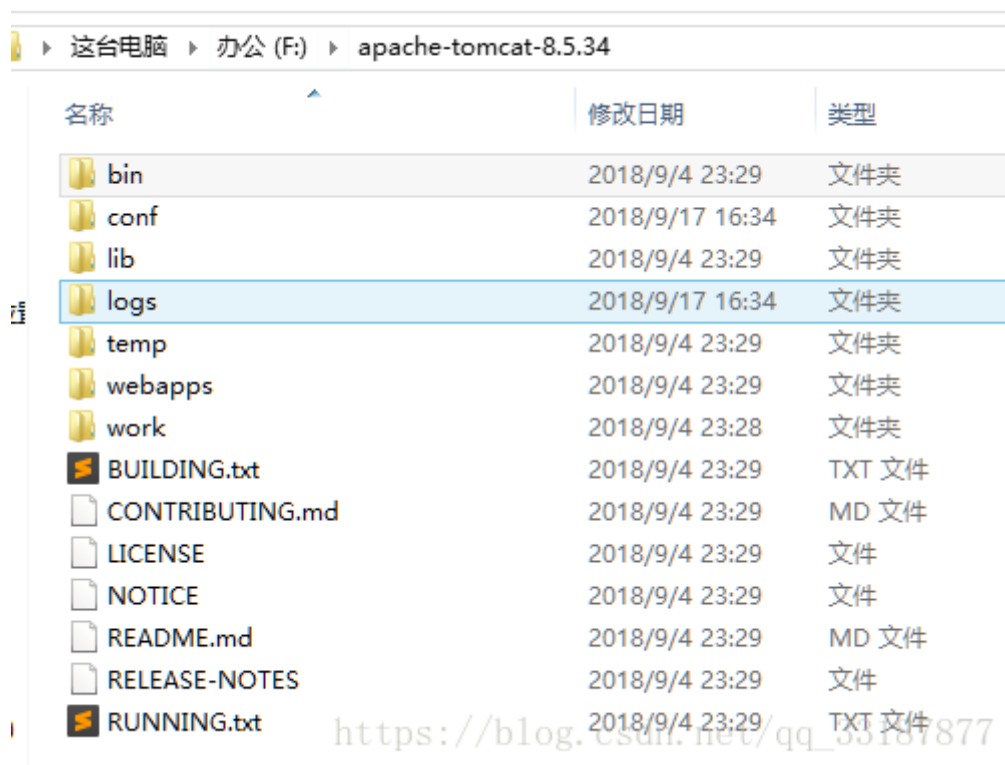
Please see the [README](#) file for packaging information. It explains what every distribution contains.

Binary Distributions

- Core:
 - [zip \(pgp, sha512\)](#)
 - [tar.gz \(pgp, sha512\)](#)
 - [32-bit Windows zip \(pgp, sha512\)](#)
 - [64-bit Windows zip \(pgp, sha512\)](#)
 - [32-bit/64-bit Windows Service Installer \(pgp, sha512\)](#)
- Full documentation:
 - [tar.gz \(pgp, sha512\)](#)
- Deployer:
 - [zip \(pgp, sha512\)](#)
 - [tar.gz \(pgp, sha512\)](#)
- Extras:
 - [JMX Remote jar \(pgp, sha512\)](#)
 - [Web services jar \(pgp, sha512\)](#)
- Embedded:
 - [tar.gz \(pgp, sha512\)](#)
 - [zip \(pgp, sha512\)](#)

https://blog.csdn.net/qq_33187877

根据电脑位数进行下载，我的是64位的，然后下载并解压，我的放到了F盘，解压目录为F:\apache-tomcat-8.5.34，如下图：



这台电脑 > 办公 (F:) > apache-tomcat-8.5.34		
名称	修改日期	类型
bin	2018/9/4 23:29	文件夹
conf	2018/9/17 16:34	文件夹
lib	2018/9/4 23:29	文件夹
logs	2018/9/17 16:34	文件夹
temp	2018/9/4 23:29	文件夹
webapps	2018/9/4 23:29	文件夹
work	2018/9/4 23:28	文件夹
BUILDING.txt	2018/9/4 23:29	TXT 文件
CONTRIBUTING.md	2018/9/4 23:29	MD 文件
LICENSE	2018/9/4 23:29	文件
NOTICE	2018/9/4 23:29	文件
README.md	2018/9/4 23:29	MD 文件
RELEASE-NOTES	2018/9/4 23:29	文件
RUNNING.txt	2018/9/4 23:29	TXT 文件

https://blog.csdn.net/qq_33187877

下面是Tomcat相关的环境变量（正确写法）

变量	值
CATALINA_HOME	F:\apache-tomcat-8.5.34
JAVA_HOME	C:\Program Files\Java\jdk1.8.0_144
PATH	%CATALINA_HOME%\lib;%CATALINA_HOME%\bin

这其中有几点要注意：

- JAVA_HOME中的路径是JDK的路径，别错写成JRE的路径；
- JAVA_HOME中的路径不能使用分号结尾，如：C:\Program Files\Java\jdk1.8.0_144;
- CATALINA_BASE, CATALINA_HOME中的路径不能以“\”结尾
- Path路径里配置环境变量时要确保变量前分号是英文状态下

验证Tomcat

最后运行startup.bat文件，然后在浏览器中输入<http://localhost:8080/>，看能否进入下面页面，同时要保持开启startup.bat文件打开的dos窗口

遇到的问题

1.闪退

闪退查看错误信息方法：

- 右键编辑闪退的文件
- 在最后一行的下一行写上“pause”（暂停）
- 再次运行文件，这时DOS界面会显示错误原因

```

C:\Windows\system32\cmd.exe
Using CATALINA_BASE:   "F:\apache-tomcat-8.5.34\11"
Using CATALINA_HOME:   "F:\apache-tomcat-8.5.34"
Using CATALINA_TMPDIR: "F:\apache-tomcat-8.5.34\11\temp"
Using JRE_HOME:        "C:\Program Files\Java\jdk1.8.0_144"
Using CLASSPATH:       "F:\apache-tomcat-8.5.34\bin\bootstrap.jar;F:\apache-tomcat-8.5.34\bin\tomcat-juli.jar"
九月 17, 2018 5:06:26 下午 org.apache.catalina.startup.Catalina stopServer
严重: Catalina.stop:
java.io.FileNotFoundException: F:\apache-tomcat-8.5.34\11\conf\server.xml (系统找不到指定的路径。)
    at java.io.FileInputStream.open(Native Method)
    at java.io.FileInputStream.open(FileInputStream.java:195)
    at java.io.FileInputStream.<init>(FileInputStream.java:138)
    at org.apache.catalina.startup.Catalina.stopServer(Catalina.java:474)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:498)
    at org.apache.catalina.startup.Bootstrap.stopServer(Bootstrap.java:406)
    at org.apache.catalina.startup.Bootstrap.main(Bootstrap.java:498)

请按任意键继续. . .

```

我之前就是因为变量CATALINA_BASE的路径写错了...才会报的错...

2.端口占用

netstat -ano | findstr "8080" 查看被占用端口对应的PID

taskkill /pid 20020 结束该进程

- /f: 强制结束正在运行的进程。注意：远程进程将忽略此参数; 所有远程进程都被强制终止。
- /t: 终止指定的进程以及由它启动的任何子进程。终止指定的进程以及由它启动的任何子进程。
- /pid: 进程 ID
- /im: 进程名称

3.tomcat日志乱码

方法打开 apache-tomcat/conf/logging.properties 文件，把里面能看得到的UTF-8全部改成GBK

4.Springboot项目部署到外部Tomcat

1 在pom.xml依赖配置中新增 <packaging>war</packaging>

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <artifactId>MesAdmin</artifactId>
  <packaging>war</packaging>
  <name>MesAdmin</name>
  <description>Mes Admin Model</description>

  <parent>
    <groupId>com.orientech</groupId>
    <artifactId>CloudMesProject</artifactId>
    <version>0.0.1-SNAPSHOT</version>
  </parent>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <project.reporting.outputEncoding>UTF-
8</project.reporting.outputEncoding>
    <java.version>1.8</java.version>
    <skipTests>true</skipTests>
  </properties>

  <dependencies>
    .....
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-maven-plugin</artifactId>
        <configuration>
          <fork>true</fork><!--热部署：注意要修改这里-->
        </configuration>
      </plugin>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-resources-plugin</artifactId>
        <configuration>
          <delimiters>
            <delimiter>@</delimiter>
          </delimiters>
          <useDefaultDelimiters>>false</useDefaultDelimiters>
        </configuration>
      </plugin>
    </plugins>
    <resources>
```

```

        <resource>
            <directory>src/main/resources</directory>
            <includes>
                <include>**/*.properties</include>
                <include>**/*.xml</include>
                <include>**/*.xls</include>
                <include>**/*.xlsx</include>
            </includes>
            <filtering>>false</filtering>
        </resource>
        <resource>
            <directory>src/main/resources</directory>
            <!-- 是否替换@xx@表示的maven properties属性值 -->
            <filtering>>true</filtering>
            <includes>
                <include>application.yml</include>
                <include>application-${profileActive}.yml</include>
            </includes>
        </resource>
    </resources>
</build>

</project>

```

2 在springBoot启动类的同级目录添加一个继承SpringBootServletInitializer 的类，该类在idea开发工具启动springBoot main函数时不会被初始化，在Tomcat启动时才会被初始化

```

import org.springframework.boot.builder.SpringApplicationBuilder;
import org.springframework.boot.web.support.SpringBootServletInitializer;

public class ServletInitializer extends SpringBootServletInitializer {

    public ServletInitializer() {
        System.out.println("初始化ServletInitializer");
    }

    @Override
    protected SpringApplicationBuilder configure(SpringApplicationBuilder application) {
        //application.sources(MyApplication.class)中的类名必须是启动类的类名，不然会无效
        return application.sources(MyApplication.class); //MyApplication是启动类名
    }
}

```

3 mvn clean compile package -P dev -Dmaven.test.skip=true 就可以在项目的target目录下生成对应的war

-P dev 选择运行环境 如POM文件中的\${profileActive} 如application文件中@profileActive@

4 把生成的war放在Tomcat的webapps目录下，Tomcat的端口号最好配置成跟springBoot项目配置文件的端口号一样，不然项目运行后访问会以Tomcat的端口号为准

5 运行并访问，运行时初始化继承SpringBootServletInitializer的类

6 在浏览器中访问对应控制器的请求，访问时跟idea开发工具时的请求方式不一样，在Tomcat运行后访问方式需要带上war解压后的文件夹根目录名称，如下图，访问方式需要类似这样<http://localhost:8090/basetest-1.0/getAssignBean>

5.Tomcat部署多个服务

在conf/server.xml修改一下内容

```
<Service name="Catalina">
  <Connector port="10083" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443" />
  <Engine name="Catalina" defaultHost="localhost">
    <Realm className="org.apache.catalina.realm.LockOutRealm">
      <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
        resourceName="UserDatabase"/>
    </Realm>

    <Host name="localhost" appBase="webapps"
      unpackWARs="true" autoDeploy="true">

      <Valve className="org.apache.catalina.valves.AccessLogValve"
        directory="logs"
        prefix="localhost_access_log" suffix=".txt"
        pattern="%h %l %u %t &quot;%r&quot; %s %b" />

    </Host>
  </Engine>
</Service>
<Service name="Catalina2">
  <Connector port="10082" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443" />
  <Engine name="Catalina" defaultHost="localhost">
    <Realm className="org.apache.catalina.realm.LockOutRealm">
      <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
        resourceName="UserDatabase"/>
    </Realm>

    <Host name="localhost" appBase="webapps2"
      unpackWARs="true" autoDeploy="true">

      <Valve className="org.apache.catalina.valves.AccessLogValve"
        directory="logs"
        prefix="localhost_access_log" suffix=".txt"
        pattern="%h %l %u %t &quot;%r&quot; %s %b" />

    </Host>
  </Engine>
</Service>
```

6.war包部署到服务器后,如何直接访问,而不需要在地址后面加war包名

<http://localhost:10083/MesAdmin-0.0.1-SNAPSHOT/pla/workorder/op/test>

<http://localhost:10082/MesStatic-0.0.1-SNAPSHOT/login>

-->

<http://localhost:10083//pla/workorder/op/test>

<http://localhost:10082/login>

方式一:修改服务器Tomcat的server.xml配置

添加

保存配置后重启Tomcat服务, 此时访问地址是:服务器ID:端口, 若以购买域名且解析完成,就可以自己访问网址了

方式二:删除Tomcat内ROOT文件夹

Tomcat启动时会去webapps下访问默认ROOT目录下的应用程序, 这是conf/server.xml配好的默认规则。

将webapps目录下ROOT文件夹删除掉, 复制war包到webapps目录下, 更名为ROOT.war, 重启Tomcat

3.6、Redis

安装Redis

下载地址: <https://github.com/tporadowski/redis/releases>。

Redis 支持 32 位和 64 位。这个需要根据你系统平台的实际情况选择, 这里我们下载 **Redis-x64-xxx.zip**压缩包到 C 盘, 解压后, 将文件夹重新命名为 **redis**。

Downloads

 Redis-x64-3.2.100.msi	5.8 MB
 Redis-x64-3.2.100.zip	4.98 MB
 Source code (zip)	
 Source code (tar.gz)	

打开文件夹, 内容如下:

名称	修改日期	类型	大小
dump.rdb	2017/7/20 14:34	RDB 文件	15
EventLog.dll	2016/7/1 16:27	应用程序扩展	1
Redis on Windows Release Notes.do...	2016/7/1 16:07	Microsoft Office...	13
Redis on Windows.docx	2016/7/1 16:07	Microsoft Office...	17
redis.windows.conf	2016/7/1 16:07	CONF 文件	48
redis.windows-service.conf	2016/7/1 16:07	CONF 文件	48
redis-benchmark.exe	2016/7/1 16:28	应用程序	400
redis-benchmark.pdb	2016/7/1 16:28	PDB 文件	4,268
redis-check-aof.exe	2016/7/1 16:28	应用程序	251
redis-check-aof.pdb	2016/7/1 16:28	PDB 文件	3,436
redis-cli.exe	2016/7/1 16:28	应用程序	488
redis-cli.pdb	2016/7/1 16:28	PDB 文件	4,420
redis-server.exe	2016/7/1 16:28	应用程序	1,628
redis-server.pdb	2016/7/1 16:28	PDB 文件	6,916
Windows Service Documentation.docx	2016/7/1 9:17	Microsoft Office...	14

打开一个 **cmd** 窗口 使用 **cd** 命令切换目录到 **C:\redis** 运行:

```
redis-server.exe redis.windows.conf
```

如果想方便的话, 可以把 **redis** 的路径加到系统的环境变量里, 这样就省得再输路径了, 后面的那个 **redis.windows.conf** 可以省略, 如果省略, 会启用默认的。输入之后, 会显示如下界面:

```

管理员: C:\Windows\system32\cmd.exe - redis-server.exe redis.windows.conf

C:\redis>redis-server.exe redis.windows.conf

Redis 3.2.100 (00000000/0) 64 bit

Running in standalone mode
Port: 6379
PID: 15292

http://redis.io

[15292] 25 Nov 14:11:13.469 # Server started, Redis version 3.2.100
[15292] 25 Nov 14:11:13.469 * The server is now ready to accept connections on p
ort 6379
  
```

验证Redis

这时候另启一个 **cmd** 窗口, 原来的不要关闭, 不然就无法访问服务端了。切换到 **redis** 目录下运行:

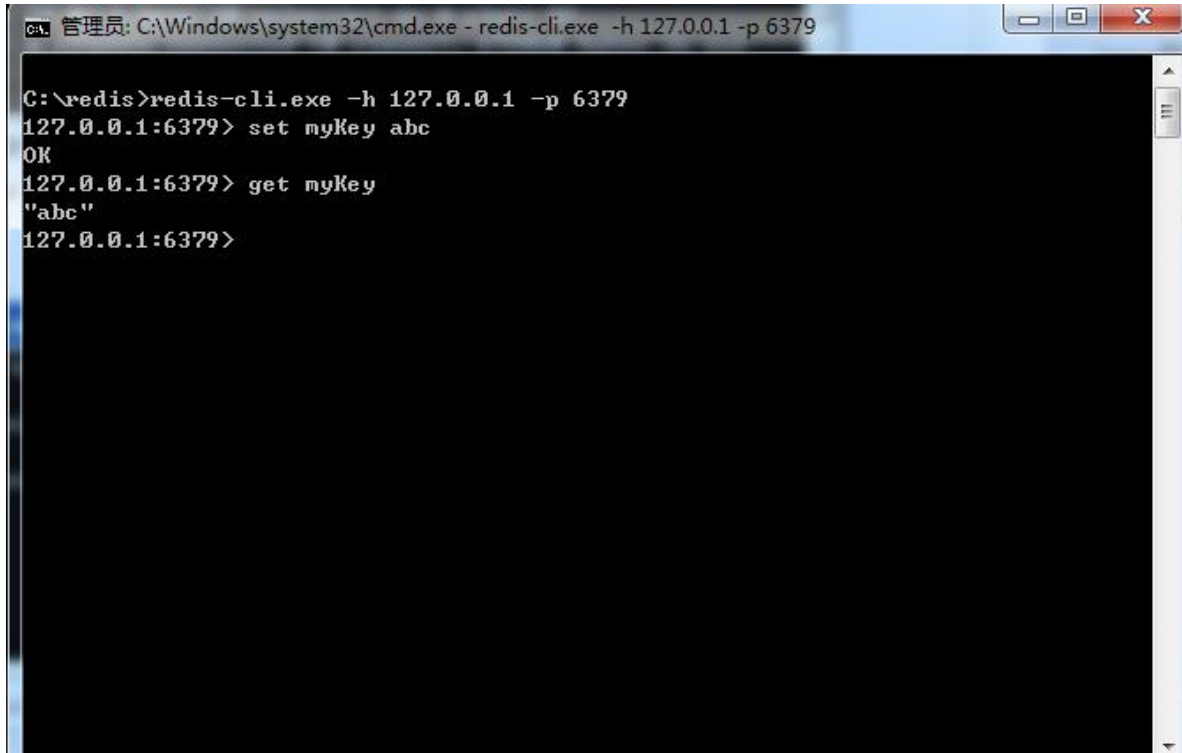
```
redis-cli.exe -h 127.0.0.1 -p 6379
```

设置键值对:

```
set myKey abc
```

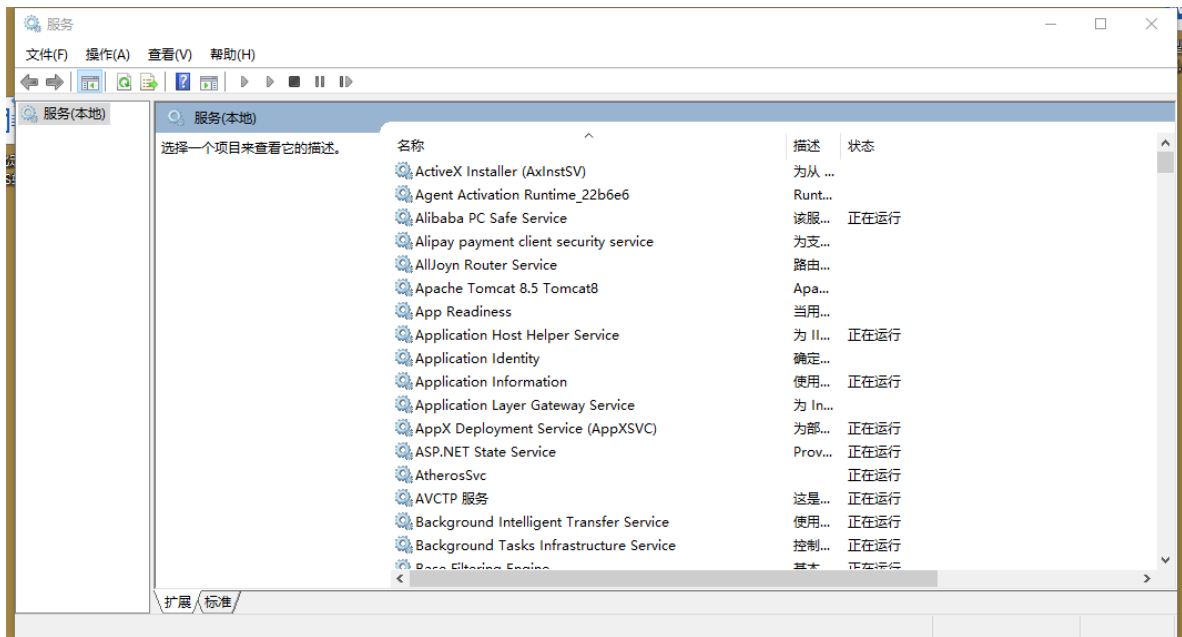
取出键值对:

```
get myKey
```



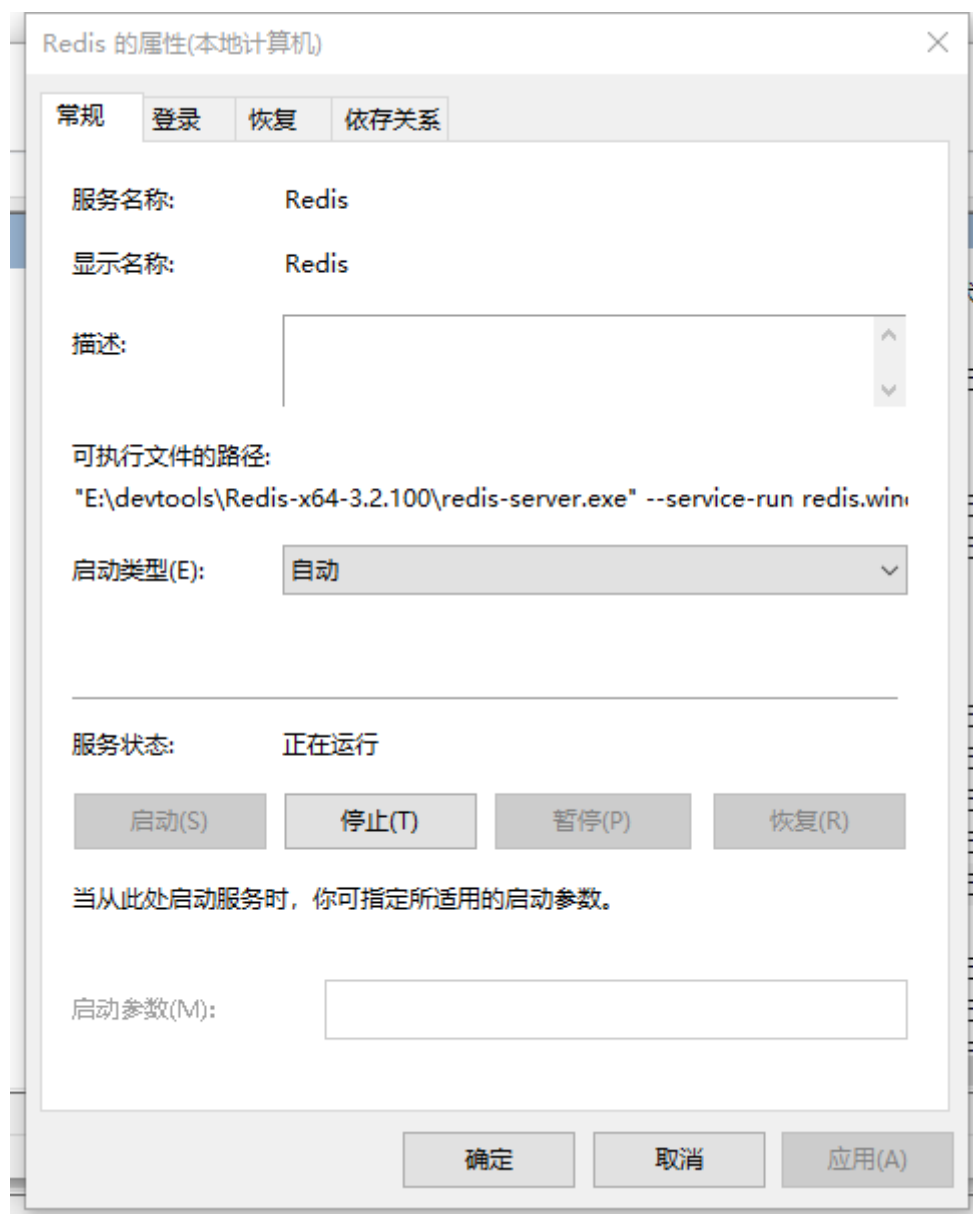
开机自启动Redis

首先发现服务中是没用这个Redis服务的。 `cmd -> services.msc` 打开服务



添加服务: `redis-server.exe--service-install redis.windows-service.conf`

输入命令之后没有报错，表示成功了，刷新服务，会看到多了一个redis服务。在属性中设置启动类型为自动即可。



4.工具

4.1 Markdown 编辑器 —— Typora

Markdown

Markdown 是一种轻量级标记语言，它允许人们使用易读易写的纯文本格式编写文档。Markdown 语言在 2004 年由约翰·格鲁伯（英语：John Gruber）创建。Markdown 编写的文档可以导出 HTML、Word、图像、PDF、Epub 等多种格式的文档。Markdown 编写的文档后缀为 **.md**, **.markdown**。

Markdown 教程：<https://www.runoob.com/markdown/md-tutorial.html>

Typora

Typora是一款轻便简洁的Markdown编辑器，支持即时渲染技术，这也是与其他Markdown编辑器最显著的区别，将写作与预览窗口相结合，为用户呈现所见即所得的编辑模式，支持Markdown的各种基础语法，支持快捷键操作，是好用的一款Markdown编辑器。即时渲染使得你写Markdown就想是写Word文档一样流畅自如，不像其他编辑器的有编辑栏和显示栏。

Typora 官网: <https://typora.io/>

下载地址: <https://typora.io/#windows>

Typora-setup-x64.exe

4.2 Google 浏览器

下载地址: <https://www.google.cn/chrome/>

ChromeSetup.exe

4.3 API调试工具 —— postman

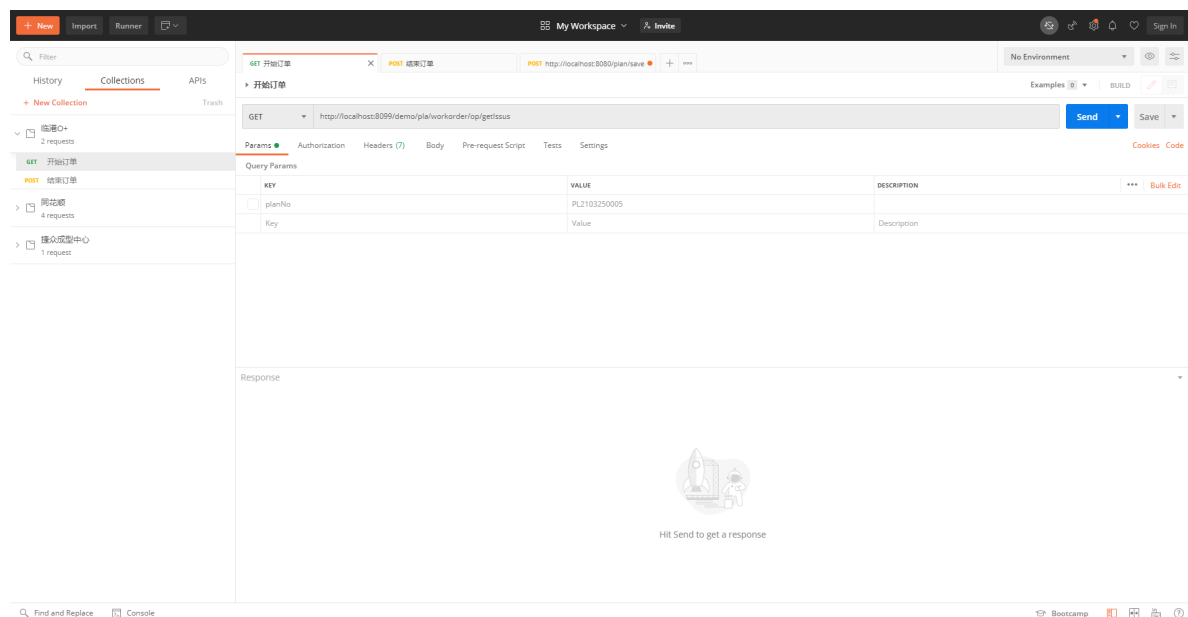
Postman一款非常流行的API调试工具。其实，开发人员用的更多。因为测试人员做接口测试会有更多选择，例如 Jmeter、soapUI 等。不过，对于开发过程中去调试接口，Postman确实足够的简单方便，而且功能强大。

官方网站: <https://www.getpostman.com/>

下载地址: <https://www.postman.com/downloads/>

Postman-win64-8.4.0-Setup.exe

安装与使用: <https://www.cnblogs.com/fnng/p/9136434.html>



4.4 颜色提取工具

colors lite是一款屏幕颜色抓取工具，colors lite可以帮你轻松的抓到各种颜色的值。颜色抓取工具也能够从当前显示的画面内容选择一种颜色，并支持多种不同的颜色模式，其中包括：RGB，HSV，HLS，CMYK。

颜色提取工具.zip