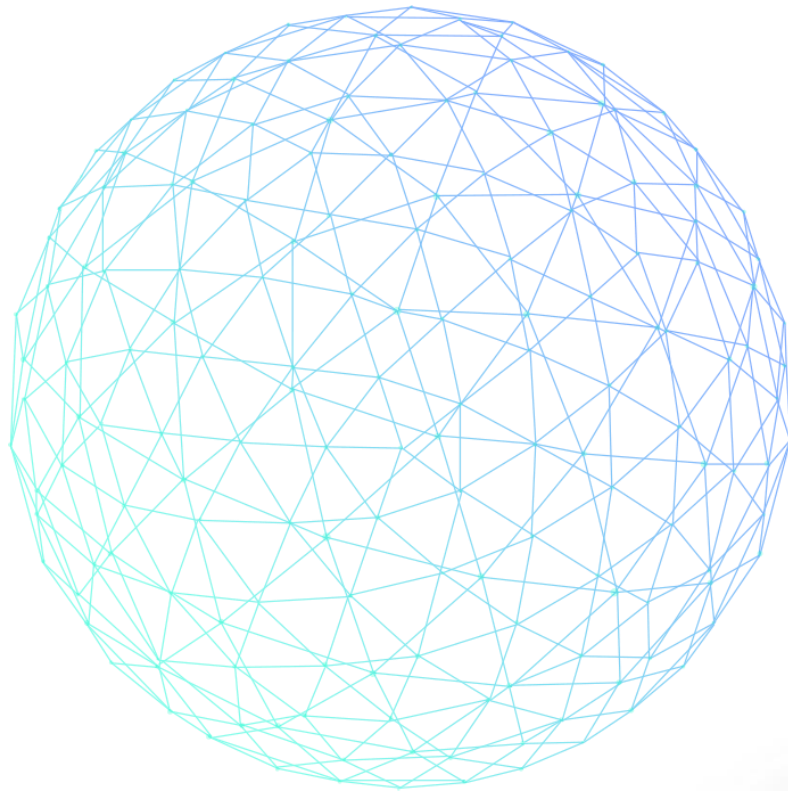


SQLynx

Installation Documentation



【Version : 3.0.0】

Menu

1. LINUX OPERATING SYSTEM	1
1.1 Environmental inspection	1
1.2 Install SQLynx software	2
1.2.1 Download the installation package	2
1.2.2 Decompress software	2
1.2.3 Starting SQLynx	3
1.2.4 Modify configuration	6
1.2.4.1 Modify port number	6
1.2.4.2 Modify JVM heap size	8
2. WINDOWS OPERATING SYSTEM	10
2.1 Environmental inspection	10
2.2 Install SQLynx software	11
2.2.1 Download the installation package	11
2.2.2 Decompress software	11
2.2.3 Start SQLynx	12
2.2.4 Close SQLynx	13
2.2.5 Modify configuration	13
2.2.5.1 Modify port number	13
2.2.5.2 Modify JVM heap size	15

1. Linux operating system

1.1 Environmental inspection

SQLynx provides two installation package versions with built-in JDK and without JDK. You can choose the installation package according to your own needs. Open the terminal and execute the command to check whether the JDK is installed and the installed JDK version. JDK1.8 or above is required.

Open the terminal and execute the command to check the JDK version , which requires JDK1.8 or above :

```
java -version
```

```
@localhost ~ % java -version
openjdk version "20.0.1" 2023-04-18
OpenJDK Runtime Environment (build 20.0.1+9-29)
OpenJDK 64-Bit Server VM (build 20.0.1+9-29, mixed mode, sharing)
```

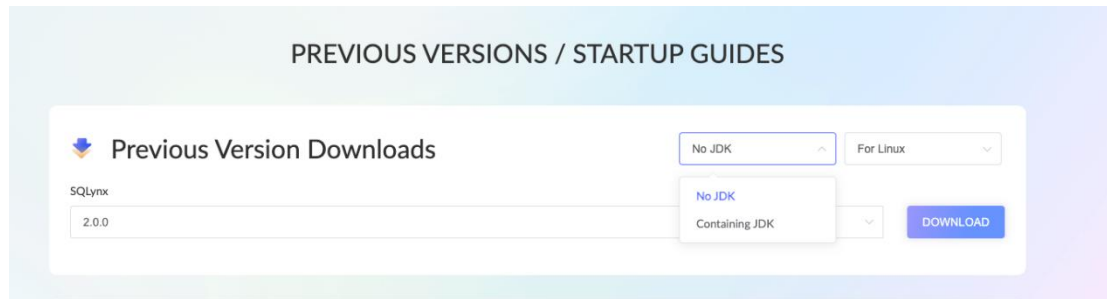
If JDK version 1.8 or above is not installed, you need to download the installation package of the version that comes with the JDK, or install the JDK yourself and then download the installation package of the No-JDK version.

Note: The JDK in the installation package that comes with the JDK only supports AMD64 (x86) architecture. If the server is for other architectures, you need to manually replace the corresponding JDK.

1.2 Install SQLynx software

1.2.1 Download the installation package

Visit the download page <https://www.sqlynx.com>, select the appropriate software version, and click to download.



The following steps take the No-JDK version as an example. The downloaded installation package is named:

sqlynx_enterprise_linux_no_jdk_2.0.0.zip

Note: SQL Studio has now been renamed SQLynx

1.2.2 Decompress software

Unzip the installation package to the current folder, command:

```
unzip <filename>
```

Example:

```
unzip sqlynx_enterprise_linux_no_jdk_2.0.0.zip
```

```

@localhost downloads % unzip sqlynx_enterprise_linux_no_jdk_2.0.0.zip
Archive:  sqlynx_enterprise_linux_no_jdk_2.0.0.zip
  creating: sqlynx/
  inflating: sqlynx/maicong-sqlynx.sh
  inflating: sqlynx/devops-maicong-sqlynx-1.0.0-release-jar-with-dependencies.jar
  creating: sqlynx/config/
  inflating: sqlynx/config/maicong.yaml
  creating: sqlynx/ext/
  inflating: sqlynx/ext/sdtype.jar
  inflating: sqlynx/ext/sdoapi.jar
  inflating: sqlynx/ext/sdoutl.jar
  inflating: sqlynx/ext/sdodep3prt.jar
  creating: sqlynx/dep_lib/
  inflating: sqlynx/dep_lib/jaxb-impl-2.2.3-1.jar
  inflating: sqlynx/dep_lib/druid-1.1.24.jar
  inflating: sqlynx/dep_lib/spring-boot-starter-test-2.6.7.jar
  inflating: sqlynx/dep_lib/jsonassert-1.5.0.jar
  inflating: sqlynx/dep_lib/jakarta.xml.bind-api-2.3.3.jar
  inflating: sqlynx/dep_lib/json-path-2.6.0.jar
  inflating: sqlynx/dep_lib/mongodb-driver-sync-4.9.0.jar
  inflating: sqlynx/dep_lib/spring-context-5.3.19.jar
  inflating: sqlynx/dep_lib/fastjson-1.2.83.jar
  inflating: sqlynx/dep_lib/assertj-core-3.21.0.jar
  inflating: sqlynx/dep_lib/hadoop-auth-2.7.3.jar
  inflating: sqlynx/dep_lib/httpcore-4.4.15.jar
  inflating: sqlynx/dep_lib/slf4j-api-1.7.36.jar
  inflating: sqlynx/dep_lib/junit-jupiter-5.8.2.jar
  inflating: sqlynx/dep_lib/xmlbeans-3.0.1.jar

```

You can also unzip to the specified directory and unzip the installation package to the specified directory with the command

```
unzip <filename> -d <path>
```

If unzip is not installed, you can execute the following command to install it

```
yum install -y unzip zip
```

1.2.3 Starting SQLYnx

1. After decompressing the installation package, a folder named sqlynx will be generated. Execute the command

```
cd sqlynx
```

Enter the file directory

```

@localhost downloads % cd sqlynx
@localhost sqlynx %

```

2、Execute the ls command and you can see that there is a maicong-sqlynx.sh file in the directory.

```
@localhost sqlynx % ls
Maicong-SQLynx-2.0.0-release.jar      devops-maicong-sqlynx-1.0.0-release-jar-with-dependencies.jar
README_cn.md                         devops-maicong-sqlynx.sh
README_en.md                         ext
config                               lib
data                                  maicong-sqlynx.sh
dep_lib                              static
```

3. Execute the command:

```
./maicong-sql ynx .sh
```

The following prompt content is displayed:

```
[ @localhost sqlynx % ./maicong-sqlynx.sh
*****
**
**      maicong-sqlynx  commands      **
**
*****
**      sh maicong-sqlynx.sh start    **
**      sh maicong-sqlynx.sh stop    **
**      sh maicong-sqlynx.sh restart **
*****
```

The three instructions displayed are:

```
sh maicong-sqlynx.sh start
```

Start service

```
sh maicong-sqlynx.sh stop
```

Out of service

```
sh maicong-sqlynx.sh restart
```

Restart service

4. Execute the command sh maicong-sqlynx.sh start to start the service

```

@localhost sqlynx % sh maicong-sqlynx.sh start

maicong does

-e maicong-sqlynx server is started
JAVA_OPTS:
-server
-Xms256m
-Xmx4g
-XX:+UseG1GC
-XX:+UseStringDeduplication
-Xloggc:./log/maicong-sqlynx-gc.log
-XX:+HeapDumpOnOutOfMemoryError
-XX:HeapDumpPath=./log/maicong-sqlynx-heapdump
-Dfile.encoding=utf-8

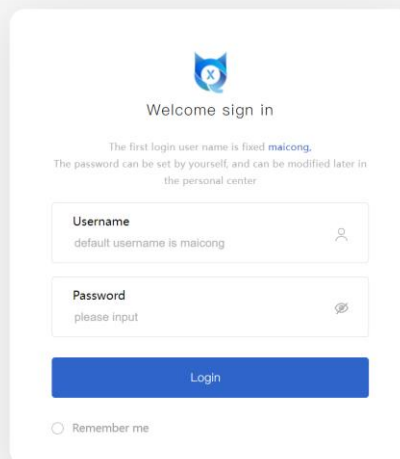
-e please waiting start

-e maicong-sqlynx server start complete

```

5. After startup, you can use a browser to log in to the SQLynx web page: <http://<server ip address>:18888>, 18888 is the default port after SQLynx installation.

The following login page appears, indicating that SQLynx has been installed successfully.



Initial username : maicong password: setting by user's input.

1.2.4 Modify configuration

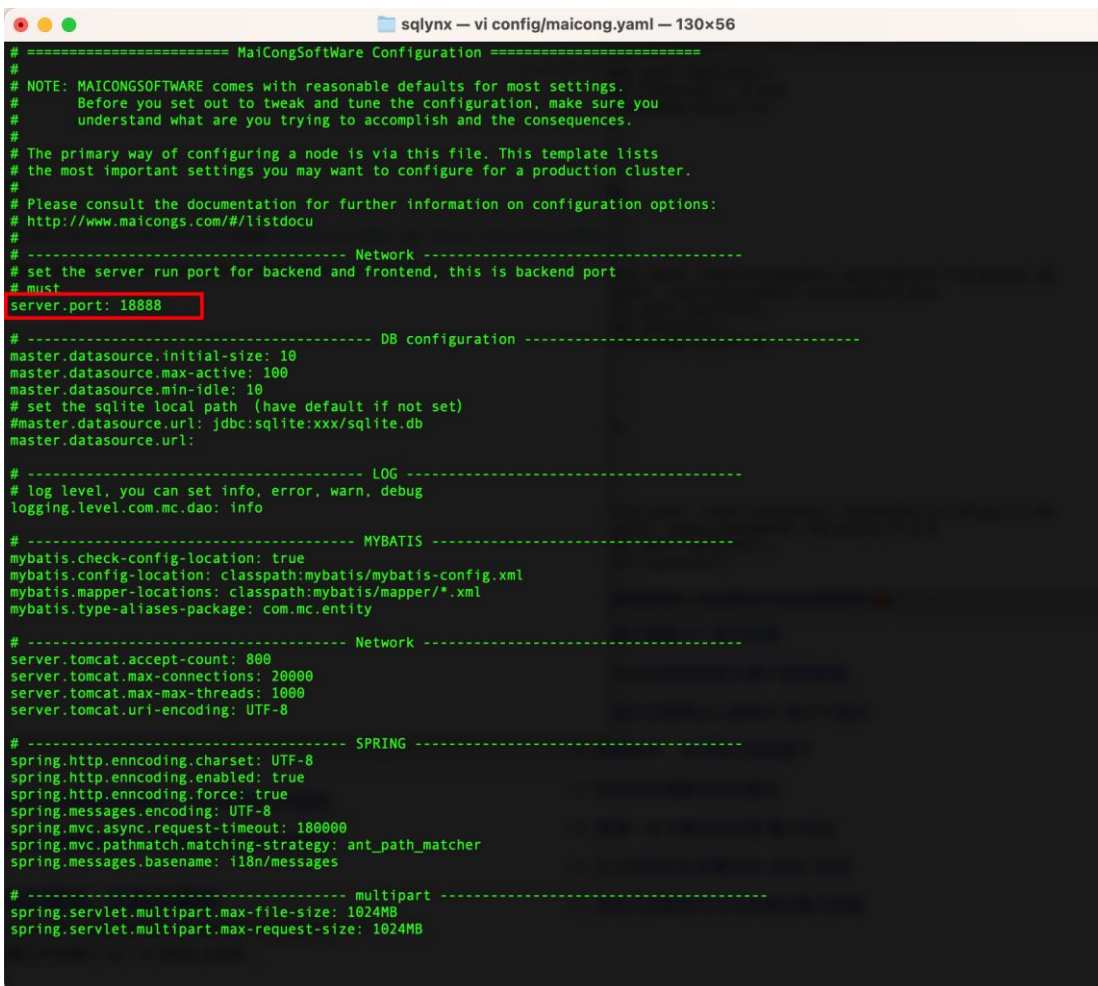
1.2.4.1 Modify port number

Enter the sqlynx directory and update the configuration file according to the example.

1. Execute the command `vi config/maicong.yaml`

```
@localhost sqlynx % vi config/maicong.yaml
```

The following content is displayed:



```

===== MaicongSoftware Configuration =====
#
# NOTE: MAICONGSOFTWARE comes with reasonable defaults for most settings.
#       Before you set out to tweak and tune the configuration, make sure you
#       understand what are you trying to accomplish and the consequences.
#
# The primary way of configuring a node is via this file. This template lists
# the most important settings you may want to configure for a production cluster.
#
# Please consult the documentation for further information on configuration options:
# http://www.maicong.com/#/listdocu
#
# ----- Network -----
# set the server run port for backend and frontend, this is backend port
# must
server.port: 18888

# ----- DB configuration -----
master.datasource.initial-size: 10
master.datasource.max-active: 100
master.datasource.min-idle: 10
# set the sqLite local path (have default if not set)
#master.datasource.url: jdbc:sqlite:xxx/sqlite.db
master.datasource.url:

# ----- LOG -----
# log level, you can set info, error, warn, debug
logging.level.com.mc.dao: info

# ----- MYBATIS -----
mybatis.check-config-location: true
mybatis.config-location: classpath:mybatis/mybatis-config.xml
mybatis.mapper-locations: classpath:mybatis/mapper/*.xml
mybatis.type-aliases-package: com.mc.entity

# ----- Network -----
server.tomcat.accept-count: 800
server.tomcat.max-connections: 20000
server.tomcat.max-max-threads: 1000
server.tomcat.uri-encoding: UTF-8

# ----- SPRING -----
spring.http.encoding.charset: UTF-8
spring.http.encoding.enabled: true
spring.http.encoding.force: true
spring.messages.encoding: UTF-8
spring.mvc.async.request-timeout: 180000
spring.mvc.pathmatch.matching-strategy: ant_path_matcher
spring.messages.basename: i18n/messages

# ----- multipart -----
spring.servlet.multipart.max-file-size: 1024MB
spring.servlet.multipart.max-request-size: 1024MB

```

`server.port` represents the port number

2. Press the `i` key to enter editing mode and modify the port number.


```

sqlynx — vi config/maicong.yaml — 130x56
# ===== Ma1CongSoftWare Configuration =====
#
# NOTE: MAICONGSOFTWARE comes with reasonable defaults for most settings.
#       Before you set out to tweak and tune the configuration, make sure you
#       understand what are you trying to accomplish and the consequences.
#
# The primary way of configuring a node is via this file. This template lists
# the most important settings you may want to configure for a production cluster.
#
# Please consult the documentation for further information on configuration options:
# http://www.maicongs.com/#/listdocu
#
# ----- Network -----
# set the server run port for backend and frontend, this is backend port
# must
server.port: 18888

# ----- DB configuration -----
master.datasource.initial-size: 10
master.datasource.max-active: 100
master.datasource.min-idle: 10
# set the sqlite local path (have default if not set)
#master.datasource.url: jdbc:sqlite:xxx/sqlite.db
master.datasource.url:

# ----- LOG -----
# log level, you can set info, error, warn, debug
logging.level.com.mc.dao: info

# ----- MYBATIS -----
mybatis.check-config-location: true
mybatis.config-location: classpath:mybatis/mybatis-config.xml
mybatis.mapper-locations: classpath:mybatis/mapper/*.xml
mybatis.type-aliases-package: com.mc.entity

# ----- Network -----
server.tomcat.accept-count: 800
server.tomcat.max-connections: 20000
server.tomcat.max-max-threads: 1000
server.tomcat.uri-encoding: UTF-8

# ----- SPRING -----
spring.http.encoding.charset: UTF-8
spring.http.encoding.enabled: true
spring.http.encoding.force: true
spring.messages.encoding: UTF-8
spring.mvc.async.request-timeout: 180000
spring.mvc.pathmatch.matching-strategy: ant_path_matcher
spring.messages.basename: i18n/messages

# ----- multipart -----
spring.servlet.multipart.max-file-size: 1024MB
spring.servlet.multipart.max-request-size: 1024MB

~ INSERT ~

```

3. Press the esc key to exit the editing mode, enter the command :wq to save changes and exit.

```

sqlynx — vi config/maicong.yaml — 130x56
# ----- MaiCongSoftWare Configuration -----
#
# NOTE: MAICONGSOFTWARE comes with reasonable defaults for most settings.
# Before you set out to tweak and tune the configuration, make sure you
# understand what are you trying to accomplish and the consequences.
#
# The primary way of configuring a node is via this file. This template lists
# the most important settings you may want to configure for a production cluster.
#
# Please consult the documentation for further information on configuration options:
# http://www.maicongs.com/#/listdocu
#
# ----- Network -----
# set the server run port for backend and frontend, this is backend port
# must
server.port: 18888

# ----- DB configuration -----
master.datasource.initial-size: 10
master.datasource.max-active: 100
master.datasource.min-idle: 10
# set the sqlite local path (have default if not set)
#master.datasource.url: jdbc:sqlite:xxx/sqlite.db
master.datasource.url:

# ----- LOG -----
# log level, you can set info, error, warn, debug
logging.level.com.mc.dao: info

# ----- MYBATIS -----
mybatis.check-config-location: true
mybatis.config-location: classpath:mybatis/mybatis-config.xml
mybatis.mapper-locations: classpath:mybatis/mapper/*.xml
mybatis.type-aliases-package: com.mc.entity

# ----- Network -----
server.tomcat.accept-count: 800
server.tomcat.max-connections: 20000
server.tomcat.max-max-threads: 1000
server.tomcat.uri-encoding: UTF-8

# ----- SPRING -----
spring.http.encoding.charset: UTF-8
spring.http.encoding.enabled: true
spring.http.encoding.force: true
spring.messages.encoding: UTF-8
spring.mvc.async.request-timeout: 180000
spring.mvc.pathmatch.matching-strategy: ant_path_matcher
spring.messages.basename: i18n/messages

# ----- multipart -----
spring.servlet.multipart.max-file-size: 1024MB
spring.servlet.multipart.max-request-size: 1024MB

:wq

```

1.2.4.2 Modify JVM heap size

Enter the sqlynx directory and update the configuration file according to the example.

1、Excuting an order

```
vi maicong-sqlynx.sh
```

```
@localhost sqlynx % vi maicong-sqlynx.sh
```

The following content is displayed:

2. Windows operating system

2.1 Environmental inspection

SQLynx provides two installation package versions with built-in JDK and without JDK. You can choose the installation package according to your own needs. Open the terminal and execute the command to check whether the JDK is installed and the installed JDK version. JDK1.8 or above is required.

Press win + r , enter cmd to open the terminal and execute the command to check the JDK version . JDK1.8 or above is required :

```
java -version
```

```
C:\Users\86152>java -version
java version "1.8.0_102"
Java(TM) SE Runtime Environment (build 1.8.0_102-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.102-b14, mixed mode)
```

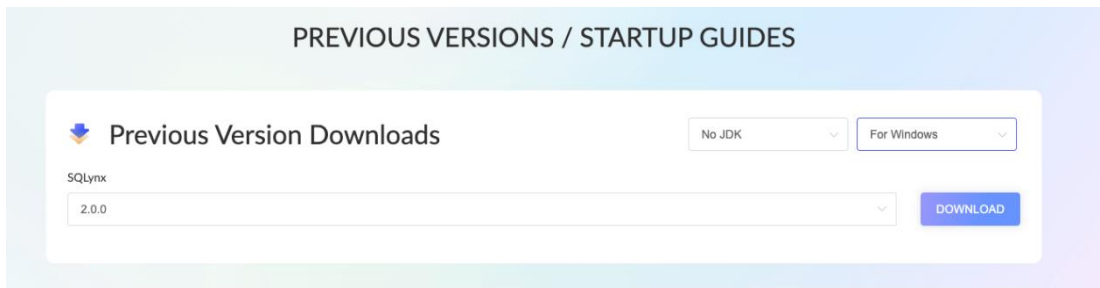
If JDK version 1.8 or above is not installed, you need to download the installation package of the version that comes with the JDK, or install the JDK yourself and then download the installation package of the No-JDK version.

Note: The JDK in the installation package that comes with the JDK only supports AMD64 (x86) architecture. If the server is for other architectures, you need to manually replace the corresponding JDK.

2.2 Install SQLynx software

2.2.1 Download the installation package

Visit the download page <https://www.sqlynx.com>, select the appropriate software version, and click to download.



The following steps take the No JDK version as an example. The downloaded installation package is named:

sqlynx_enterprise_win_no_jdk_2.0.0.zip

Note: SQL Studio has now been renamed SQLynx

2.2.2 Decompress software

-Right-click the installation package and extract the installation package to the current folder.

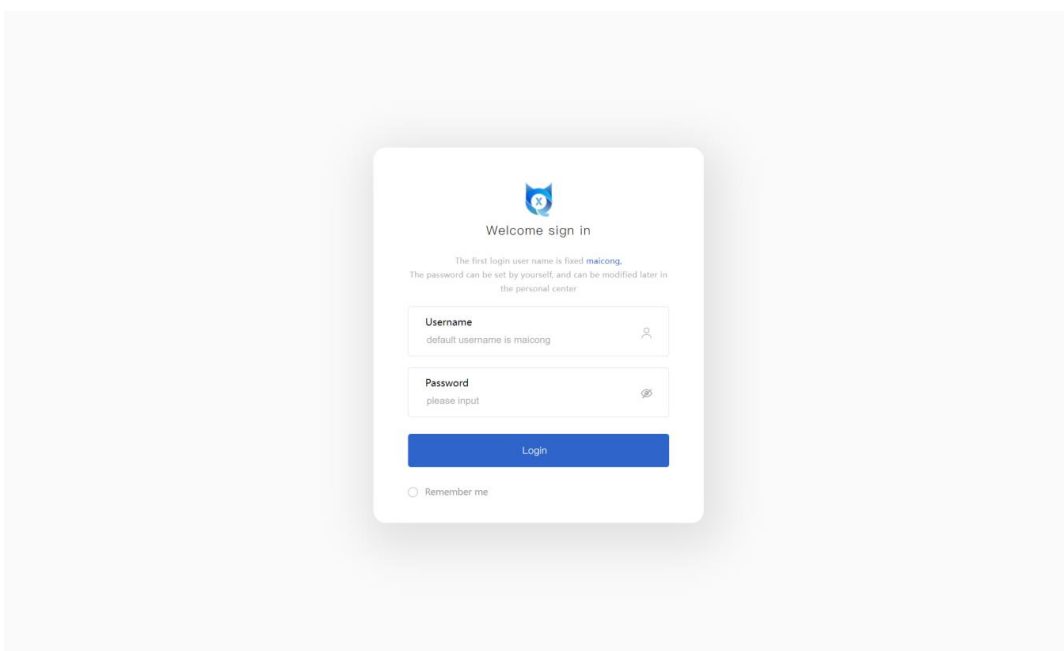
-You can also unzip it to a specified directory.

2.2.3 Start SQLynx

1. After decompressing the installation package, a folder named sqlynx will be generated and enter the file directory.
2. After entering , you can see that there is a maicong -sqlynx_startup.bat file in the directory , double-click to run

config	2023/9/13 14:31
data	2023/9/13 14:35
dep_lib	2023/9/13 14:31
ext	2023/9/13 14:31
lib	2023/9/13 14:31
log	2023/9/13 15:23
static	2023/9/13 14:31
devops-maicong-sqlynx.bat	2023/9/13 16:44
devops-maicong-sqlynx-1.0.0-rele...	2023/9/11 20:56
maicong-sqlynx_shutdown.bat	2023/9/11 20:15
maicong-sqlynx_startup.bat	2023/9/11 20:15
Maicong-SQLynx-2.0.0-release.jar	2023/9/11 19:55
README_cn.md	2023/9/11 20:35
README_en.md	2023/9/11 20:35

3. After double-clicking the maicong -sqlynx_startup.bat file , the cmd command window will pop up, as shown SQLynx start Done.
4. The following login page appears, indicating that SQLynx has been successfully installed.



Initial username : maicong password: setting by user's input.

2.2.4 Close SQLYnx

1. Enter the sqlynx folder and double-click the maicong-sqlynx_shutdown.bat file

config	2023/9/13 14:31
data	2023/9/13 14:35
dep_lib	2023/9/13 14:31
ext	2023/9/13 14:31
lib	2023/9/13 14:31
log	2023/9/13 15:23
static	2023/9/13 14:31
devops-maicong-sqlynx.bat	2023/9/13 16:44
devops-maicong-sqlynx-1.0.0-rele...	2023/9/11 20:56
maicong-sqlynx_shutdown.bat	2023/9/11 20:15
maicong-sqlynx_startup.bat	2023/9/11 20:15
Maicong-SQLYnx-2.0.0-release.jar	2023/9/11 19:55

2. The cmd command window will pop up, as shown kill Maicong-SQLYnx Done, indicating that the server has been shut down.

2.2.5 Modify configuration

2.2.5.1 Modify port number

Enter the sqlynx directory and update the configuration file according to the example.

1. Enter the config directory

config	2023/9/13 14:31
data	2023/9/13 14:35
dep_lib	2023/9/13 14:31
ext	2023/9/13 14:31
lib	2023/9/13 14:31
log	2023/9/13 15:23
static	2023/9/13 14:31
devops-maicong-sqlynx.bat	2023/9/13 16:44
devops-maicong-sqlynx-1.0.0-rele...	2023/9/11 20:56
maicong-sqlynx_shutdown.bat	2023/9/11 20:15
maicong-sqlynx_startup.bat	2023/9/11 20:15
Maicong-SQLYnx-2.0.0-release.jar	2023/9/11 19:55

Right-click maicong.yaml and open it with Notepad

The following content is displayed:

```
# ===== MaiCongSoftWare Configuration
# =====
#
# NOTE: MAICONGSOFTWARE comes with reasonable defaults for most settings.
#   Before you set out to tweak and tune the configuration, make sure you
#   understand what are you trying to accomplish and the consequences.
#
# The primary way of configuring a node is via this file. This template lists
# the most important settings you may want to configure for a production cluster.
#
# Please consult the documentation for further information on configuration options:
# http://www.maicongs.com/#/listdocu
#
# ----- Network -----
# set the server run port for backend and frontend, this is backend port
# must
server.port: 18888

# ----- DB configuration -----
master.datasource.initial-size: 10
master.datasource.max-active: 100
master.datasource.min-idle: 10
# set the sqlite local path (have default if not set)
#master.datasource.url: jdbc:sqlite:xxx/sqlite.db
master.datasource.url:

# ----- LOG -----
# log level, you can set info, error, warn, debug
logging.level.com.mc.dao: info

# ----- MYBATIS -----
```

server.port represents the port number

2、 Edit notepad and modify the port number **(note: press ctrl+s to save)**

```
# ===== MaiCongSoftWare Configuration
# =====
#
# NOTE: MAICONGSOFTWARE comes with reasonable defaults for most settings.
#   Before you set out to tweak and tune the configuration, make sure you
#   understand what are you trying to accomplish and the consequences.
#
# The primary way of configuring a node is via this file. This template lists
# the most important settings you may want to configure for a production cluster.
#
# Please consult the documentation for further information on configuration options:
# http://www.maicongs.com/#/listdocu
#
# ----- Network -----
# set the server run port for backend and frontend, this is backend port
# must
server.port: 18888

# ----- DB configuration -----
master.datasource.initial-size: 10
master.datasource.max-active: 100
master.datasource.min-idle: 10
# set the sqlite local path (have default if not set)
#master.datasource.url: jdbc:sqlite:xxx/sqlite.db
master.datasource.url:

# ----- LOG -----
# log level, you can set info, error, warn, debug
logging.level.com.mc.dao: info

# ----- MYBATIS -----
```


2.2.5.2 Modify JVM heap size

Enter the sqlynx directory and update the configuration file according to the example.

1. Right-click maicong-sqlynx_startup.bat and select Edit

The following content is displayed:

```
@echo off
SETLOCAL ENABLEDELAYEDEXPANSION

cd /d %~dp0

set SERVER_HOST=http://localhost
set SERVER_PORT=18888

set APP_NAME=Maicong-SQLynx

for /f "delims=" %%t in ('dir /B ^| find "%APP_NAME%") do set APP_JAR=%%t

for /f "tokens=1-2" %%a in ('jps -l ^| find "%APP_NAME%") do (
    echo start kill pid %%a %%b
    taskkill /F /PID %%a
)

echo sqlynx is loading, please wait...

set JAVA_OPTS= ^
-server ^
-Xms128m ^
-Xmx1024m ^
-XX:+UseG1GC ^
-XX:+UseStringDeduplication ^
-Xloggc:log\maicong-sqlynx-gc.log ^
-XX:+HeapDumpOnOutOfMemoryError ^
-XX:HeapDumpPath=log\maicong-sqlynx-heapdump ^
-Dfile.encoding=utf-8

start javaw %JAVA_OPTS% -jar %APP_JAR% --spring.config.location=config\maicong.yml
```

-Xms JVM initial allocated heap memory

-Xmx The maximum heap memory allowed to be allocated by the JVM

The heap size can be modified according to the actual situation of the server