# RVA开发环境搭建指南

蔡磊

2022-10-11

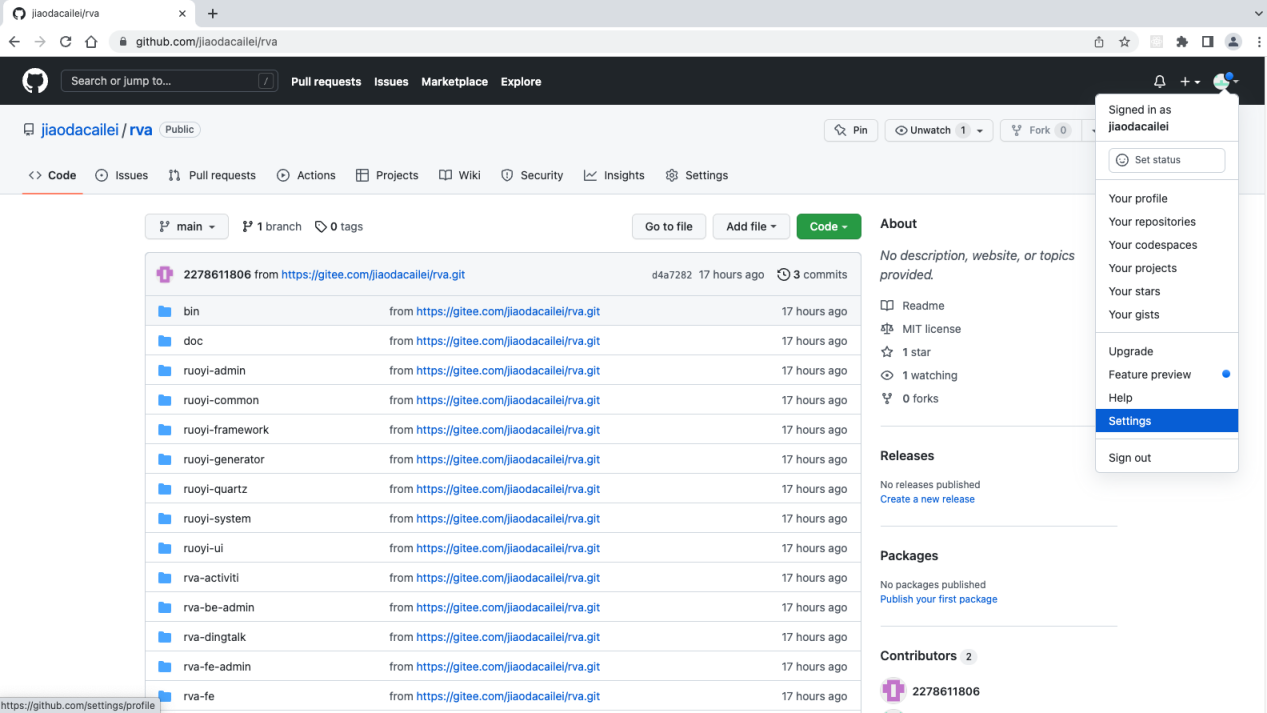
# 从github上下载项目

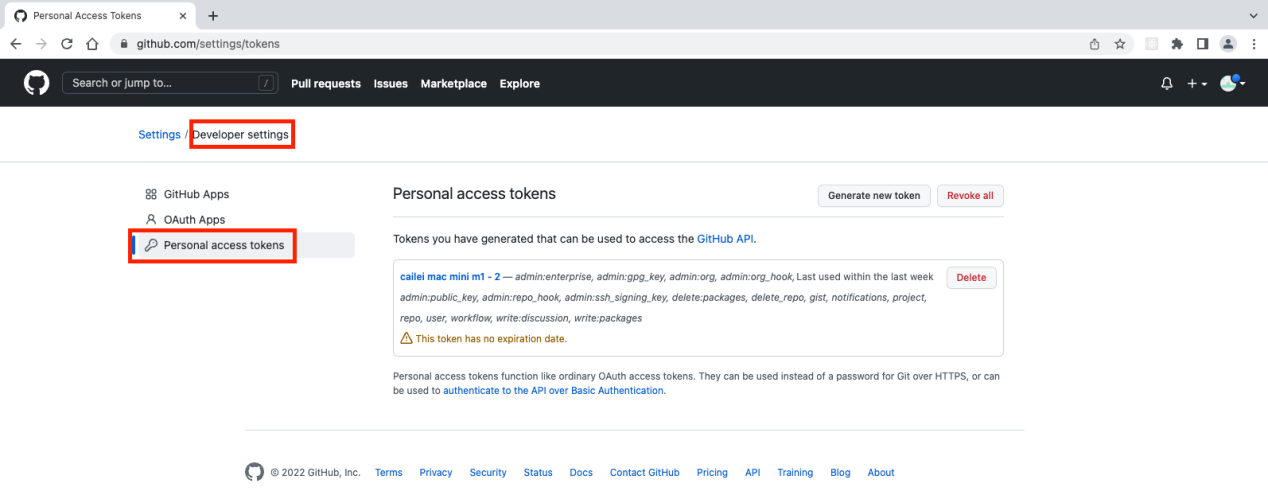
github地址：

<https://github.com/jiaodacailei/rva.git>

目前，该项目为public，无需任何其他设置；

如果项目是private（rva的应用项目，很多需要设为private），需要设置Personal access tokens，该token将作为用户的登录密码





# 安装ellipse

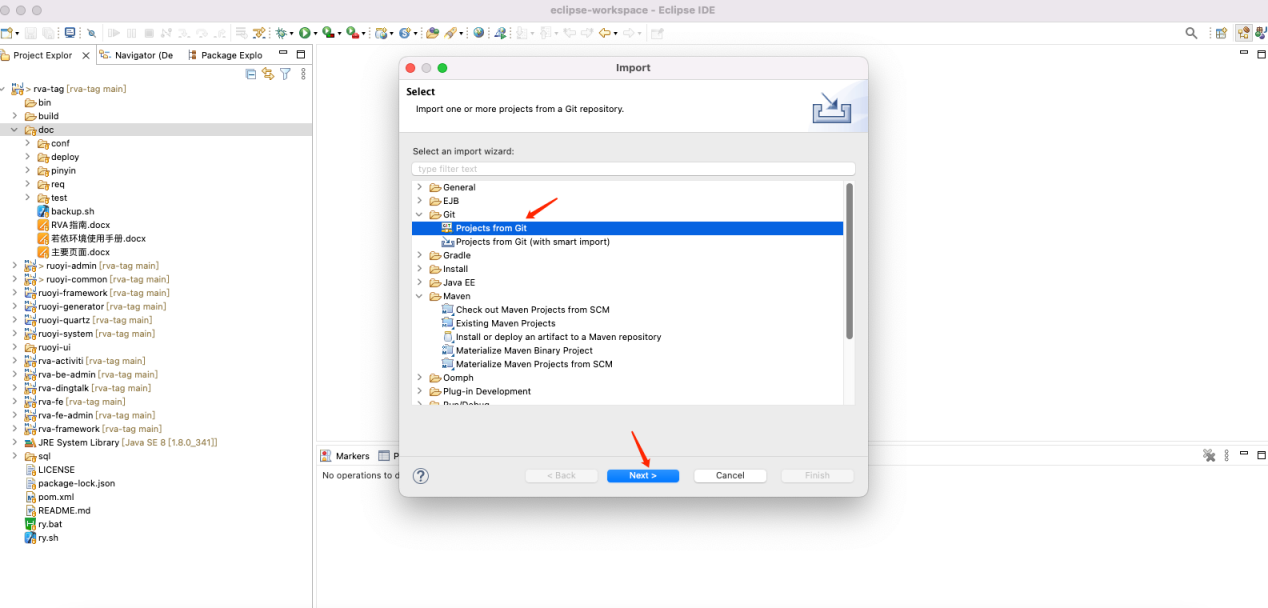
官网安装最新版，会安装jdk17：



还要安装Lombok哟！

# 导入项目

打开eclipse后，选择菜单：file - import，选择git：



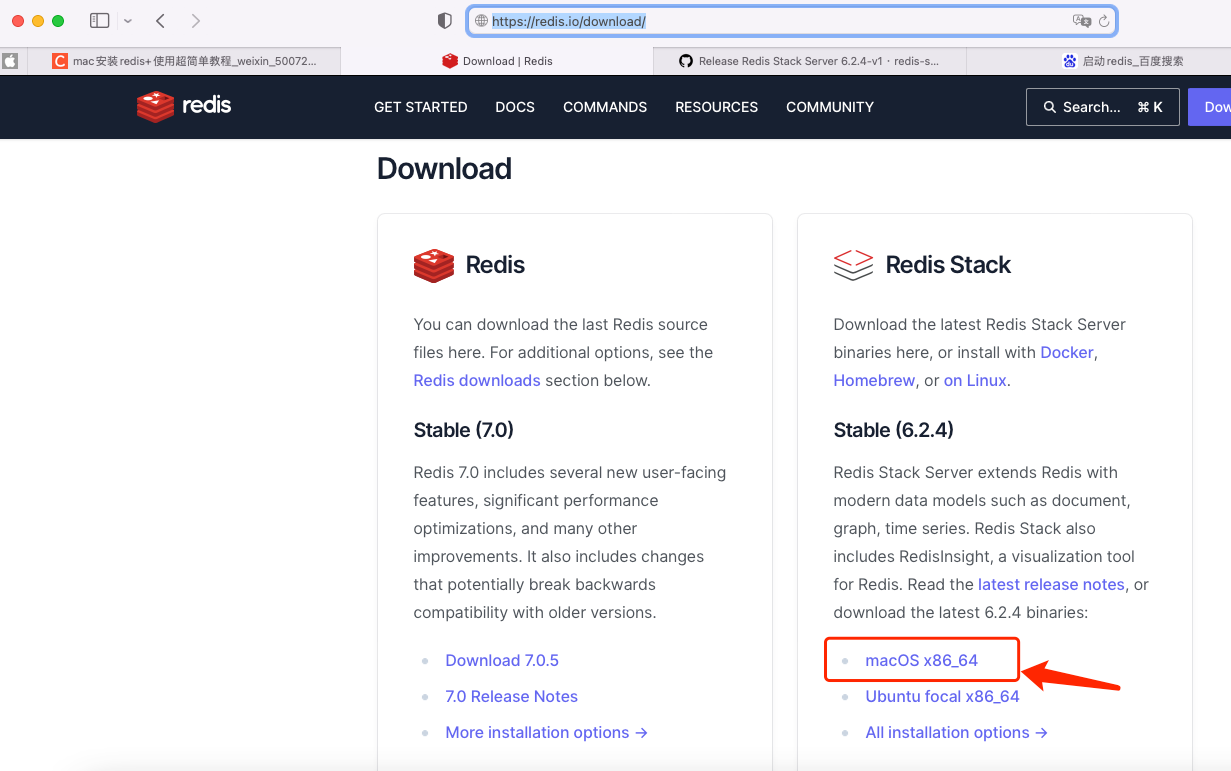
注意：导入项目后，仅仅只是将整个父项目导入了，其子项目还没有，所以需要选中父项目，接着导入所有其下的子项目

# 安装Redis

redis安装没有版本要求

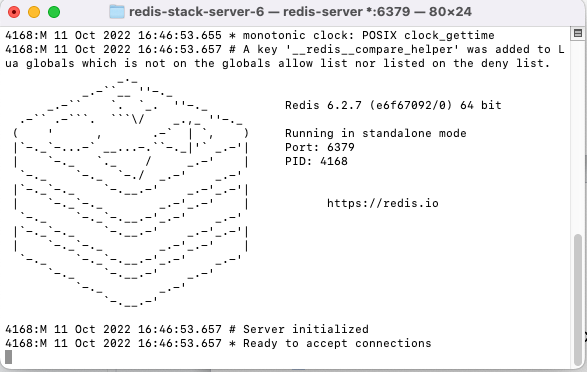
## 安装MAC版

redis官网直接下载二进制Mac版本，也就是已经make好的版本。



启动服务器：

./Downloads/redis-stack-server-6/bin/redis-server



启动客户端进行测试：

./Downloads/redis-stack-server-6/bin/redis-cli



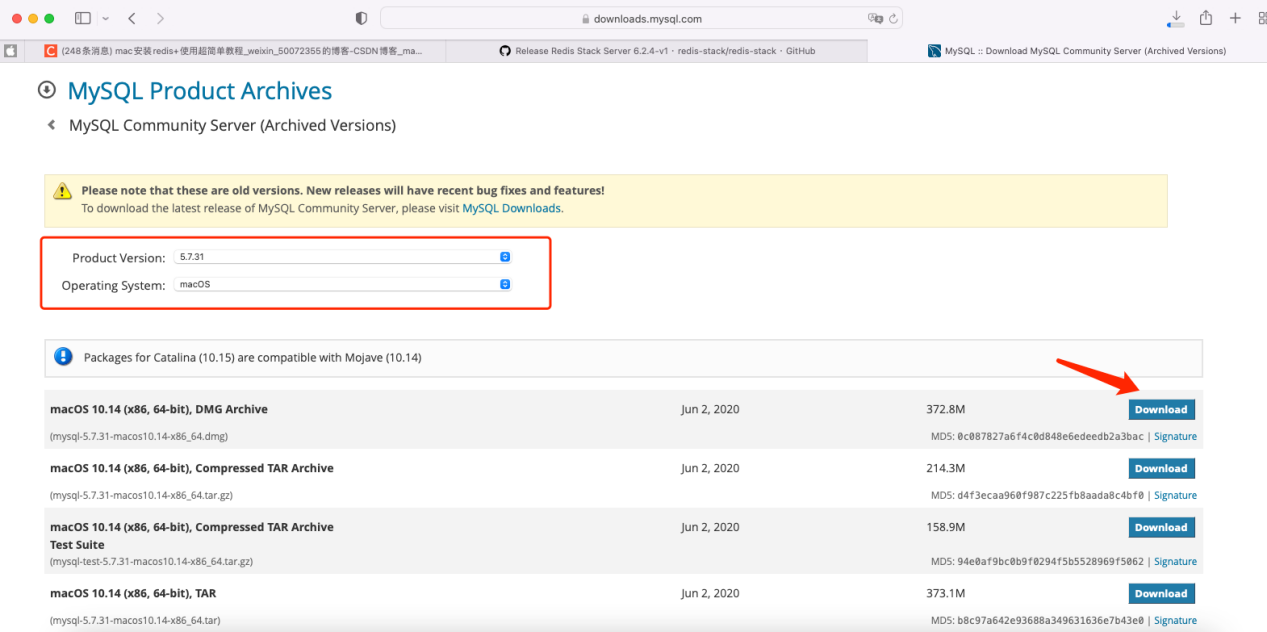
# 安装MySQL5.7

mysql采用5.7版本

## 安装MAC版

### 下载

**<https://downloads.mysql.com/archives/community/>**



5.7.31是最新的一个支持Mac的5.7版

### 安装

点击安装包直接安装即可。

**注意：**第一次安装时，系统会提示无法验证开发者而阻止安装，在安全性与隐私隐私中，允许即可。

参考：<https://www.sohu.com/a/529440723_120371013>

**注意：**安装过程中会生成root用户密码，一定要注意保存，例如：spas&8t-thtP

### **创建mysql配置文件**

sudo vi /etc/my.cnf

并加入如下内容：

|  |
| --- |
| # Example MySQL config file for small systems.  #  # This is for a system with little memory (<= 64M) where MySQL is only used  # from time to time and it\'s important that the mysqld daemon  # doesn\'t use much resources.  #  # MySQL programs look for option files in a set of  # locations which depend on the deployment platform.  # You can copy this option file to one of those  # locations. For information about these locations, see:  # http://dev.mysql.com/doc/mysql/en/option-files.html  #  # In this file, you can use all long options that a program supports.  # If you want to know which options a program supports, run the program  # with the "--help" option.    # The following options will be passed to all MySQL clients  [client]  default-character-set=utf8  #password = your\_password  port = 3306  socket = /tmp/mysql.sock    # Here follows entries for some specific programs    # The MySQL server  [mysqld]  default-storage-engine=INNODB  character-set-server=utf8  collation-server=utf8\_general\_ci  port = 3306  socket = /tmp/mysql.sock  skip-external-locking  key\_buffer\_size = 16K  max\_allowed\_packet = 1M  table\_open\_cache = 4  sort\_buffer\_size = 64K  read\_buffer\_size = 256K  read\_rnd\_buffer\_size = 256K  net\_buffer\_length = 2K  thread\_stack = 128K    # Don\'t listen on a TCP/IP port at all. This can be a security enhancement,  # if all processes that need to connect to mysqld run on the same host.  # All interaction with mysqld must be made via Unix sockets or named pipes.  # Note that using this option without enabling named pipes on Windows  # (using the "enable-named-pipe" option) will render mysqld useless!  #  #skip-networking  server-id = 1    # Uncomment the following if you want to log updates  #log-bin=mysql-bin    # binary logging format - mixed recommended  #binlog\_format=mixed    # Causes updates to non-transactional engines using statement format to be  # written directly to binary log. Before using this option make sure that  # there are no dependencies between transactional and non-transactional  # tables such as in the statement INSERT INTO t\_myisam SELECT \* FROM  # t\_innodb; otherwise, slaves may diverge from the master.  #binlog\_direct\_non\_transactional\_updates=TRUE    # Uncomment the following if you are using InnoDB tables  #innodb\_data\_home\_dir = /usr/local/mysql/data  #innodb\_data\_file\_path = ibdata1:10M:autoextend  #innodb\_log\_group\_home\_dir = /usr/local/mysql/data  # You can set ..\_buffer\_pool\_size up to 50 - 80 %  # of RAM but beware of setting memory usage too high  #innodb\_buffer\_pool\_size = 16M  #innodb\_additional\_mem\_pool\_size = 2M  # Set ..\_log\_file\_size to 25 % of buffer pool size  #innodb\_log\_file\_size = 5M  #innodb\_log\_buffer\_size = 8M  #innodb\_flush\_log\_at\_trx\_commit = 1  #innodb\_lock\_wait\_timeout = 50    [mysqldump]  quick  max\_allowed\_packet = 16M    [mysql]  no-auto-rehash  # Remove the next comment character if you are not familiar with SQL  #safe-updates    [myisamchk]  key\_buffer\_size = 8M  sort\_buffer\_size = 8M    [mysqlhotcopy]  interactive-timeout |

### **启动服务器**

sudo /usr/local/mysql/bin/mysqld --defaults-file=/etc/my.cnf --user=root &

### **修改密码**

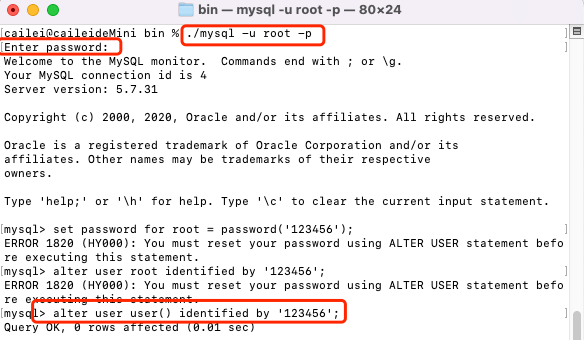
登录mysql客户端：

/usr/local/mysql/bin/mysql -u root -p

然后输入root密码，即可登录成功

输入如下命令，更改用户密码：

alter user user() identified by '123456';



### **创建数据库并导入sql**

创建数据库：

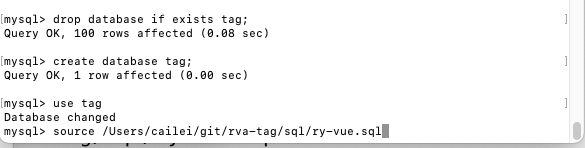
create database tag;

进入数据库：

use tag

导入sql：

source /Users/cailei/git/rva-tag/sql/ry-vue.sql



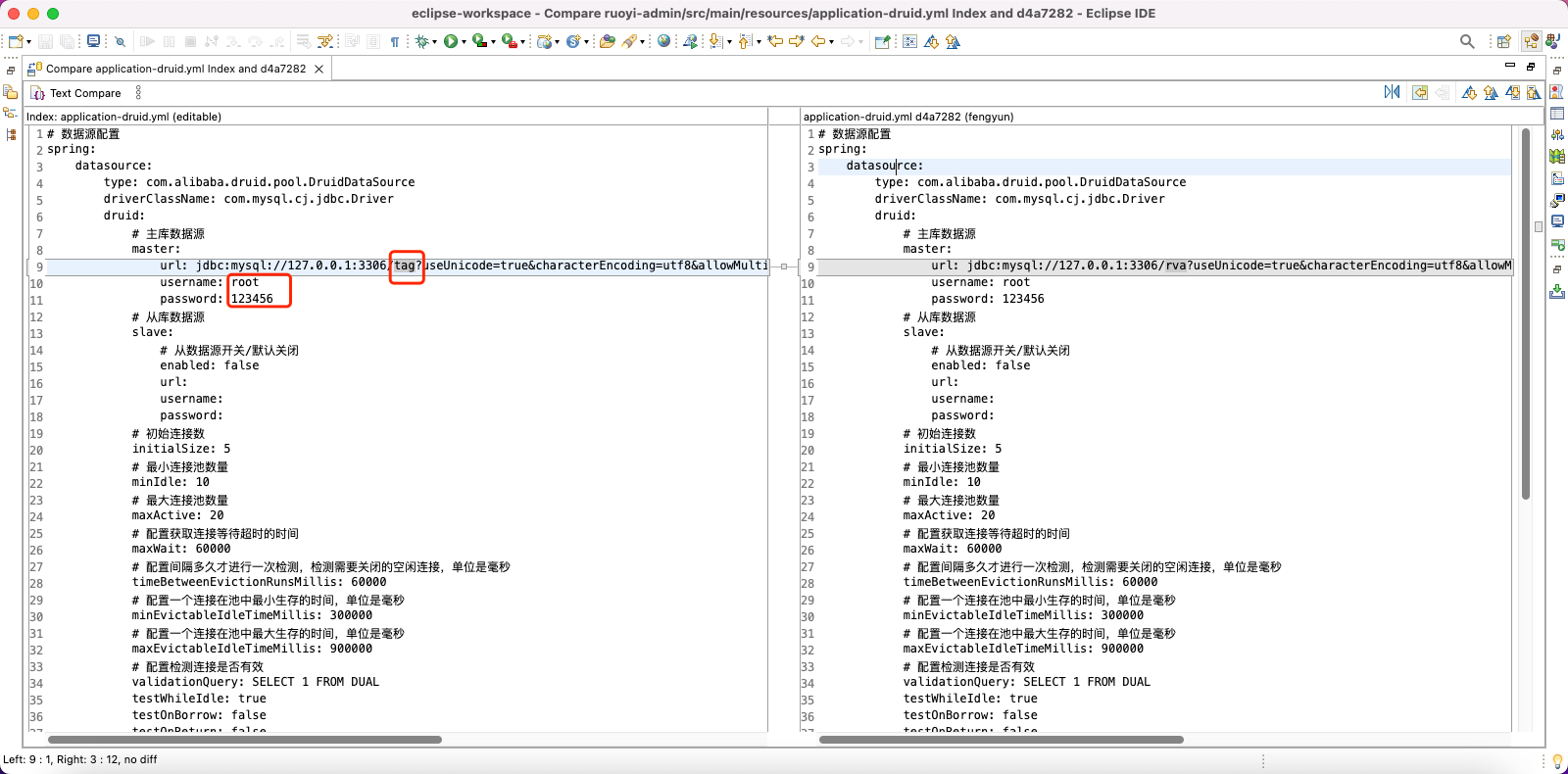
### **导出数据库**

sudo /usr/local/mysql/bin/mysqldump -u root -p123456 tag > /Users/cailei/git/rva-tag/sql/ry-vue.sql

# 启动RVA后台

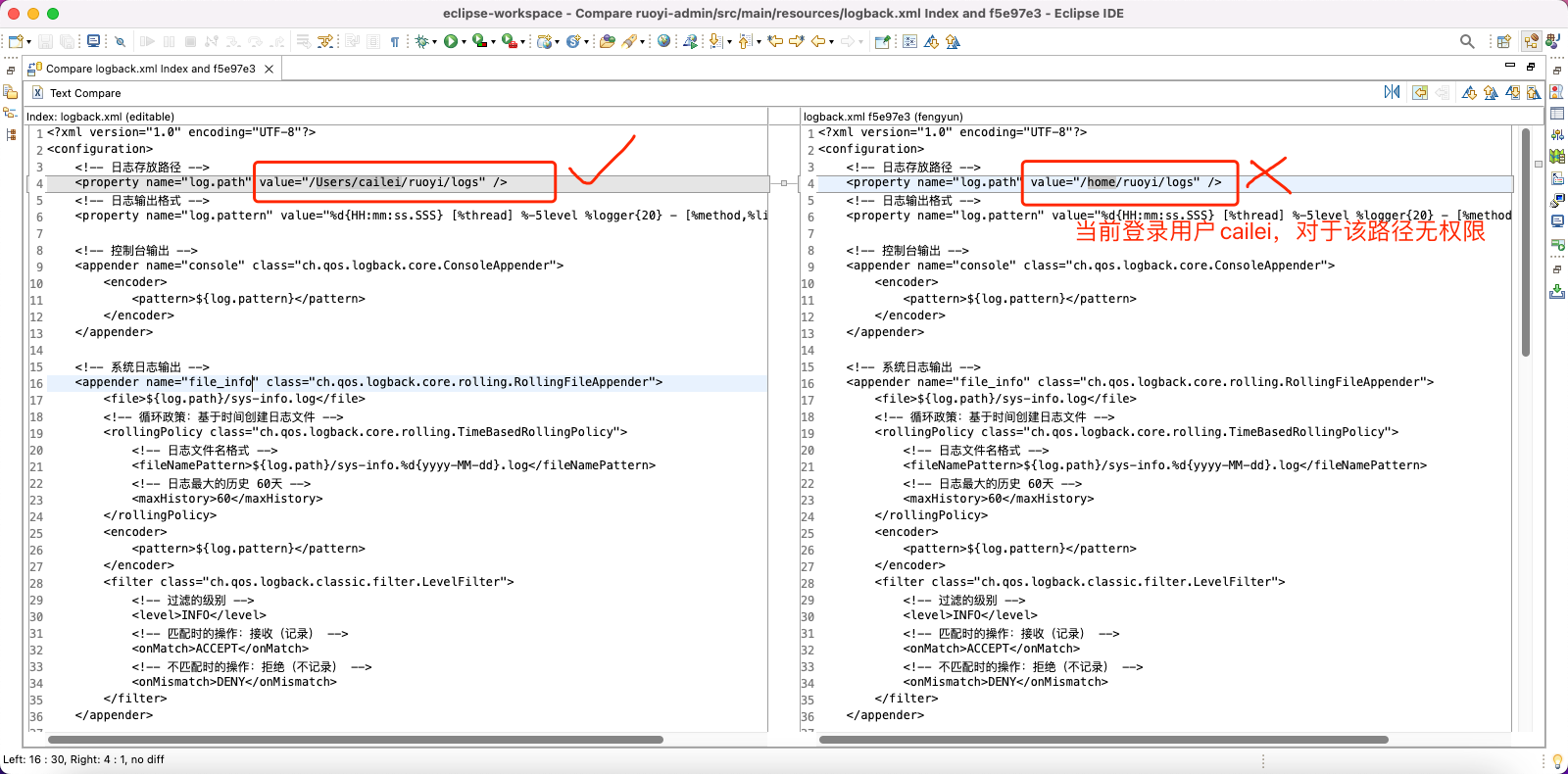
修改数据库连接参数：

ruoyi-admin/src/main/resources/application-druid.yml



Mac或者Linux系统下，需要修改日志文件位置，将其放在当前用户有权限的目录下：

ruoyi-admin/src/main/resources/logback.xml



# 启动RVA前台

## 构建和运行

命令行切换到ruoyi-ui目录

构建执行：

npm install --legacy-peer-deps

运行执行：

npm run dev

自动启动浏览器后，用户名和密码：

admin/admin123

## 修改路径中的rva-admin

**该步骤不是必须的**，如需调整路径中的rva-admin为其他名称，才需要执行本步骤。

修改如下文件，替换rva-admin，并重新构建和运行前台：

ruoyi-ui/.env.development（1）

ruoyi-ui/.env.production（1）

ruoyi-ui/.env.staging（1）

ruoyi-ui/src/router/index.js（1）

doc/conf/nginx.conf（1）

ruoyi-ui/vue.config.js（**2**）

括号中的数字表示rva-admin出现的次数，替换时需要特别注意。