

屌！一套完整牛 X 的后台管理系统（已开源），接外包的同学靠它赚了不少 Money~

点击关注  [Java基基](#) Yesterday

点击上方“[Java基基](#)”，选择“设为星标”

做积极的人，而不是积极废人！

源码精品专栏

- [原创 | Java 2020 超神之路，很肝~](#)
- [中文详细注释的开源项目](#)
- [RPC 框架 Dubbo 源码解析](#)
- [网络应用框架 Netty 源码解析](#)
- [消息中间件 RocketMQ 源码解析](#)
- [数据库中间件 Sharding-JDBC 和 MyCAT 源码解析](#)
- [作业调度中间件 Elastic-Job 源码解析](#)
- [分布式事务中间件 TCC-Transaction 源码解析](#)
- [Eureka 和 Hystrix 源码解析](#)
- [Java 并发源码](#)

来源：cnblogs.com/huanzi-qch/p/11534203.html

- 前言
 - 技术栈
 - 工程结构说明
 - 运行预览
 - 功能演示
 - 关键点讲解
 - 后记
 - 补充更新
 - 代码开源
-
-



前言

前段时间我们写一个简单的后台模板SpringBoot系列——Security + Layui实现一套权限管理后台模板，今天我们把它完善成一个简单通用的后台管理系统，我把它叫做：Base Admin

这套Base Admin是一套简单通用的后台管理系统，主要功能有：权限管理、菜单管理、用户管理，系统设置、实时日志，实时监控，API加密，以及登录用户修改密码、配置个性菜单等

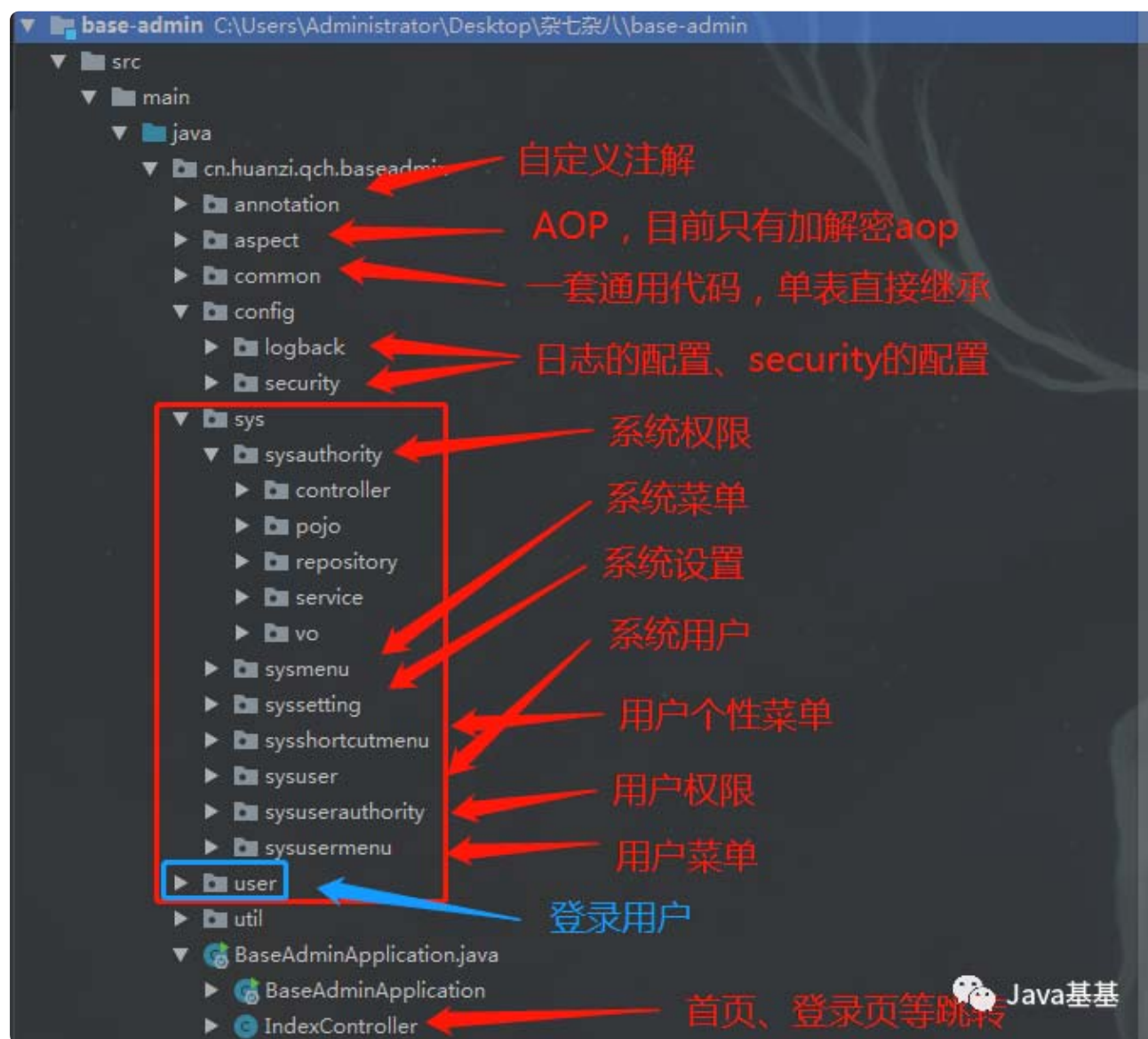
技术栈

前端：layui

java后端：SpringBoot + Thymeleaf + WebSocket + Spring Security + SpringData-Jpa + MySql

工程结构说明

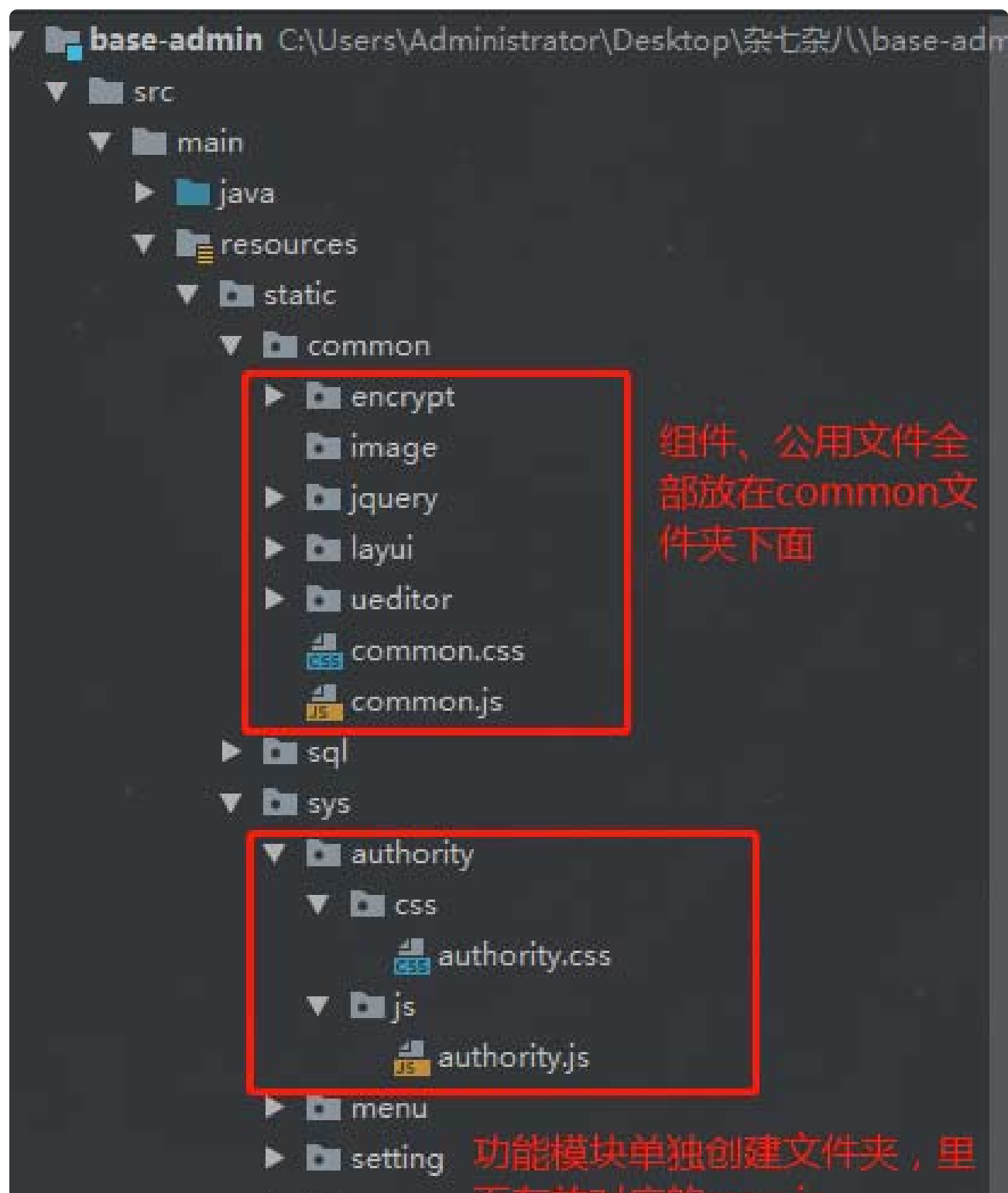
java部分、html、js、css部分都是大目录下面按单表一个子目录存放

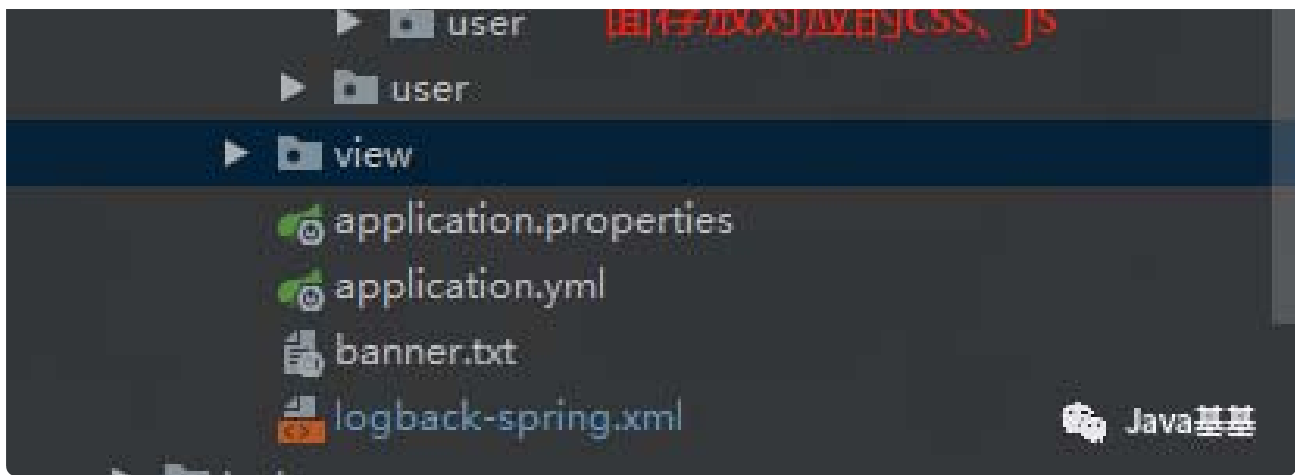




2020-06-10更新

之前调整过前端文件结构布局，现在重新更新说明一下





运行预览

效果先睹为快，具体介绍在下方，按功能点进行详情介绍

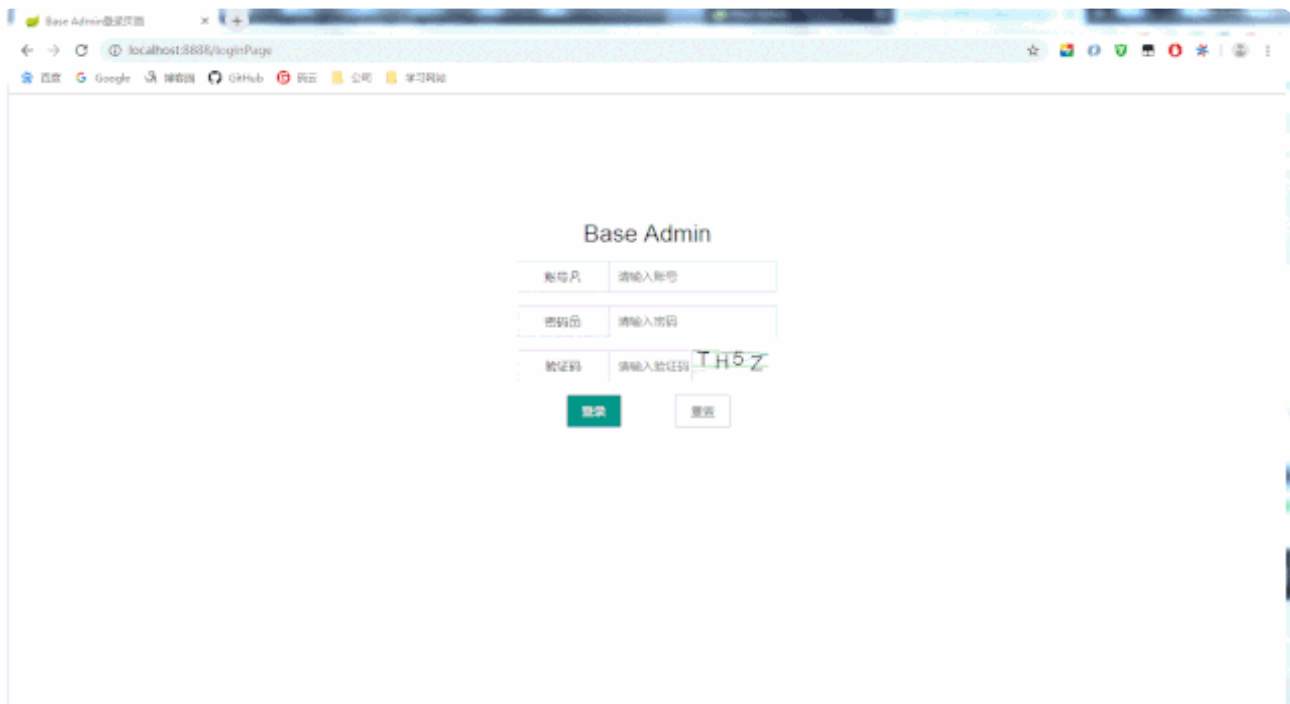
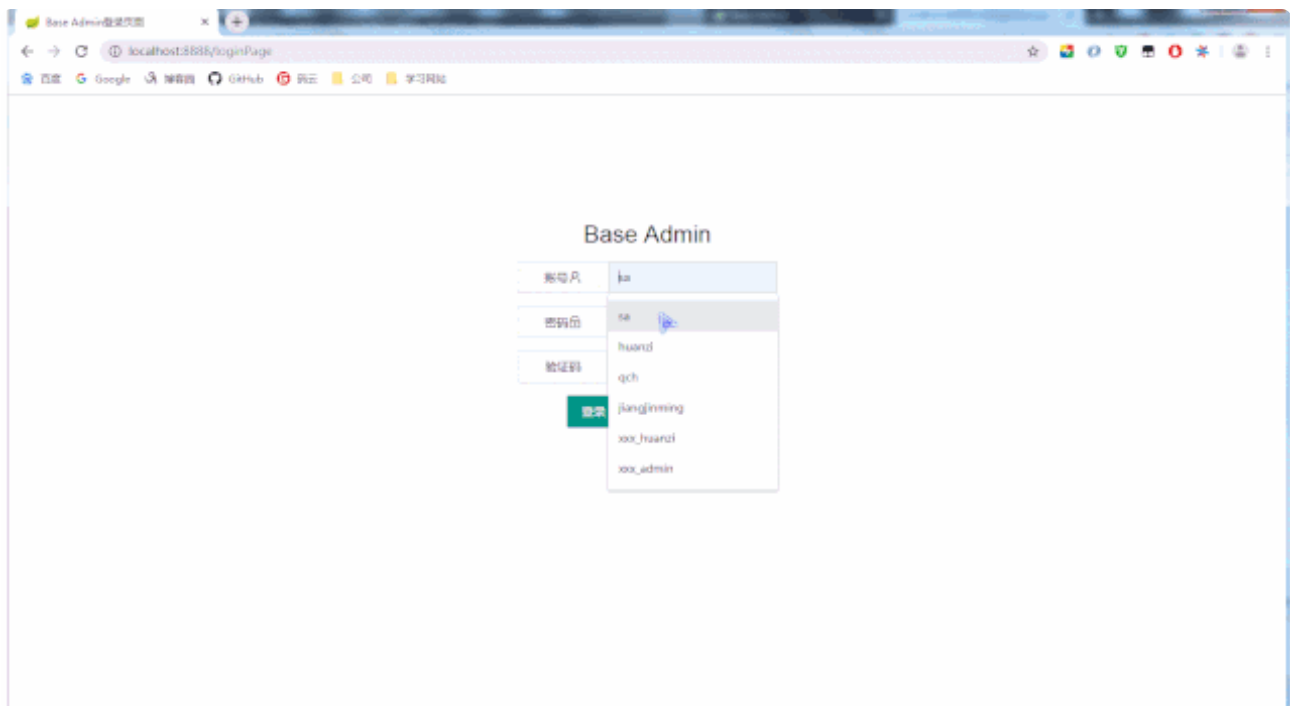


功能演示

登录

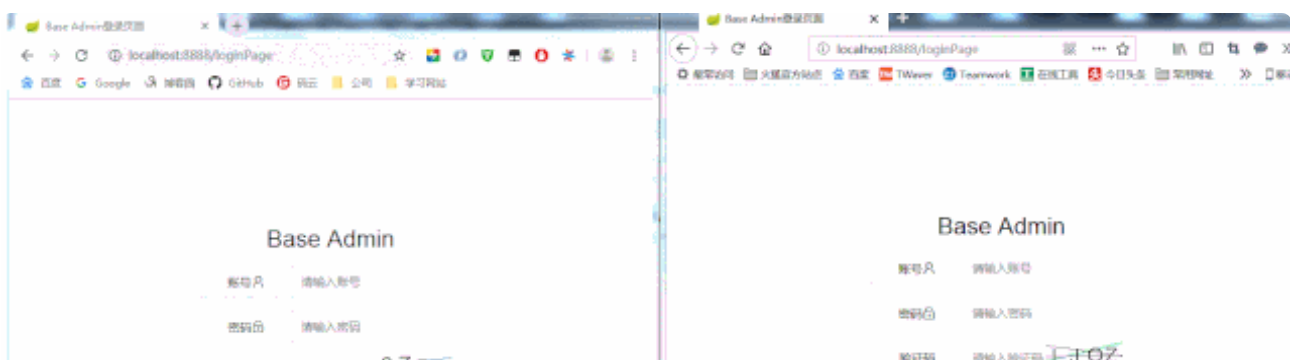
(为了方便演示，密码输入框的类型改成text)

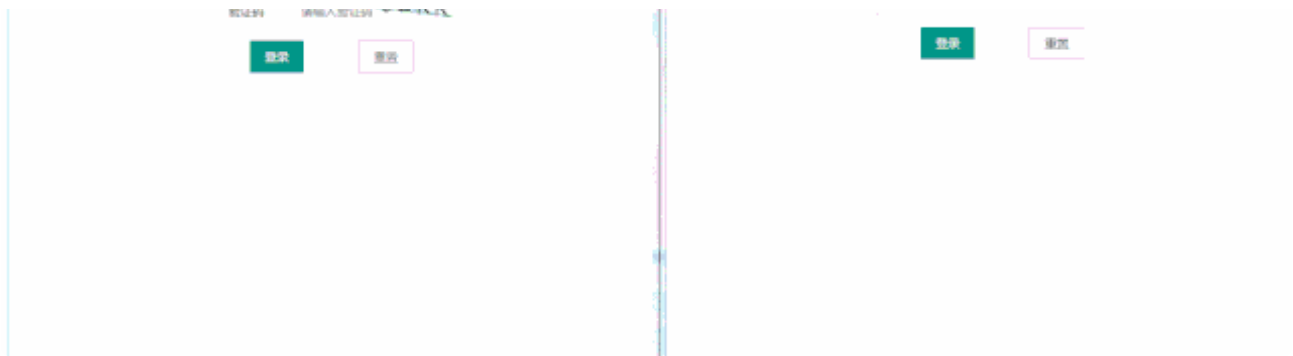
配置文件分支选择，dev环境无需输入验证码



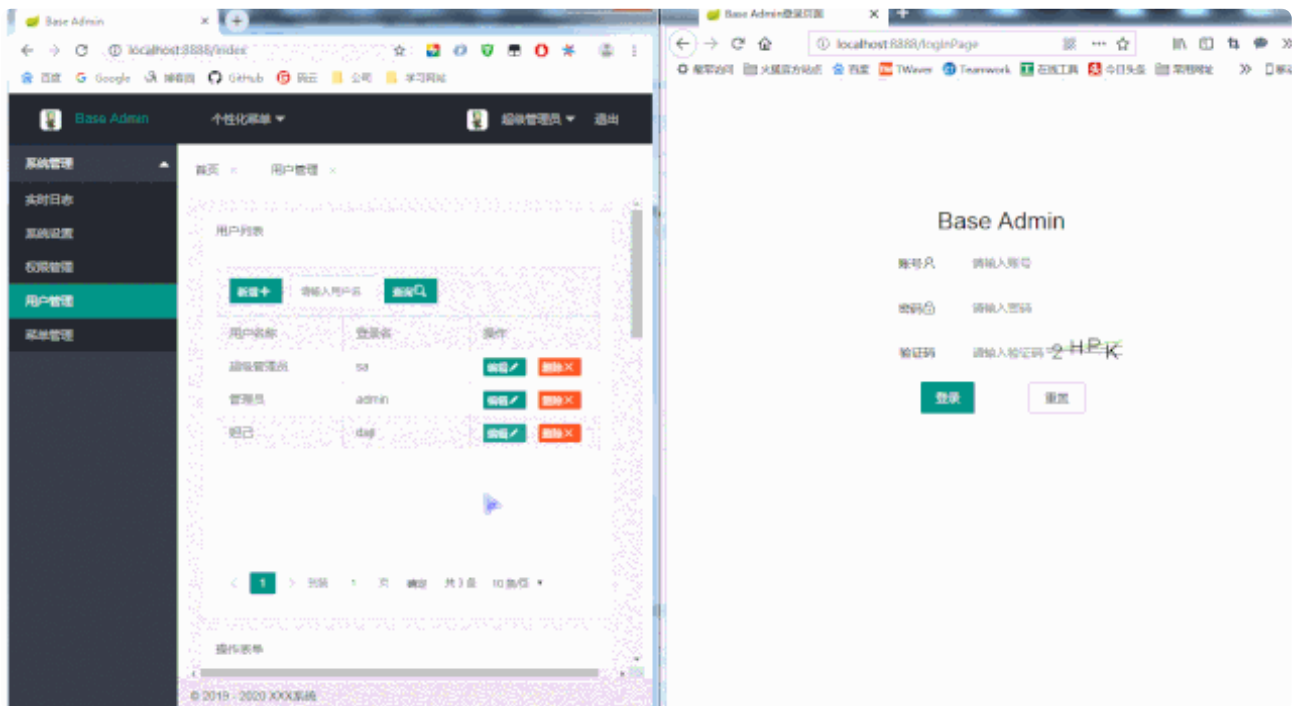
同时支持多种登录限制

允许/禁止账号多人在线

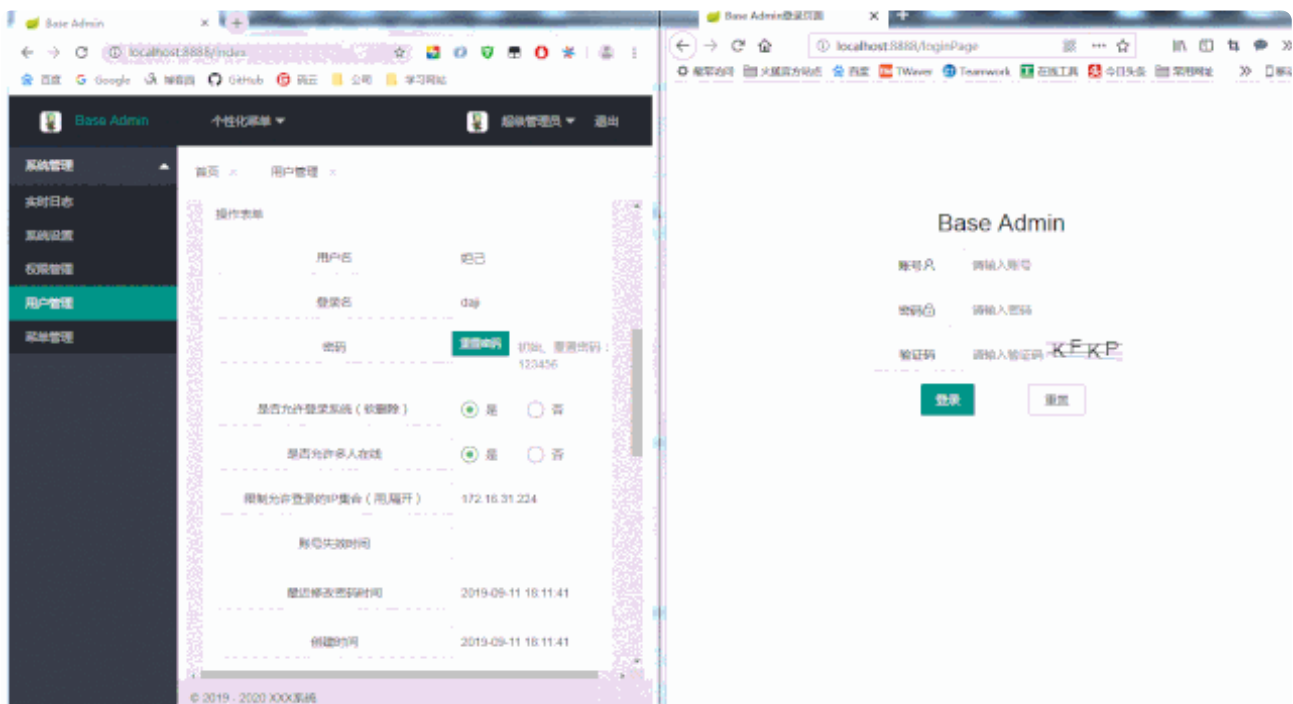




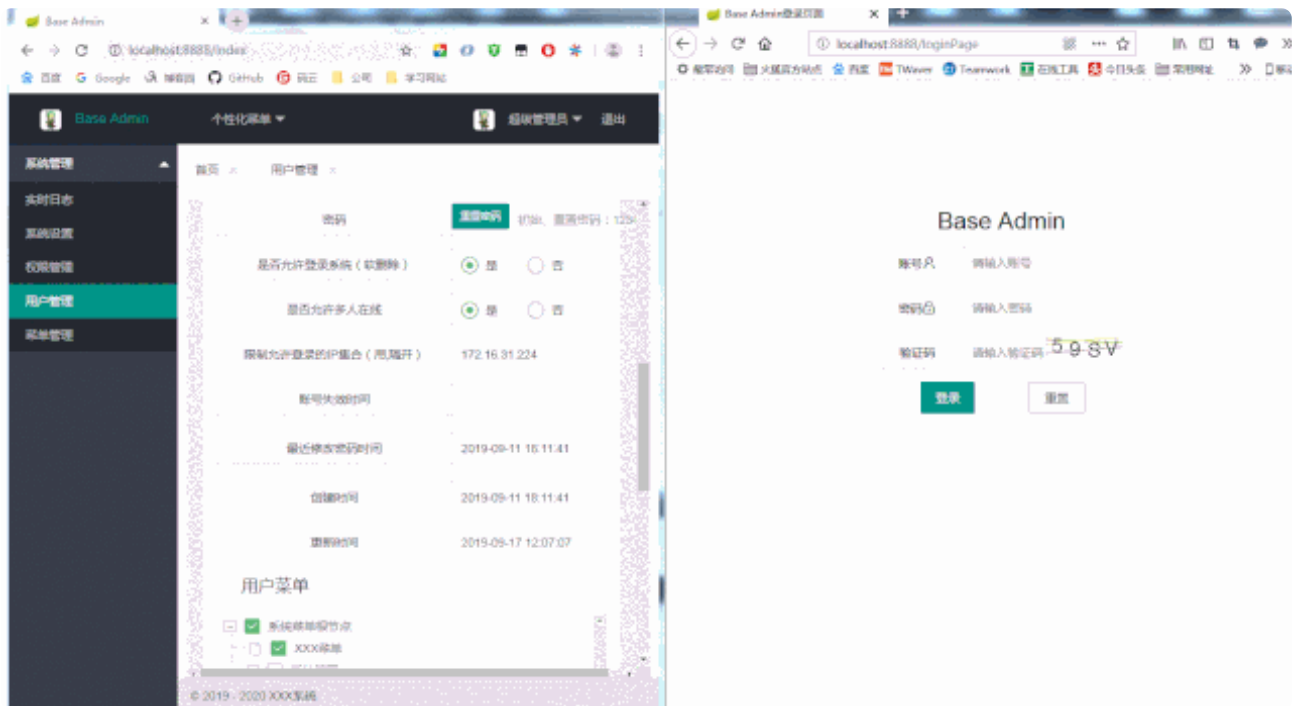
软删除



限制登录IP地址



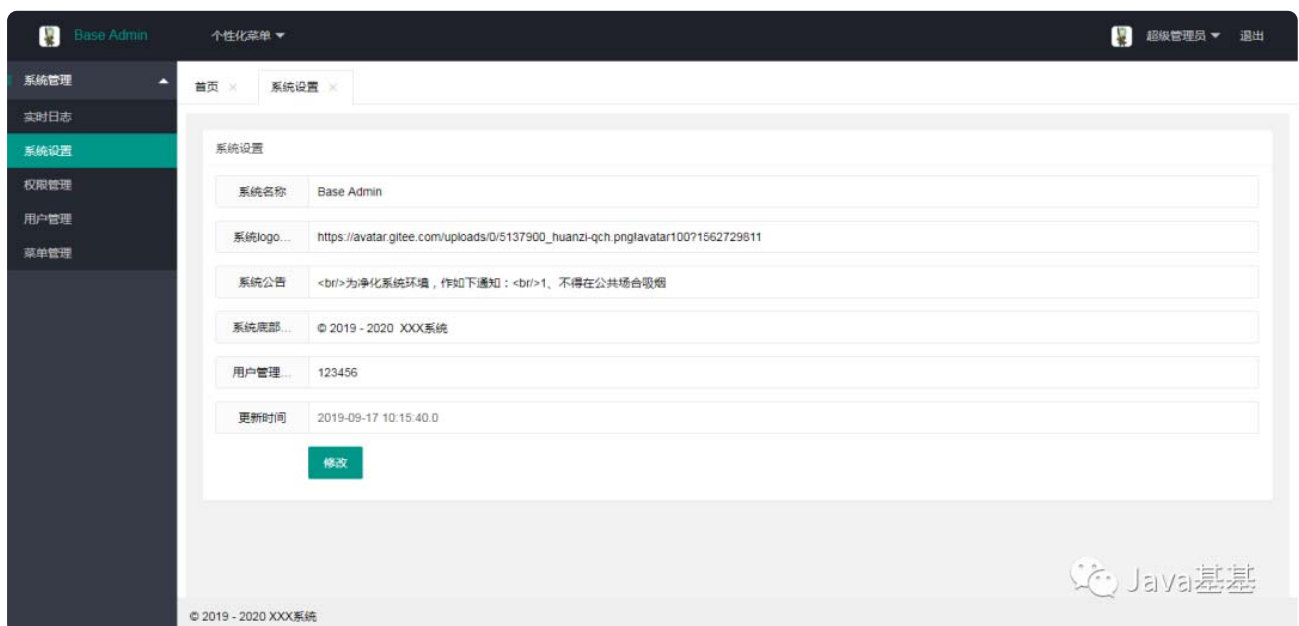
账号过期



更多登录限制，还可以继续扩展

系统设置

一下简单的系统属性设置，想支持更多的配置可自行扩展（比如这里的：用户管理初始、重置密码）

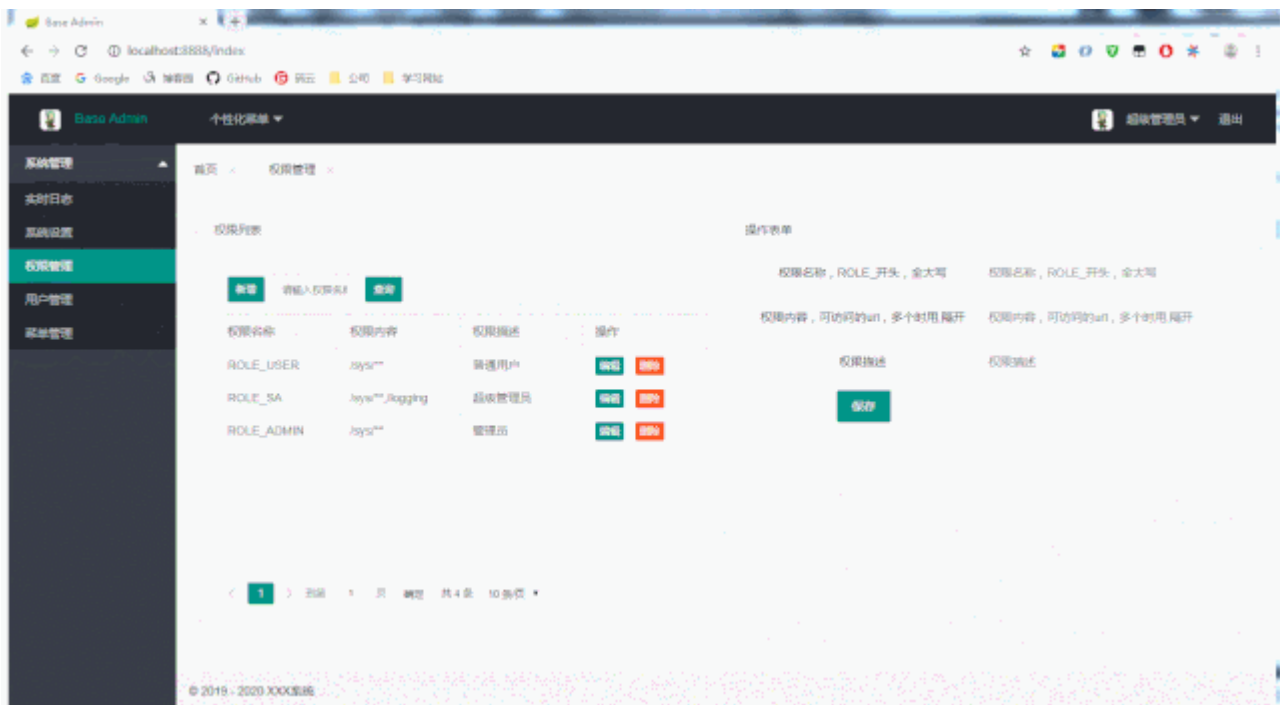


系统设置新增部分功能，详见文末“补充更新”

菜单管理

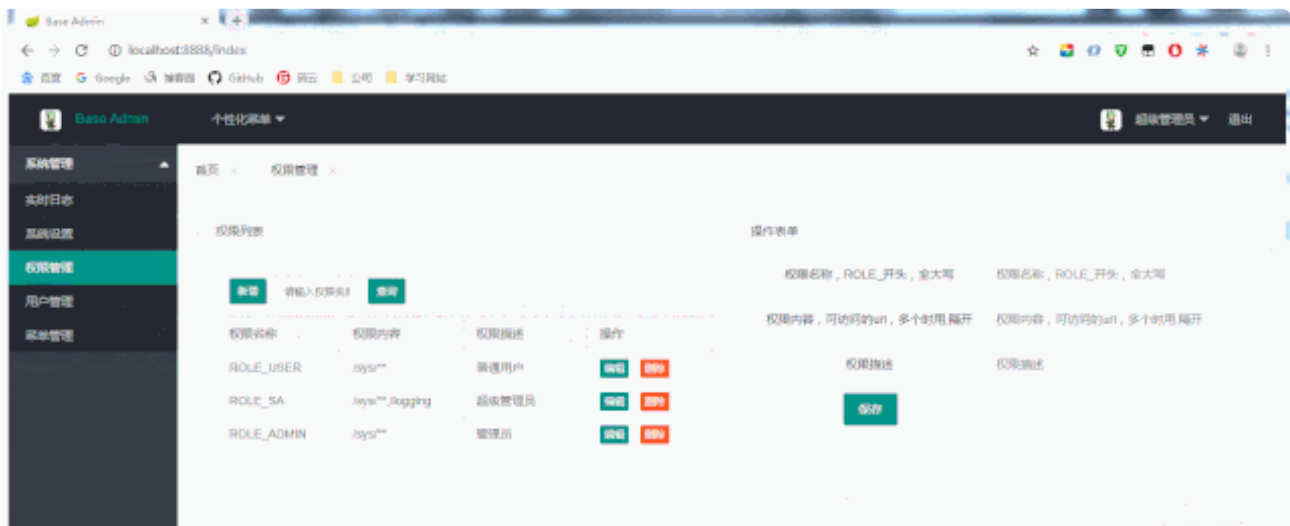
菜单管理是一棵layui的Tree

增删改



权限管理

增删改查

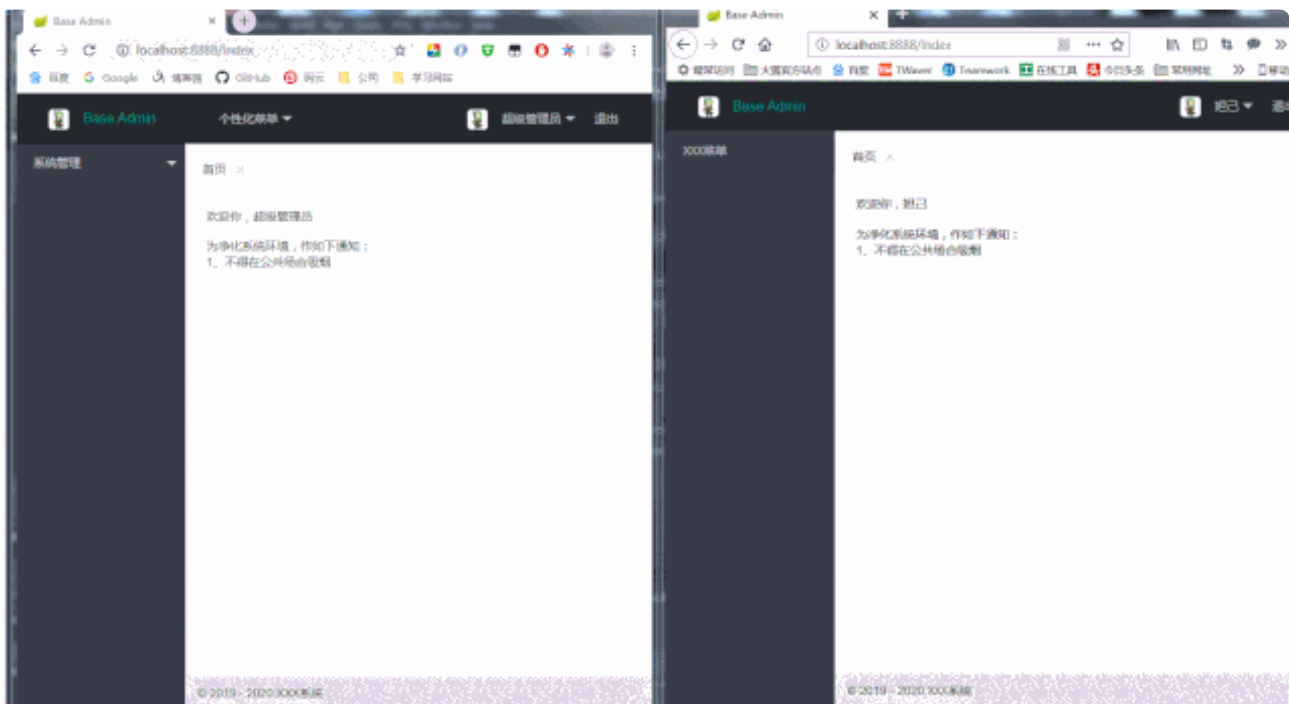




动态权限加载

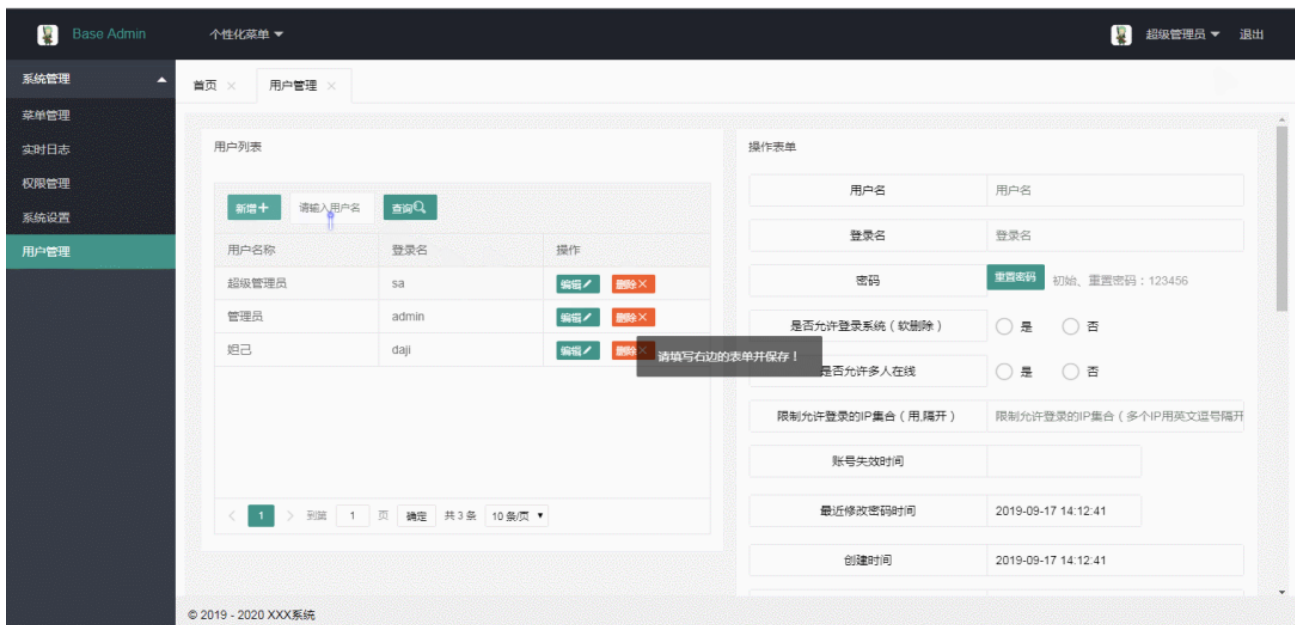
权限的加载并不是写死在代码，而是动态从数据库读取，每次调用save方法时更新权限集合

- 1、姐已是ROLE_USER权限，权限内容为空，无权访问/sys/下面的路径（<http://localhost:8888/sys/sysUser/get/1>）
- 2、使用sa超级管理员进行权限管理编辑，给ROLE_USER的权限内容添加 /sys/**，姐已立即有权限访问（<http://localhost:8888/sys/sysUser/get/1>）

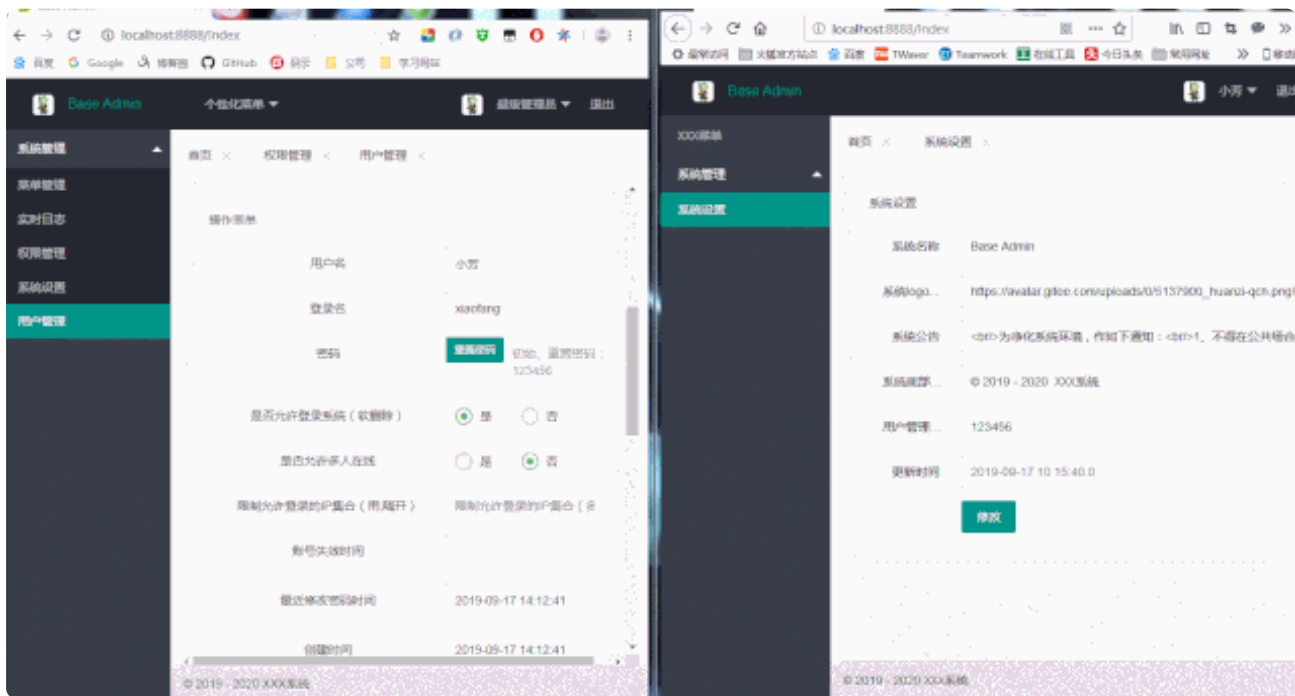


用户管理

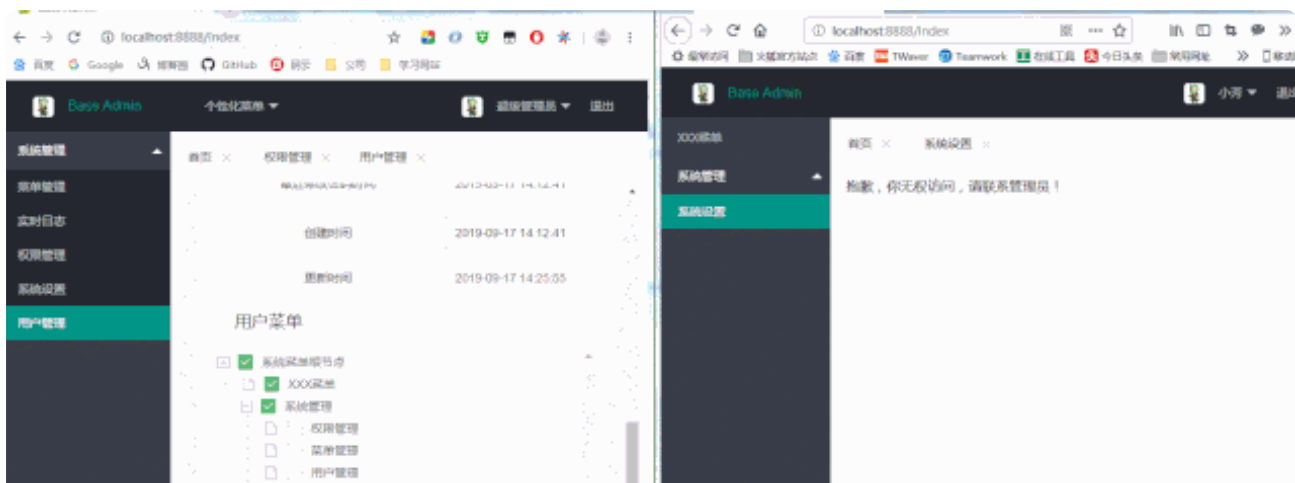
主要包括用户信息、登录限制的维护，菜单、权限的分配等



修改用户权限是下一次登录生效



修改用户菜单是刷新系统即可生效



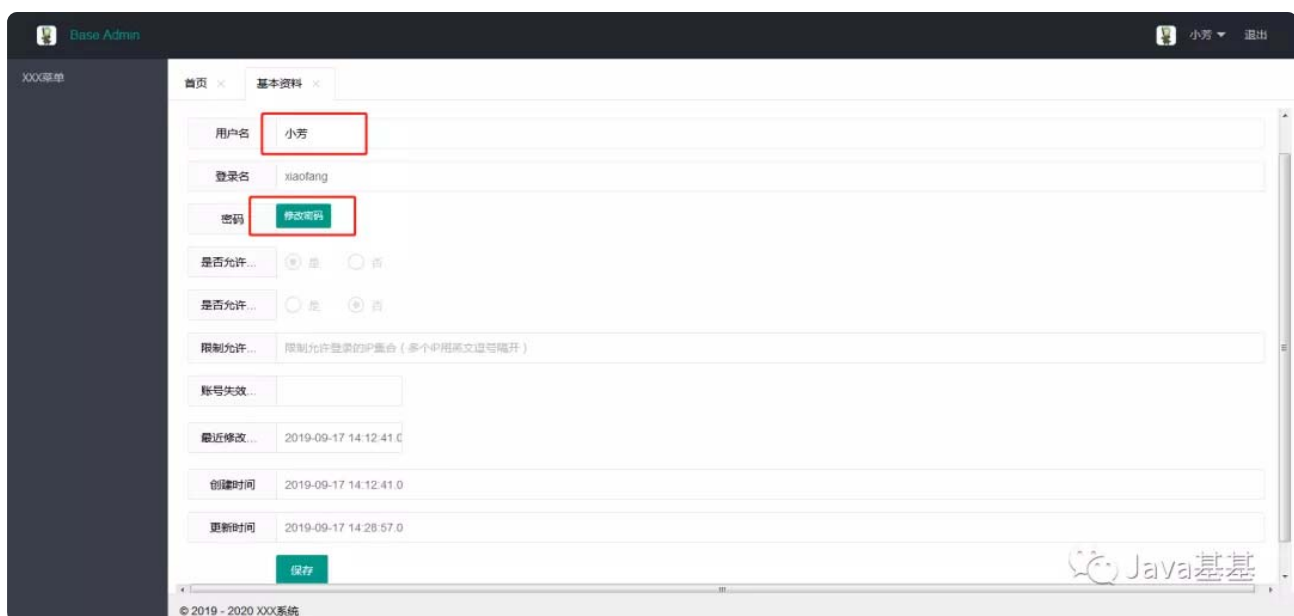


用户管理新增“当前在线用户”管理，详见文末“补充更新”

登录用户信息

基本信息

登录用户只能修改部分信息，例如名称、修改密码



修改密码

密码使用的是MD5加密并转换为16进制字符串存储，用户除了能主动修改密码外，还能叫管理员重置密码





修改密码成功，请重新登录系统！

修改



Java

修改密码



原密码

123456

新密码

111111

修改



修改密码成功，请重新登录系统！



Java

```
/**
 * 生成MD5加密串
 */
public static String getMD5(String message) {
    String md5 = "";
    try {
        //创建一个md5算法对象
        MessageDigest md = MessageDigest.getInstance("MD5");
        byte[] messageByte = message.getBytes(StandardCharsets.UTF_8);
        //获得MD5字节数组,16*8=128位
        byte[] md5Byte = md.digest(messageByte);
        //转换为16进制字符串
        md5 = ByteUtil.bytesToHex(md5Byte);
    } catch (Exception e) {
        //抛出异常("MD5加密失败!")
    }
}
```

```
log.error("getMD5报错!", e);  
}  
  
return md5;  
}
```

Java基金

localhost:8888/loginPage

Base Admin

账号	请输入账号
密码	请输入密码
验证码	请输入验证码 5453

登录 重置

个性菜单

用户可以自行配置自己的个性化快捷菜单



实时日志

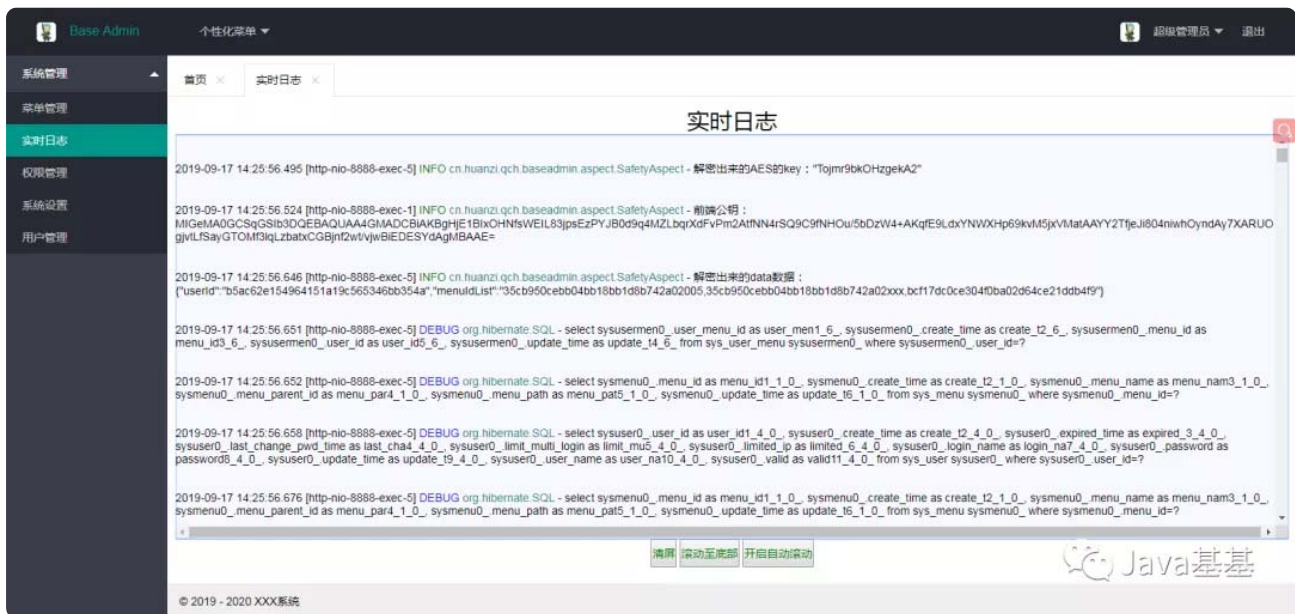
使用websocket，实时将日志输出到web页面，1秒刷新一次

详情请看我之前的博客：SpringBoot系列——Logback日志，输出到文件以及实时输出到web页面

注意：这里的日志配置只配置了dev环境，prod环境尚未为空，发布生产环境前记得先配置，否则生成的日志文件将不会输入日志内容！




```
65         <logger level="WARN" name="org.springframework"/>
66         <logger level="WARN" name="com.netflix"/>
67         <logger level="DEBUG" name="org.hibernate.SQL"/>
68     </springProfile>
69     <springProfile name="prod">
70
71     </springProfile>
72 </configuration>
```



实时监控

2020-06-10更新

实时监控的是系统硬件环境、以及jvm运行时内存，使用websocket，实时将数据输出到web页面，1秒刷新一次

windows环境



已用内存: 10.00 (G)	已用磁盘: 36.00 (G)
JVM堆内存	JVM非堆内存
初始大小: 256.00 (M)	初始大小: 2.00 (M)
最大可用: 3623.00 (M)	最大可用: .00 (M)
已使用: 463.00 (M)	已使用: 105.00 (M)
已申请: 739.00 (M)	已申请: 110.00 (M)
© 2019 - 2020 XXX系统	

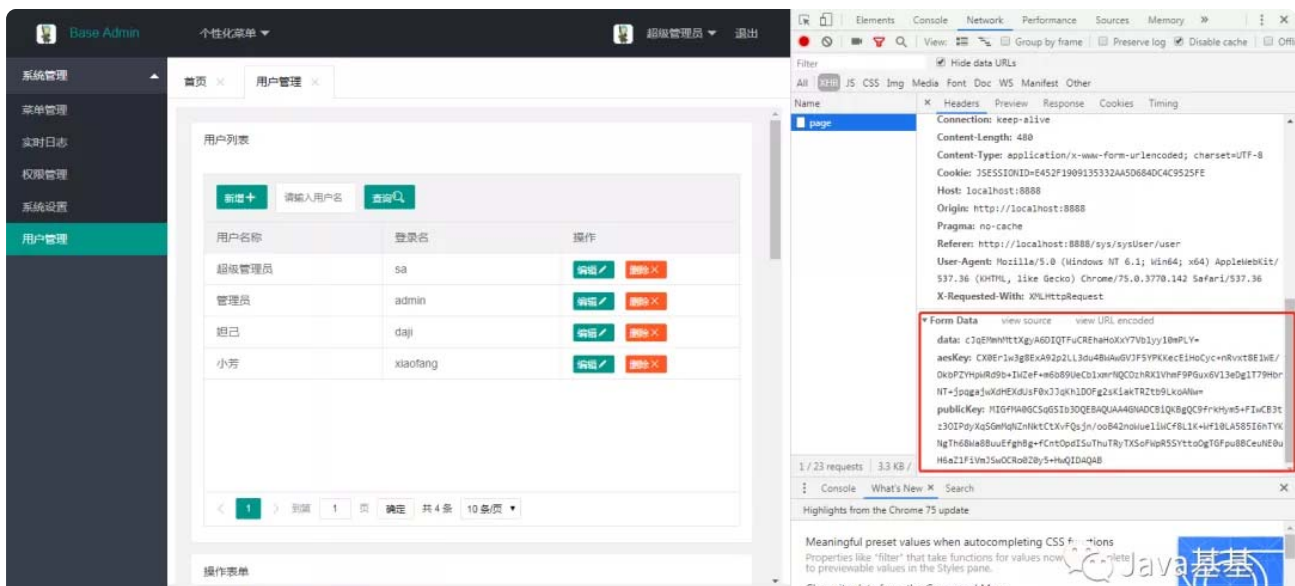
Linux环境



API加密

详情请看我之前的博客：前后端API交互数据加密——AES与RSA混合加密完整实例

请求参数加密





响应数据加密

The screenshot shows the 'Base Admin' system interface. On the left is a sidebar with navigation links: 系统管理, 菜单管理, 实时日志, 权限管理, 系统设置, and 用户管理 (highlighted). The main content area shows the '用户列表' (User List) with a table of users: 超级管理员 (sa), 管理员 (admin), 姐己 (daji), and 小芳 (xiaofang). Each user has '编辑' (Edit) and '删除' (Delete) buttons. A search bar and pagination controls are at the bottom. On the right, the browser's network console shows a successful response from the 'page' endpoint, indicating a successful operation.

2020-06-04更新

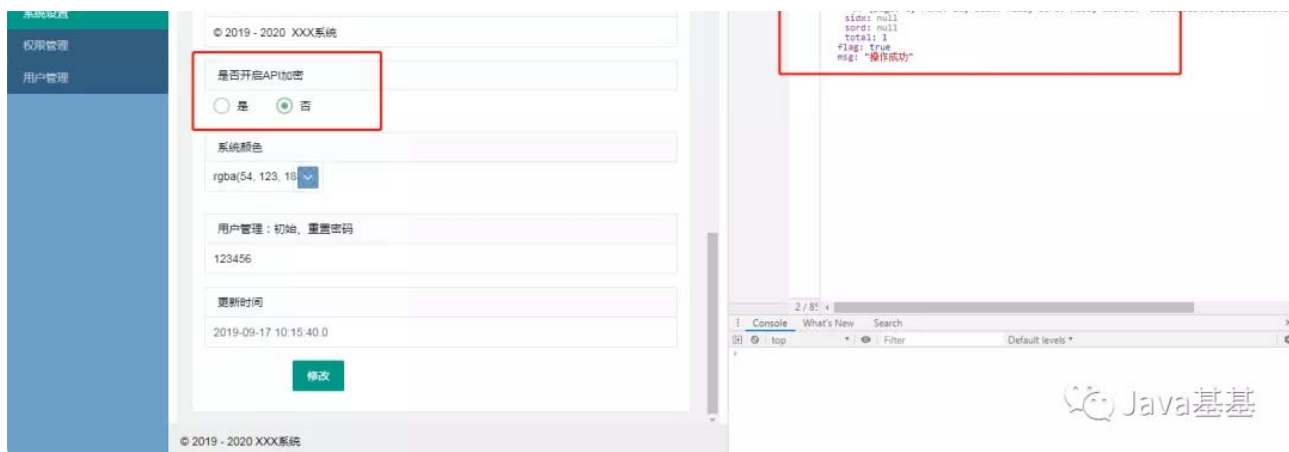
1、系统设置新增API加密开关，可一键关闭、开启API加密；

开启API加密

The screenshot shows the 'Base Admin' system interface. On the left is a sidebar with navigation links: 系统管理, 菜单管理, 实时日志, 权限管理, 系统设置, and 用户管理. The main content area shows the '系统设置' (System Settings) page. It includes fields for '系统名称' (System Name), '系统颜色' (System Color), '用户管理: 初始、重置密码' (User Management: Initial, Reset Password), and '更新时间' (Update Time). A red box highlights the '是否开启API加密' (Whether to enable API encryption) setting, which is currently set to '是' (Yes). On the right, the browser's network console shows a successful response from the 'page' endpoint, indicating that the API encryption has been successfully enabled.

关闭API加密

The screenshot shows the 'Base Admin' system interface. On the left is a sidebar with navigation links: 系统管理, 菜单管理, 实时日志, 权限管理, 系统设置, and 用户管理. The main content area shows the '系统设置' (System Settings) page. It includes fields for '系统名称' (System Name), '系统颜色' (System Color), '用户管理: 初始、重置密码' (User Management: Initial, Reset Password), and '更新时间' (Update Time). A red box highlights the '是否开启API加密' (Whether to enable API encryption) setting, which is currently set to '否' (No). On the right, the browser's network console shows a successful response from the 'page' endpoint, indicating that the API encryption has been successfully disabled.



关键点讲解

1、定制url访问权限，动态权限读取，需要自定义配置认证数据源、认证管理器、拦截器，详情步骤请参考：<https://www.jianshu.com/p/0a06496e75ea>；

2、API加密中，由于登录校验是Spring Security做的，因此我们要在UsernamePasswordAuthenticationFilter获取账号、密码之前完成解密操作，正好我们的校验验证码操作就是在它之前，同时要做响应数据的加密操作，所以登录部分的API加密光按照我们之前的博客来还是不够的，需要在CaptchaFilterConfig进行解密操作，解密后new一个自定义RequestWrapper设置Parameter，并将这个新对象传到doFilter交由下一步处理

2019-09-18补充：

3、还是API加密问题，我们是在程序启动的时候生成后端RSA密钥对，正常来说我们在访问登录页面进行登录的时候前端获取一下就可以了，但在开发环境中，我们通常开启热部署功能，改完代码程序可能会自动重启，但登录用户信息仍然保持在本地线程，系统依旧处于登录状态没有跳转到登录页面，导致后端公钥已经改变，但前端依旧用的是旧的后端公钥，所有导致加解密失败；解决：在访问index首页时也获取一下后端公钥，这样在开发的时候idea热部署后刷新页面就可以了（已提交最新代码，解决热部署后刷新页面还是API加解密失败问题；现在热部署后刷新页面即可）

2020-07-23补充：

4、好多人都不知道，项目有工具类CodeDOM.java可以生成一套单表的完整增删改查后台代码，具体详情可以看之前的博客：SpringBoot系列——Spring-Data-JPA（究极进化版）自动生成单表基础增、删、改、查接口

配置好数据库，指定代码生成父位置

```
/** 自动生成代码 ...*/
@Slf4j
public class CodeDOM {

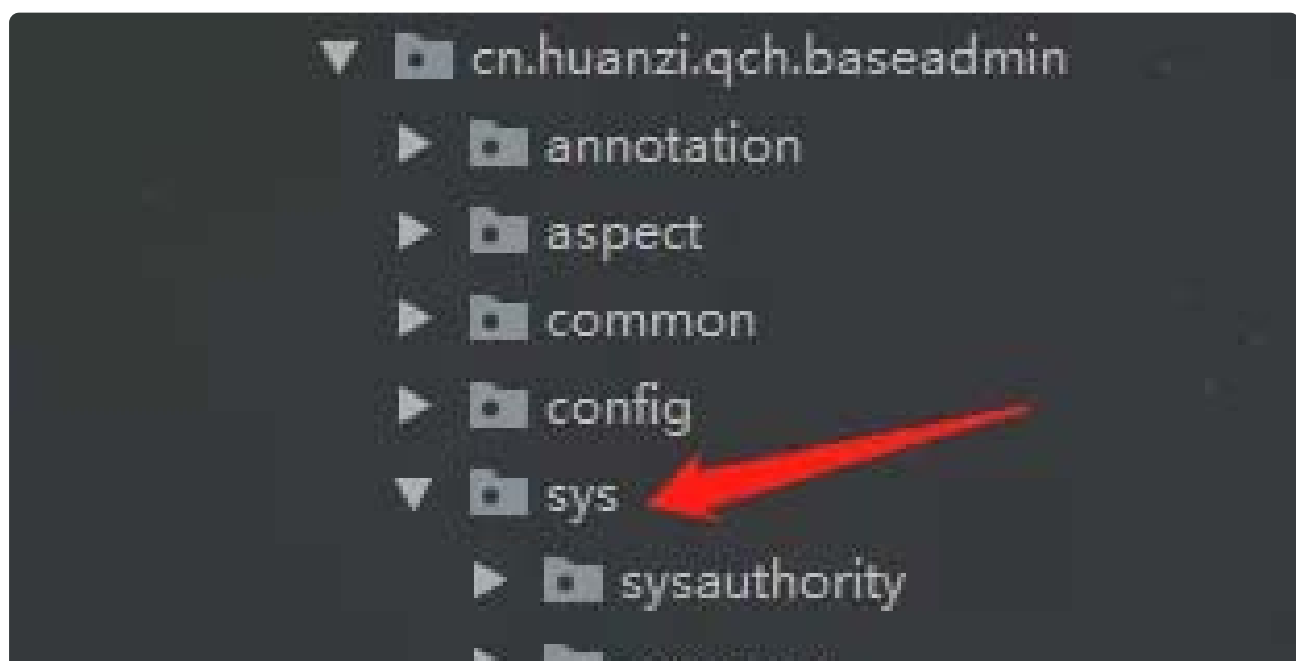
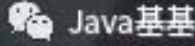
    /** 构造参数，出入表名 ...*/
    private CodeDOM(String tableName) {
        this.tableName = tableName;
        basePackage_ = "cn\\huanzi\\qch\\baseadmin\\sys\\";
        package_ = basePackage_ + StringUtil.camelCaseName(tableName).toLowerCase() + "\\";
        //System.getProperty("user.dir") 获取的是项目所在路径，如果我们是子项目，则需要添加一层路径
        basePath = System.getProperty("user.dir") + "\\src\\main\\java\\" + package_;
        //basePackage_ = "cn\\huanzi\\qch\\baseadmin\\";
    }

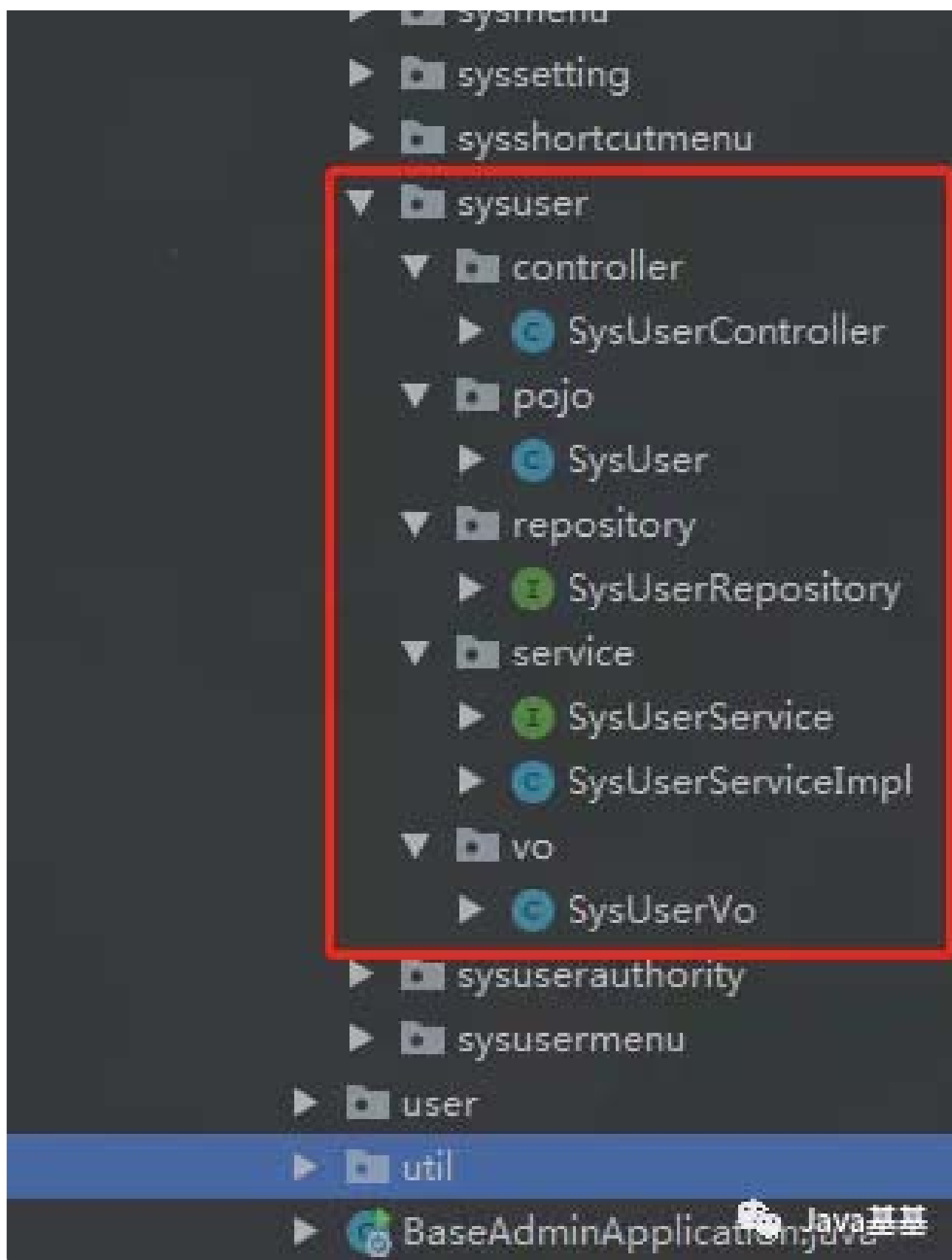
    /** 数据连接相关 ...*/
    private static final String URL = "jdbc:mysql://localhost:3306/test?serverTimezone=GMT%2B8&characterEncoding=utf-8";
    private static final String USERNAME = "root";
    private static final String PASSWORD = "123456";
    private static final String DRIVER_CLASSNAME = "com.mysql.jdbc.Driver";
}
```



运行main函数即可一键生成一套单表增删改查后台代码

```
public static void main(String[] args) {
    String[] tables = {"sys_user"};
    for (String table : tables) {
        String msg = new CodeDOM(table).create();
        System.out.println(msg);
    }
}
```





后记

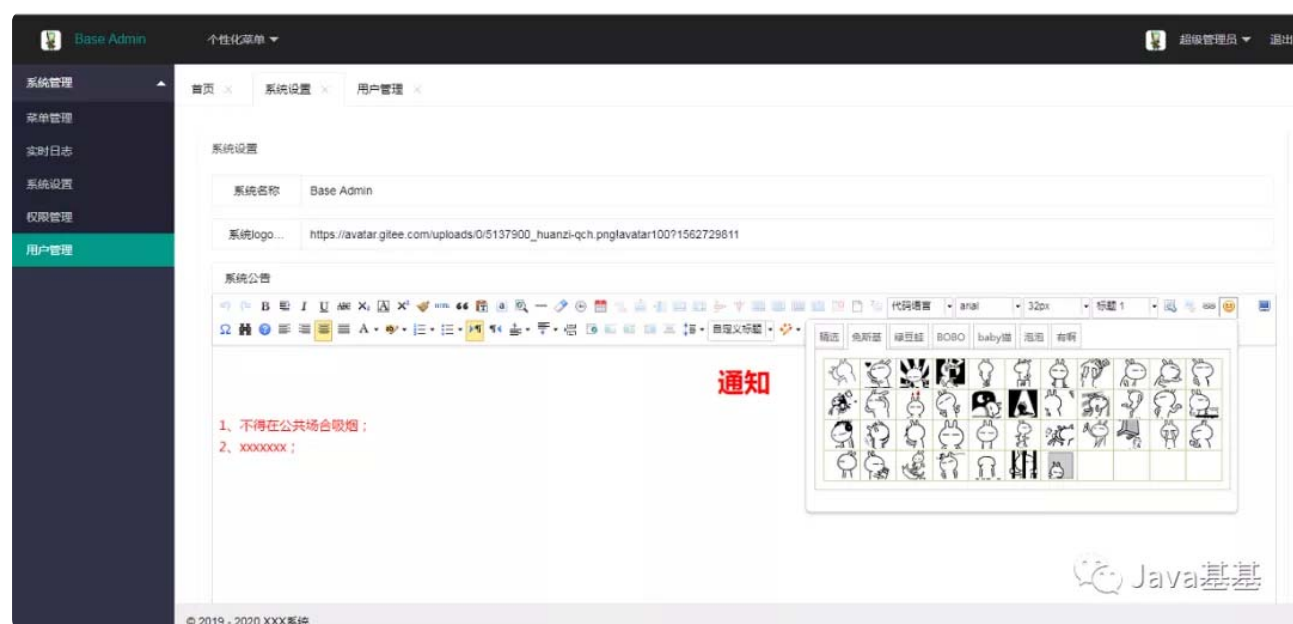
这个只是一个比较简单通用的后台系统，如果加入工作流，就可以升级成基础平台，为简化业务开发，将部分通用系统功能整理成独立项目，具体业务功能通过iframe嵌入

补充更新

2020-05-14更新

1、新增百度富文本的使用，但还没配置上传接口：UEditor文档

对应字段类型，mysql要改成longtext

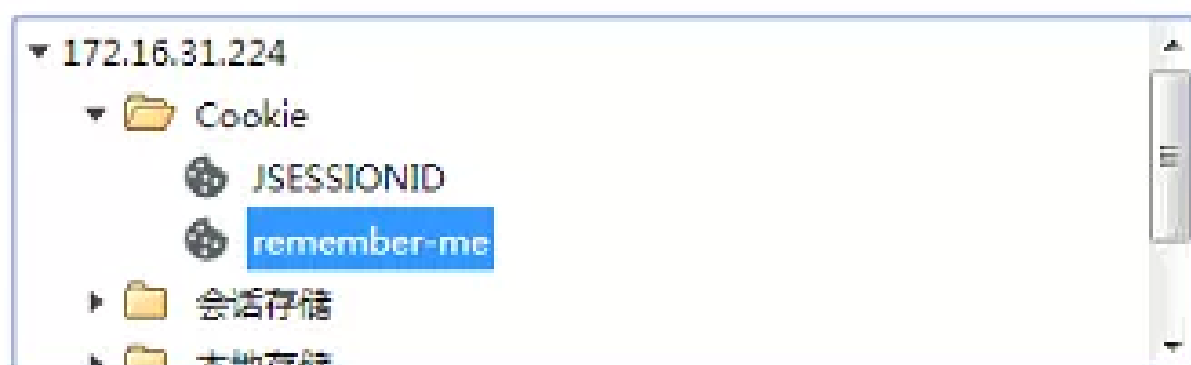


2、新增“记住我”功能，也就是rememberMe，原理以及源码探究请看这位大佬的博客：https://blog.csdn.net/qq_37142346/article/details/80114609



正在使用的 Cookie

以下 Cookie 是系统在您查看此网页时设置的



名称	remember-me
内容	ZU9MUkp4NEpUeEM2djFveGtKOGF3JTNEJTNE
域名	172.16.31.224
路径	/
为何发送	各种连接
创建时间	2020年5月14日星期四 下午4:46:36
到期时间	2020年5月21日星期四 下午4:46:36

禁止

删除

完成
Java 基金

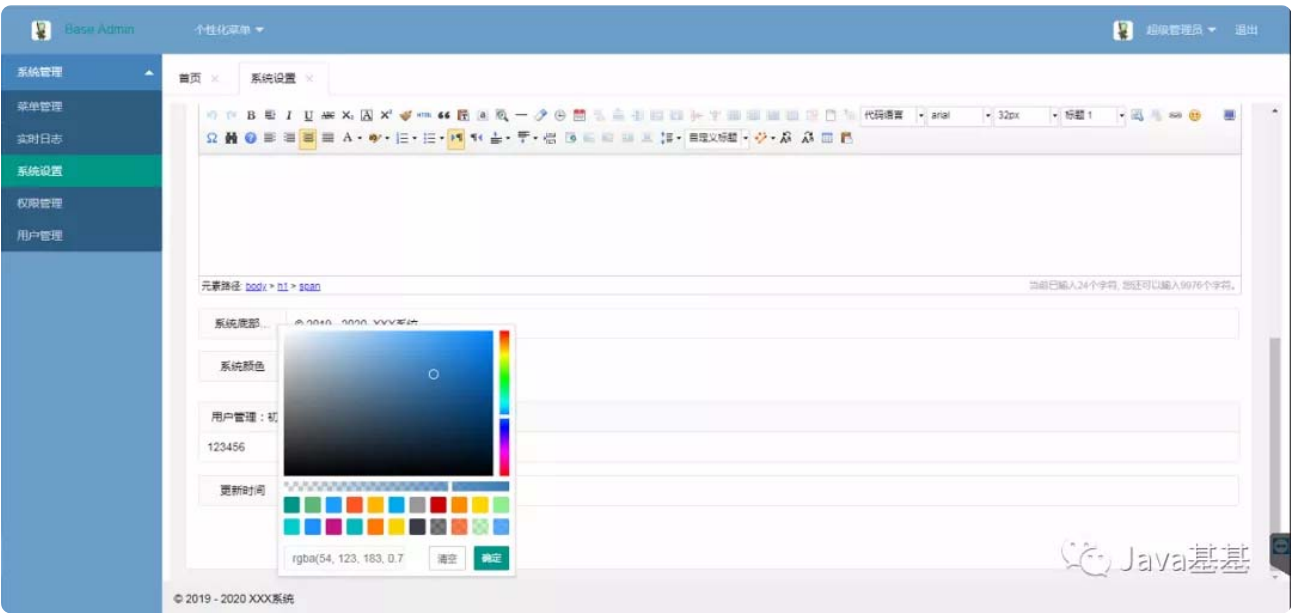
需要新增一张表，SQL文件我也以及更新了

对象 persistent_logins @test (loc...			
series	username	token	last_used
5EOtsD/OXFw+U4KV+VfNYA==	sa	reOLRJx4JTxC6v1oxkJ8aw==	2020-05-14 16:48:36

3、修复若干bug，使其系统越来越完善

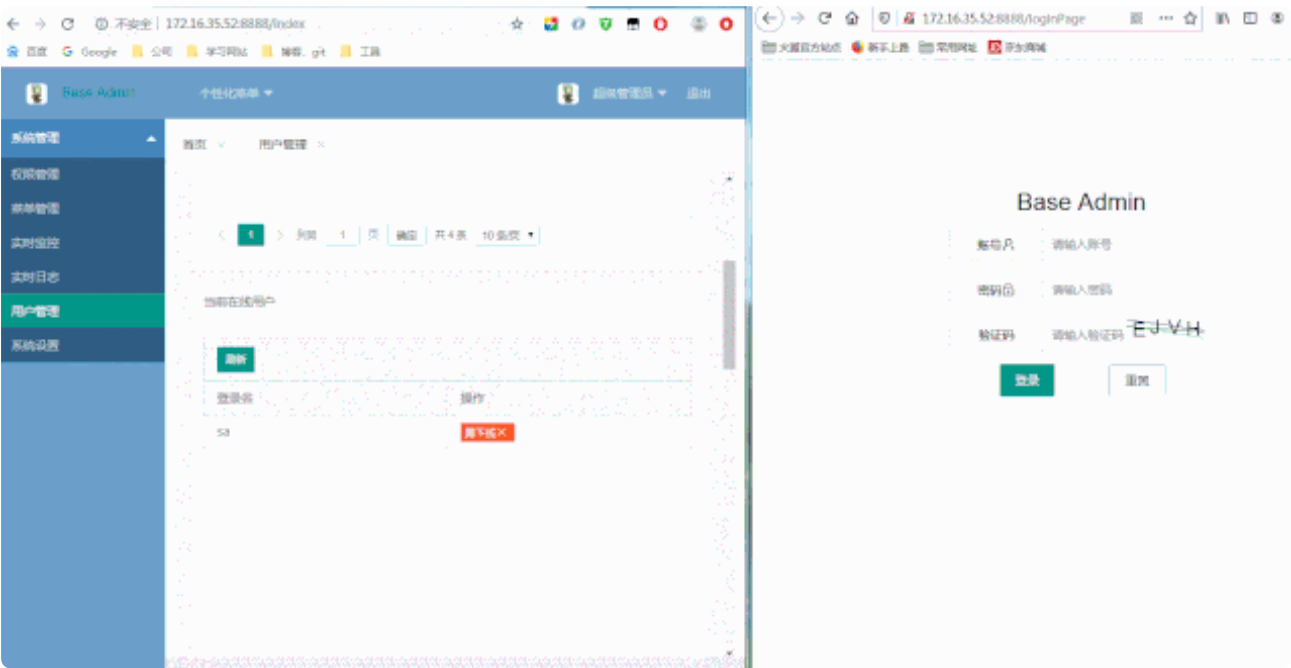
2020-05-16更新

1、系统设置新增系统颜色，头部、左侧菜单的颜色可按心情切换（SQL文件已同步更新）



2020-07-02更新

1、用户管理模块新增“当前在线用户”管理，可实时查看当前在线用户，以及对当前在线用户进行强制下线操作



代码开源

注：数据库文件在resources/static/sql目录下面

代码已经开源、托管到我的GitHub、码云：

GitHub：<https://github.com/huanzi-qch/base-admin>[

](<https://github.com/huanzi-qch/springBoot>)

码云：<https://gitee.com/huanzi-qch/base-admin>