

Spring Boot2x +redis+JWT+Swagger2+Mybatis 实战脚手架

背景

最近团队内来了几个新的同学，还是在校的大学生来公司实习的。大领导特意嘱咐我要好好的培养，经过和大领导的眼神交流后，深深的明白走向人生巅峰的机会来啦 ☺(^_^).....。通过考察发现，他们都有一个特点就是 java 基础但是没有实战经验，ssh\ssm都没有接触过。这就有些头痛了，我该怎么样才能让他们快速的融入我这个项目组呢？因为我这个项目组是 spring cloud 微服务，要想快速上手那就得先熟悉spring boot。找到了切入点后我就立马着手计划，首先搭建一个 Spring Boot2x + JWT + shiro + redis + Swagger2 + Mybatis 的脚手架，再把用户中心服务里的权限管理的业务给抽离出来。接入到我搭建的这个框架，这样他们就能学习spring boot也能学习到项目里的真实业务。我把框架搭建好后再把权限管理的业务实现，然后简单的给他们讲解一遍后，就要求他们每个人也搭建一个脚手架并实现权限管理模块的业务，可以完全照抄，但是不能全程复制粘贴，不会的立马提出来。经过十来天后他们已经熟悉spring boot、redis、shiro、jwt、swagger 这几个知识点了(这些都是公司里面用到的技术)，熟悉了关键知识点后我再安排他们把公司的项目给clone下来并要跑起来，然后 debug 慢慢熟悉业务。后面不用说啦圆满完成大领导布置的任务 ☺(^_^)。经过了几个新同学的实践故而现在分享给大家。

适合人群

- 在校大学生
- 刚刚毕业实习
- 想往spring cloud 方向发展的程序猿

得到什么

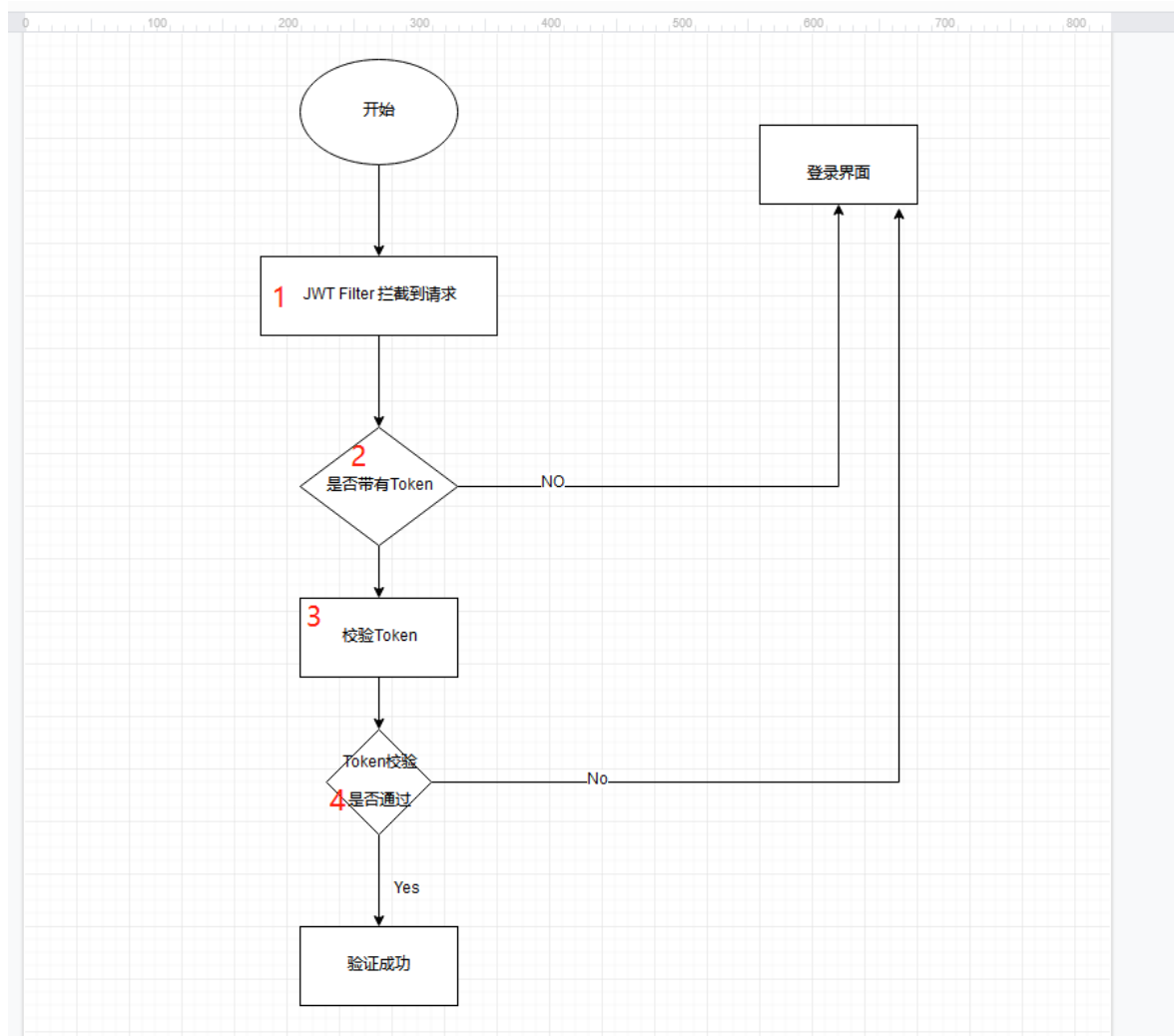
- 快速上手搭建企业实战框架(全程注释)
- 快速熟悉Spring boot
- 前后端分离 分布式sessionId 管理
- 掌握redisTemplate 的使用姿势
- 掌握 shiro
- 掌握 JWT的使用姿势(token refreshToken 使用姿势)
- 掌握Swagger2x
- 掌握generatorConfiguration 逆向生成代码
- 掌握权限管理系统的业务

关键点

sessionId 管理

- 有状态：
 - 就是用户登录进来生成一个token(UUID) 存在redis 并设置过期时间，把token返回给客户端，每次客户请求后端接口都把token带上，后端拦截器拦截到请求后校验token，如果token通过校验则进入controller处理具体业务，校验不通过引导用户到登录界面重新登录
- 无状态：
 - 用户登录进来用 JWT 生产两个token (access_token过期时间短、refresh_token过期时间长)，每次客户端请求后端接口都把access_token带上经过后端 JWT Filter 拦截后校验携带的access_token，如果通过校验则进入controller处理具体业务，校验不通过后端返回特定的状态码告知客户端携带refresh_token来刷新token，首先后端要校验refresh_token,校验通过生成新的access_token,校验不通过引导客户端到登录界面。

JWT 使用姿势



解读：

1. 首先判断access_token 是否为空

```

//为空抛出特定异常引导到登录界面
if(StringUtils.isEmpty(token)){
    throw new BusinessException(BaseResponseCode.TOKEN_ERROR);
}

```

2. 校验token

```

/**
 * 用户主动退出 引导用户到登录界面
 */
if(redisService.hasKey(Constant.JWT_REFRESH_TOKEN_BLACKLIST+accessToken)){
    throw new BusinessException(BaseResponseCode.TOKEN_ERROR);
}
/**
 * 存在这个key 说明接口已经刷新过还没有来得及更换新的 token
 * 直接放行 进入到 controller处理具体业务
 */
if(redisService.hasKey(Constant.JWT_REFRESH_STATUS+accessToken)){
    return true;
}
/**
 * token 已经过期 或者主动去刷新
 * 返回特定响应码 告知客户端去调用刷新token 接口
 * 刷新成功后拿到新的token 然后再重新请求接口
 */

if(JwtTokenUtil.isTokenExpired(accessToken,tokenSettings.getSecretKey())||redisService.hasKey(Constant.JWT_REFRESH_KEY+accessToken)){
    throw new BusinessException(BaseResponseCode.TOKEN_PAST_DUE);
}

```

Constant.JWT_REFRESH_STATUS+accessToken 这个key判断就是为了防止一个功能一次性请求多个接口多次刷新token。