# 官方文档

https://www.kancloud.cn/owenwangwen/newocp/3128719

# git地址

https://gitee.com/dromara/open-capacity-platform

# 功能

## 1、统一安全认证中心多因子融合认证

## 2、微服务架构基础支撑

（1）服务注册发现、路由与负载均衡

（2）统一配置中心

（3）服务熔断和限流

（4）统一日志中心

（5）统一文件中心

## 3、运维监控

（1）nginx监控

（2）服务器监控

（3）应用监控

# 快速启动框架

## 1、修改平台统一配置

在config-spring-boot-starter项目中

## 2、启动nacos注册中心（不是必须的）

平台将nacos源码方式集成到平台中，更方便的启动控制nacos

涉及的类：Nacos

## 3、启动用户中心

涉及的类：UserCenterApp

## 4、启动认证中心

涉及的类：AuthServerApp

## 5、启动网关

涉及的类：ApiGateWayApp

## 6、启动后台管理中心

涉及的类：BackCenterApplication

## 7、访问系统

http://localhost:8066/login.html

用户名：admin

密码：admin

# 项目模块

## business-center（业务中心）

### base-center

基础服务

#### 包含的项目中的依赖

banner-spring-boot-starter

config-spring-boot-starter

db-spring-boot-starter

map-spring-boot-starter

redis-spring-boot-starter

log-spring-boot-starter

common-spring-boot-starter

swagger-spring-boot-starter

sentinel-spring-boot-starter

loadbalancer-spring-boot-starter

uaa-client-spring-boot-starter

### user-center

用户中心

#### 包含的项目中的依赖

## inner-intergration

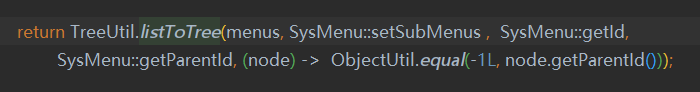
内部集成二方包

### base-spring-boot-starter（基础能力）

#### 包含项目的中依赖

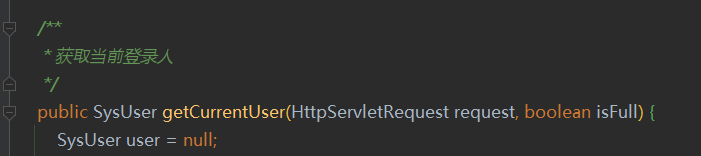
log-spring-boot-starter

#### 菜单树



#### 获取用户的信息

第一种方法：



SysUser user = UserUtils.getCurrentUser(request, true);

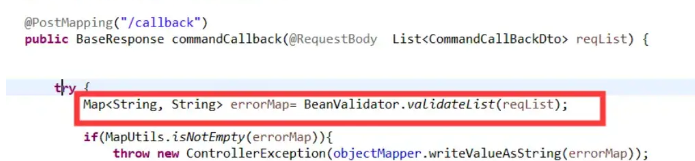
第二种方法：

com.open.capacity.uaa.common.util.AuthUtils.getLoginAppUser()

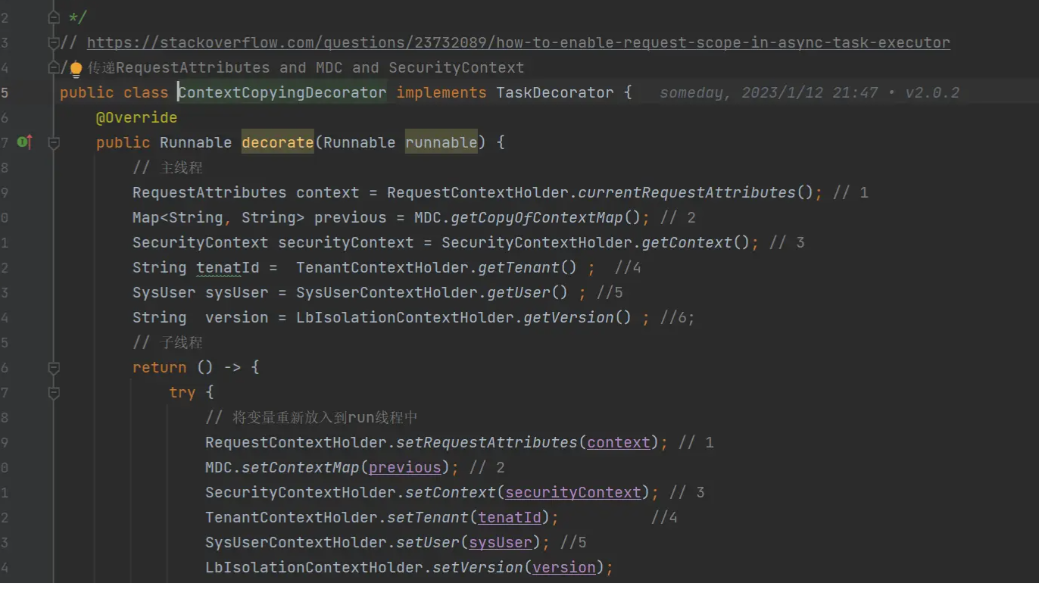
#### 通用分页对象

PageResult.<Agents>builder().data(pageInfo.getList()).statusCodeValue(0).count(pageInfo.getTotal()).build();

#### BeanValidator参数校验

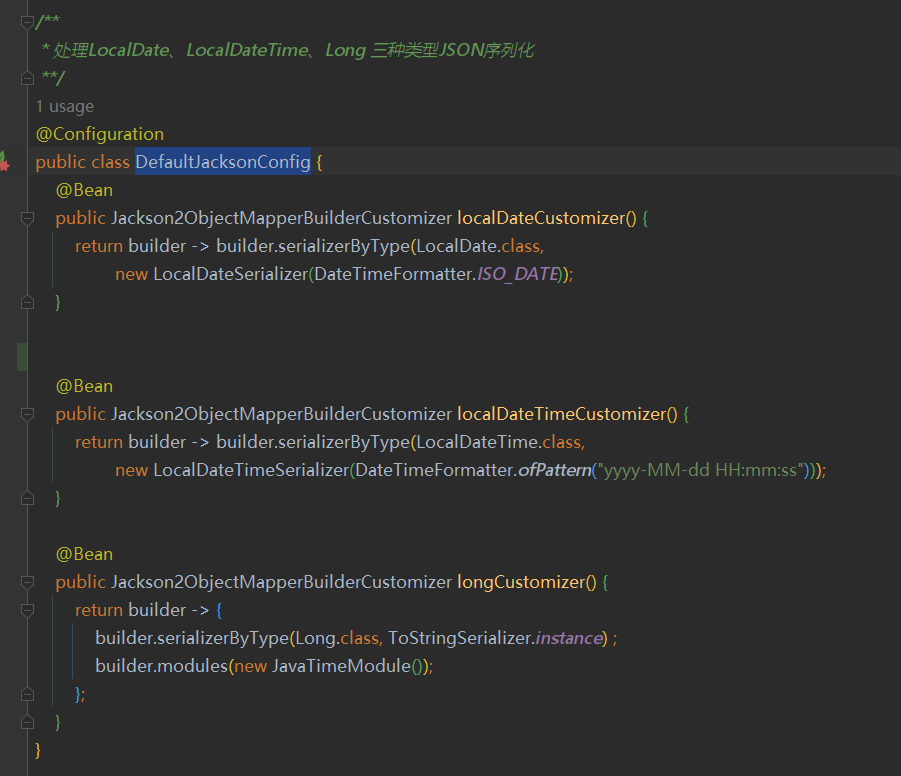


#### 父子线程异步传递问题

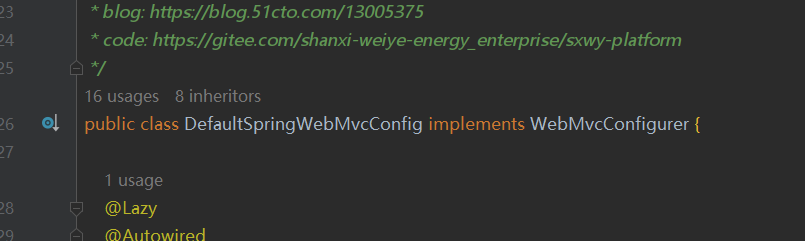


#### json通用配置

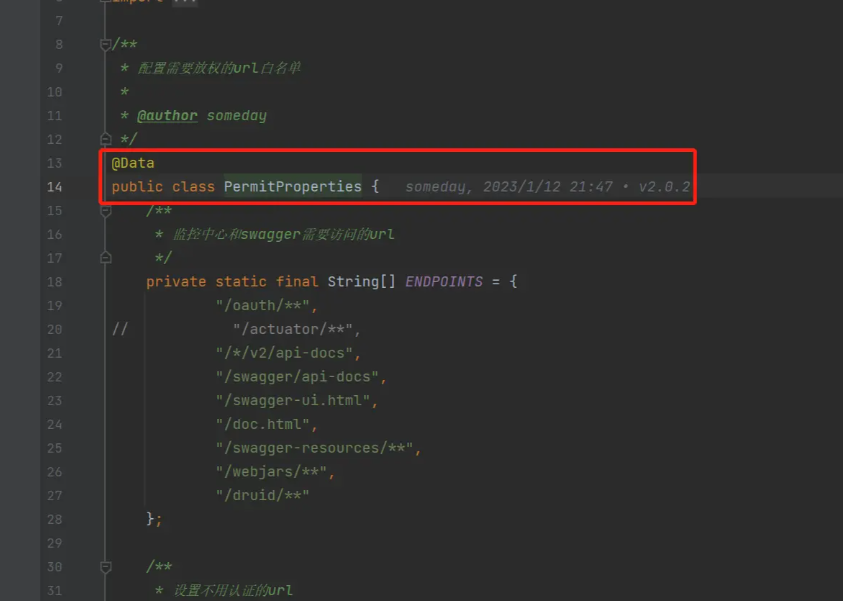
在DefaultJacksonConfig这个类中配置



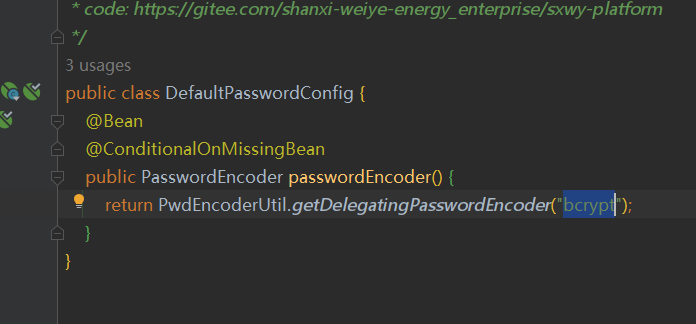
#### spring 安全防火墙配置



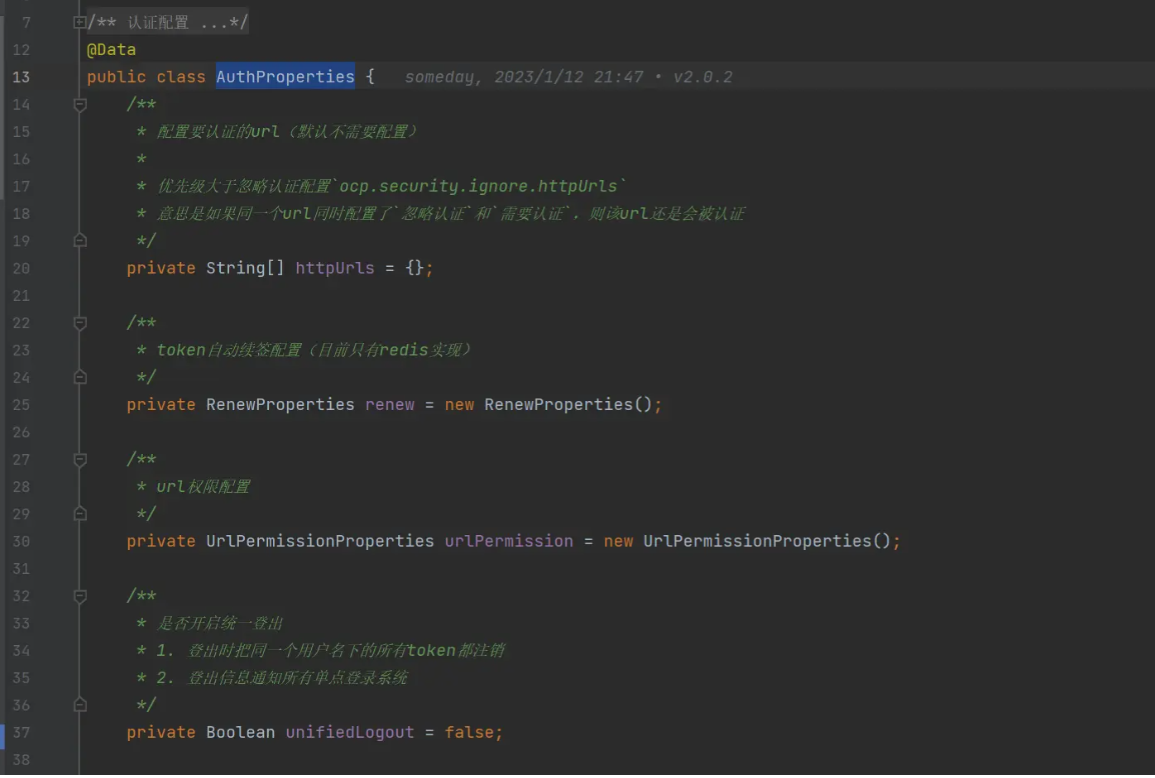
#### 白名单配置



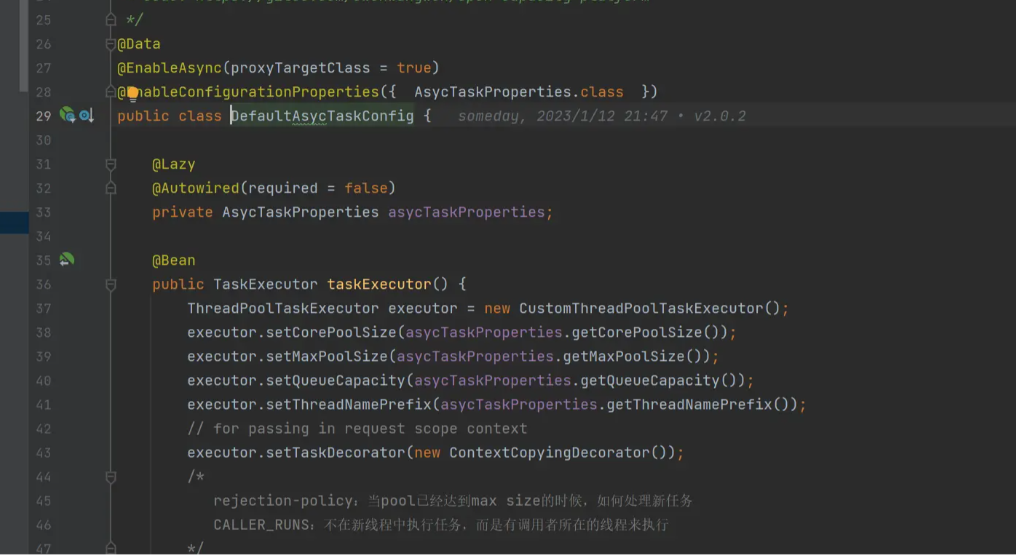
#### spring自定义密码处理器



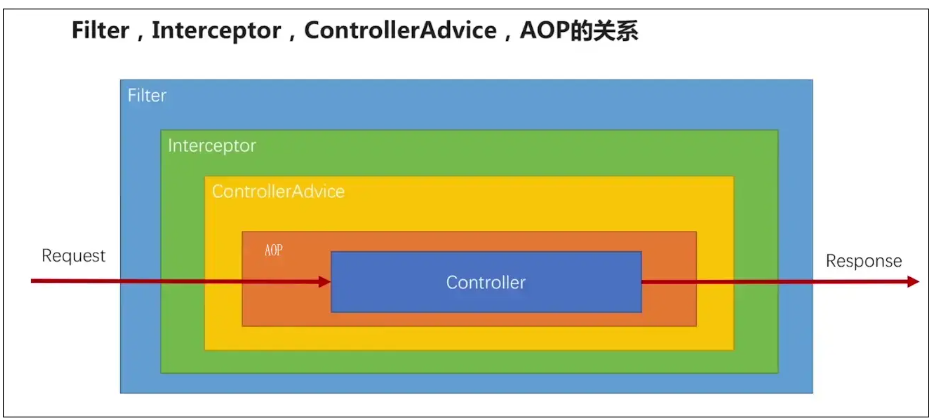
#### 白名单配置类

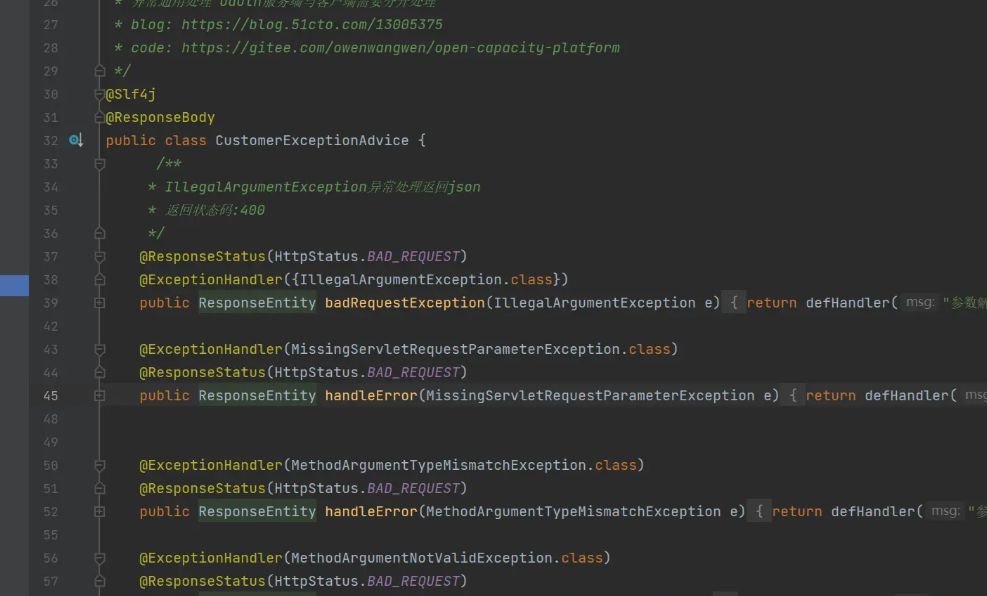


#### 通用线程池配置

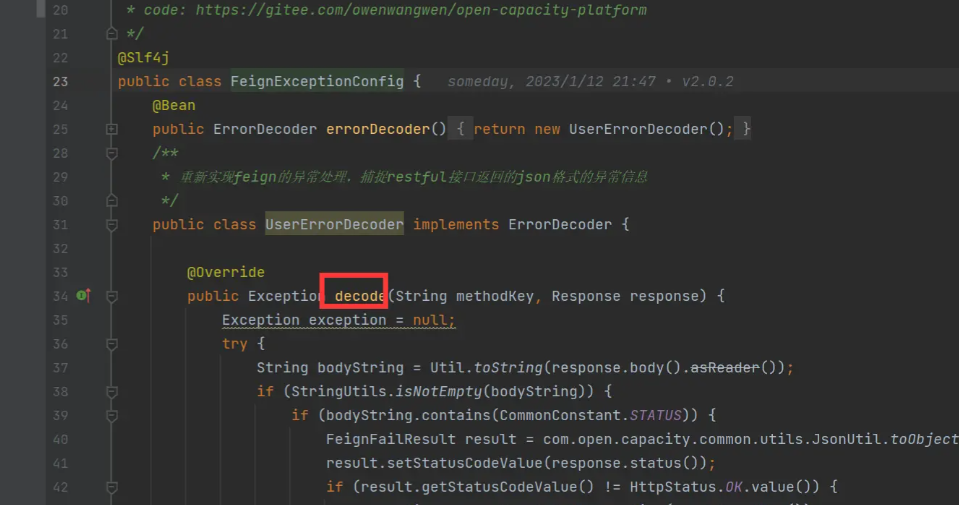


#### springboot异常





#### feign异常



### config-spring-boot-starter

配置中心

### db-spring-boot-starter（数据库支撑）

#### 包含项目中的依赖

base-spring-boot-starter

log-spring-boot-starter

### redis-spring-boot-starter（redis支撑）

#### 包含项目中的依赖

base-spring-boot-starter

#### 序列化反序列化

#### 分布式锁

#### 类

### uaa-client-spring-boot-starter（资源服务器）

资源服务器工具包

#### 基本概念

### uaa-server-spring-boot-starter（认证服务器）

认证服务器工具包

包含uaa-client-spring-boot-starter模块。

#### oauth2 客户端应用信息处理

在RedisClientDetailsService这个类中处理

#### 认证服务器配置

UAAServerConfig这个类处理

#### security配置

配置多用户插件工厂，登录处理器。

SecurityConfig这个类处理

#### 默认模式配置

### oss-spring-boot-starter

#### 基本概念

## iot-center

物联网中心

## job-center

基于nacos改造的xxl-job调度中心

## monitor-center

监控中心

## new-api-gateway

spring cloud gateway网关

## oauth-center

oauth鉴权中心

### auth-server

认证中心服务

包含uaa-server-spring-boot-starter这个模块

## register-center

注册中心

## tuning-center

## web-portal

web门户

# 服务

## ApiGateWayApp（网关服务）

### 包含的依赖

### 类

#### CustomAuthenticationManager

## AuthServerApp

认证服务

## BackCenterApplication

后台中心服务

## OperatCenterApp

运营中心服务

## UserCenterApp

用户中心服务

## BaseCenterApp

基础中心服务

# JavaMelody

JavaMelody是一个用于监控Java应用程序性能的开源库。它提供了一个简单而强大的方式来监测和分析Java应用程序在生产环境中的运行情况。JavaMelody可以集成到Java应用程序中，并在运行时收集各种性能指标。它监控应用程序的各个方面，包括HTTP请求、SQL查询、方法调用等。通过收集和记录这些数据，JavaMelody可以帮助开发人员识别潜在的性能问题和瓶颈，以及优化应用程序的性能。JavaMelody提供了一个基于Web的用户界面，可以通过浏览器访问。在该界面上，您可以查看请求的响应时间、服务调用次数、数据库查询情况等详细信息。此外，JavaMelody还提供了实时监控、报警功能，可以帮助您及时发现和解决性能问题。除了性能监测之外，JavaMelody还提供了一些其他有用的功能，例如内存使用情况、线程情况和缓存使用情况的监控。它还可以与其他日志和监控工具集成，以便更全面地监测和分析应用程序的性能。

总之，JavaMelody是一个功能丰富的监控工具，可以帮助开发人员实时跟踪和优化Java应用程序的性能，提高应用程序的稳定性和性能。

# 获取当前登录的用户

SysUser user = UserUtils.getCurrentUser(request, true);

# 类

## PageResult

### 作用

分页实体类

## TenantContextHolder

作用

## CustomerWebExceptionHandler

### 作用

自定义异常处理

## CustomAuthenticationManager

### 作用

### 说明

先执行CustomAuthenticationManager类中的代码，然后再执行CustomerWebExceptionHandler类中的代码。

