# Guns

## 1 简介

* 1. 运行

注意更改数据库连接

Main方法：执行启动main方法

Jar包(pom文件中打包方式设置为jar)：

打包: clean package –Dmaven.test.skip=true

运行jar包：java –jar 包名

War包(pom文件中打包方式设置为war)：

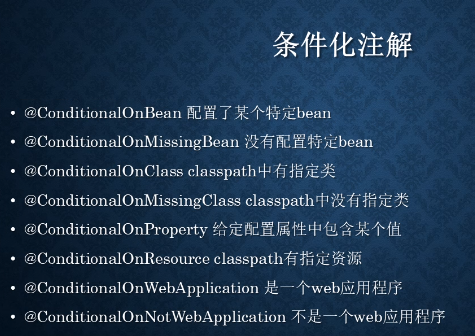
打包: clean package –Dmaven.test.skip=true

运行war包:将war包放置在tomcat安装目录webapps文件夹下。启动tomcat

## springboot

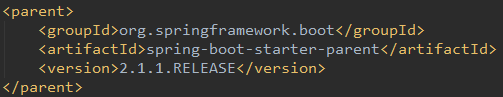
### conditionalOnXXX 注解

conditionalOnXXX注解，根据注解设置的条件来判断是否激活spring的某些配置代码。

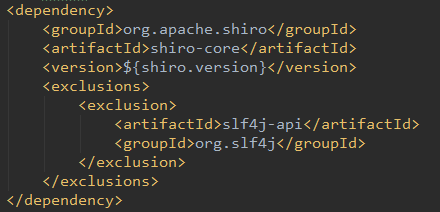


### 起步依赖

配置parent版本，springboot-starter的默认版本



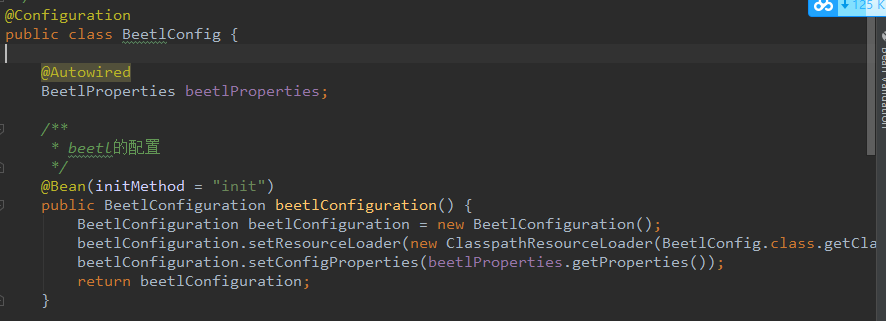
通过exclusions对springboot默认依赖进行扩展。排除自带的部分依赖，然后重新引用该依赖的其他版本。



### 2.3 添加自定义配置

在config包下创建配置类，使用@Configuration注解标记这个类。

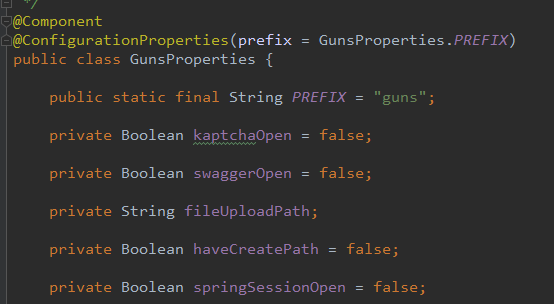
然后在类中定义方法，并用@Bean注解标记该方法，代表该bean被放置spring容器中。

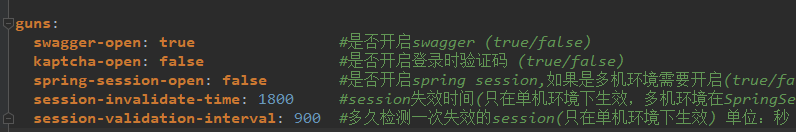


### 2.4 配置文件外置配置



将配置文件中的配置，与java类一一对应。





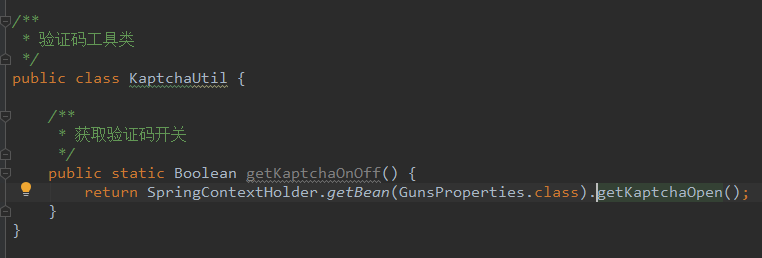
会自动将转化为驼峰形式。

使用方法：

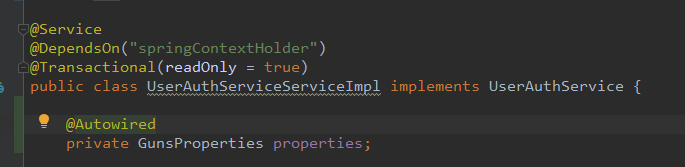
1 首先从spring容器中获取到这个bean。

2 然后在使用这个bean中所包含的属性。

由于该util类并没有被@Component注解标记，所以需要使用下面的方法获取bean，然后调用方法。



被注解标记的类，代表该类已在spring容器当中个，可以直接使用@Autowired注解直接将properties类的bean注入进来，直接使用该bean中的属性或方法。



### 2.5 spring profile



### 2.6 定制错误页面





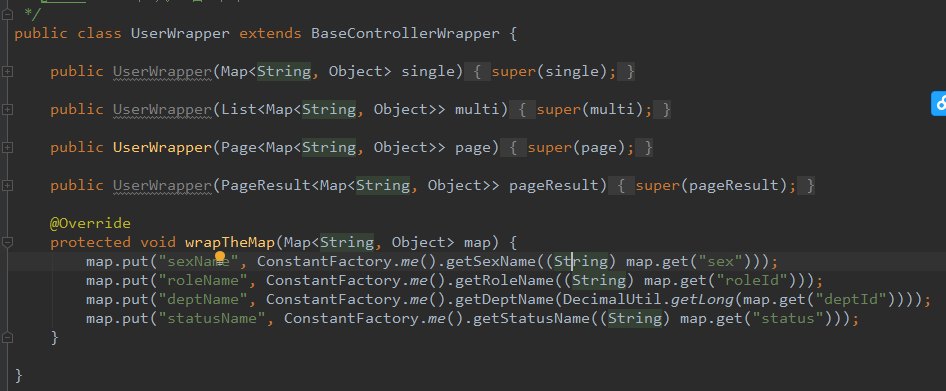


## map+wrapper使用

### 3.1 map+wrapper 具体用法



自定义wrapper类，继承BaseControllerWrapper类。重写wrapTheMap方法，在该方法中添加字段属性的key,value。



在controller中，如果返回结果是list，用List<Map<String,Object>>来接收。如果不是list，则用Map<String,Object>来接收。

如果返回结果不是这种形式，例如返回结果为User users=null;

可以不修改返回结果类型，通过

BeanKit.beanToMap(users)该方法可以将返回结果转换为map形式。

Map<String,Object> objectToMap=**BeanKit.beanToMap**(users);然后将转换返回的结果objectToMap作为返回方法的参数进行传递即可。（return new XXXWrapper(objectToMap).wrap()）

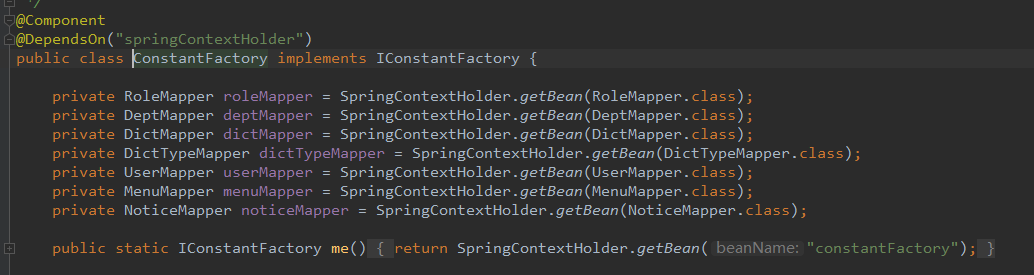
最终都用return **new XXXWrapper(users).wrap()**

### 3.2 guns中学习

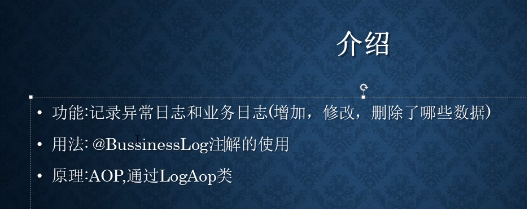
@DependsOn控制bean加载顺序

**A依赖于B，需要B首先被初始化。在A类上添加注解@DependsOn**

**可以在bean A上使用@DependsOn注解，告诉容器bean B应该先被初始化。**



## 4 日志系统



通过@BussinessLog注解来标记需要记录哪些方法的日志，该注解必须使用在controller层。