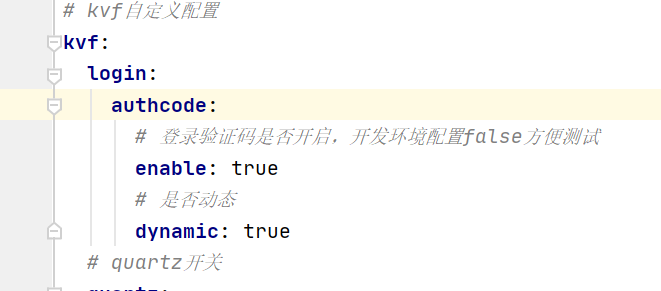
# Spring注解

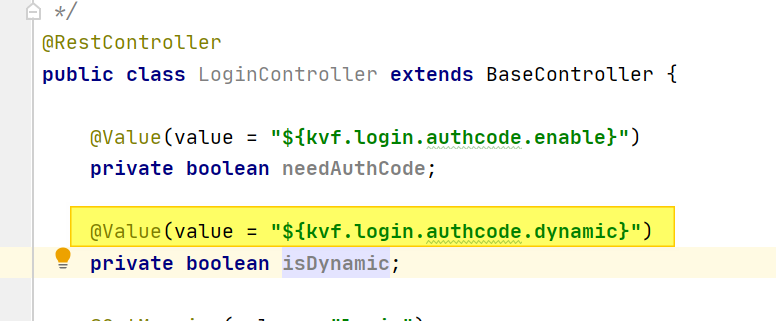
## @Value

读取application.yml中变量



代码： @Value注解

org.springframework.beans.factory.annotation.Value



# SpringBoot注解

@ConditionalOnProperty

@ConditionalOnProperty(name = **"kvf.cache.enable"**, havingValue = **"true"**)

通过其两个属性name以及havingValue来实现的，

name：从application.properties中读取某个属性值。

如果该值为空，则返回false;

如果值不为空，则将该值与havingValue指定的值进行比较，如果一样则返回true; 否则返回false。如果返回值为false，则该configuration不生效；为true则生效。

# 待整理

属性注入分为普通的属性注入和类型安全的属性注入。

普通的属性注入,主要是使用@Value注解实现 自定义 book.properties 文件

book.name=三国演义

book.author=罗贯中

book.id=1

定义实体类，配合lombok自动生成getter和setter方法，实现注入默认值

@Data //生成getter,setter等函数

@AllArgsConstructor //生成全参数构造函数

@NoArgsConstructor //生成无参构造函数

@Component

@PropertySource("classpath:book.properties")

public class Book {

@Value("${book.id}")

private Long id;

@Value("${book.name}")

private String name;

@Value("${book.author}")

private String author;

}