SpringBoot笔记

# 第一章 基础类

## 一 Date类相关问题

1、java.text.DateFormat:是日期/时间格式化子类的抽象类

1.1 作用：格式化（也就是日期 —> 文本）、解析（文本 —> 日期）

1.2 成员方法:

1. String format(Date date) 按照指定的模式,把Date日期,格式化为符合模式的字符串。
2. Date parse(String source) 把符合模式的字符串,解析为Date日期。

1.3 DateFormat类是一个抽象类,无法直接创建对象使用,可以使用DateFormat类的子类：java.text.SimpleDateFormat extends DateFormat。

(1) 构造方法：SimpleDateFormat(String pattern)：用给定的模式和默认语言环境的日期格式符号构造 SimpleDateFormat。

参数：String pattern:传递指定的模式

模式：区分大小写的

y 年

M 月

d 日

H 时

m 分

s 秒

写对应的模式,会把模式替换为对应的日期和时间

"yyyy-MM-dd HH:mm:ss"

注意:

模式中的字母不能更改,连接模式的符号可以改变

"yyyy年MM月dd日 HH时mm分ss秒"

**【测试代码】日期格式化以及日期的比较：**

public static void main(String[] args) {

SimpleDateFormat dateFormat= new SimpleDateFormat("yyyy-MM-dd hh:mm:ss");

Calendar calendar = Calendar.*getInstance*();

calendar.setTime(new Date());

calendar.add(Calendar.*DATE*,-15);

Date date = calendar.getTime();

System.*out*.println("Format-currentDate: " + dateFormat.format(date));

System.*out*.println("Date-currentDate: " + date);

System.*out*.println("Date-currentDate.getTime: " + date.getTime());

String time1 = "2022-10-25 18:55:58";

try {

Date parseTime1 = dateFormat.parse(time1);

System.*out*.println("time1: " + parseTime1);

System.*out*.println("time1.getTime: " + parseTime1.getTime());

System.*out*.println("date在time1前 : " + (date.getTime() < parseTime1.getTime()));

} catch (ParseException e) {

e.printStackTrace();

}

}

## 二 注解

### 1、@Controller

@Controller标识的类，该类代表控制器类(控制层/表现层)。这里控制层里面的每个方法，都可以去调用@Service标识的类（业务逻辑层）。

@Service标识的类中的方法可以继续调用@Resposity标识的接口实现类（Dao层/持久层）

MVC其实我们可以理解为是VCM：JSP/HTML页面数据–>控制器–>数据逻辑处理并持久化。

## 三 pom文件

### 1、<packaging>pom</packaging>

在父级项目中的pom.xml文件使用的packaging配置一定为pom。父级的pom文件只作项目的子模块的整合，在maven install时不会生成jar/war压缩包。

## 四 其他

### 1、IDEA .iml文件及.idea文件夹

#### 1.1 .iml文件

idea 对module 配置信息之意， infomation of module。每个模块都有一个iml文件。IDEA中的.iml文件是项目标识文件，缺少了这个文件，IDEA就无法识别项目。跟Eclipse的.project文件性质是一样的。并且这些文件不同的设备上的内容也会有差异，所以我们在管理项目的时候，.project和.iml文件都需要忽略掉。

iml文件是IntelliJ IDEA自己创建的模块文件，用于java应用开发，存储一些模块相关的信息，比如一个Java组件，插件组件，Maven组件等。存储一些模块路径信息，依赖信息以及别的一些设置。

#### 1.2 .idea文件

存放项目的配置信息，包括数据源，类库，项目字符编码，历史记录，版本控制信息等。

（1）Artifacts（人工制品，手工制品，手工艺品）是maven中的一个概念，表示某个module要如何打包，例如war exploded、war、jar、ear等等这种打包形式；

（2）inspectionProfiles：检查配置文件

（3）jsLinters：js语法检查配置文件

（4）libraries：类库

（5）.gitignore：git提交忽略文件

（6）misc.xml：各种杂项

（7）modules.xml：模块

（8）vcs.xml：版本控制系统

（9）workspace.xml：工作空间。保存着一下项目开发过程中的过程性信息，可以不要。