**关于前后台的时间格式数据传递问题**

1. **时间格式**

后台的时间对象Date date = new Date();

前端的时间控件type=date;

这两者之间我们要如何建立起前后台交互，例如我们在页面上新增数据的时候，时间属性通过前端控件传到后台是一串毫秒数，而后台则是一个Date对象，所以无法进行正常接收。

而我们早就意识到了这个问题，当然也提出了我们自己的解决方案，那就是将后台bean对象中的Date对象java.util.Date修改为java.sql.Date。其实还是有些不理解这两者之间的区别，但是我们知道的结果就是修改完之后就能正常从前端接收到那一串毫秒数并转义成java.sql.Date的正确时间格式，也能通过底层数据库操作正常插入到数据库中。

从字面上看，第一个java.util.Date，是java自带的工具类包，而第二个java.sql.Date是关于SQL方面的包。通过查阅博客，有了一个更直观的比较：

java.util.Date 就是在除了SQL语句的情况下面使用java.sql.Date 是针对SQL语句使用的，它只包含日期而没有时间部分

虽然还是没有理解为什么从util包修改为sql包就能够正常接收到前端时间控件的毫秒数，但是至少我们直观地理解了这两者之间的区别。一句话来概括就是，尽量都使用util包下的Date对象。

那我们如果要使用util包下的时间对象，那我们该如何实现接收前端传递的时间毫秒数？

**二、@DateTimeFormat(pattern = "yyyy-MM-dd")**

为了解决前文中提出来的问题，这就引入了副标题的注解。在框架中，我们总是追求以最少的代码实现更强大的功能，于是就避免不了注解的使用。该注解是org.springframework.format.annotation.DateTimeFormat，即spring架构下的包。

具体使用也很简单，只要找到你创建bean的地方，例如你有一个A对象，里面有一个private Date time;这一个属性，我们还是使用基础的java.util.Date，但是为了正常格式化成我们想要的时间格式，在该属性上方加@DateTimeFormat注解，括号中加上你具体要转义的格式，例如年月日就是pattern=”yyyy-MM-dd”，年月日时分秒就是pattern=”yyyy-MM-dd HH-mm-ss”.

用了该注解之后，我们再从前端接收时间毫秒数，传递到后台的这个time属性就能够正常转义了。具体如下图所示：



**三、总结**

框架中的实现让你觉得简单，只是因为设计框架的人将复杂过程进行了封装。而我一直秉持着不管简单还是复杂，解决了问题的方法就是好方法。只不过我们有时候应该有一颗深究的心，为什么这个能实现，是怎么实现的，跟之前的实现相比较能够得到什么好处。

框架也无非只是一个方便我们使用的工具，多翻翻底层源码对以后的工作有帮助。