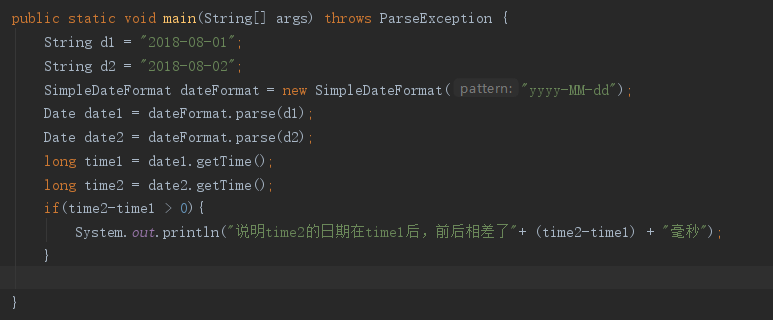
**日期比较处理**

1. **日期比较问题**

从前台对于插入编辑时间传入后台的处理，详情可见关于前后台时间属性交互.docx。

我们解决了插入与编辑时间的问题，那在面对日期筛选的问题，我们又该如何解决呢。首先，抛开业务情况，我们在java后台代码中的实现如图所示：



无非就是将Date对象调用一下getTime()方法，转换为long型的毫秒数，然后将比较日期巧妙地转换成了比较毫秒数的大小即可确定日期先后顺序。

但是在结合业务情况的基础上，例如我想要查询2018-08-01~2018-08-31之间的数据，我难道要从数据库里查询出来的每一个数据列表在后台代码中进行二次判断是否符合筛选要求，这显然是不合理的，所以我们应该将从前台获取到的两个Date对象传入到底层的mybatis的sql语句中进行判断。

1. **具体实现**

<if test="beginDate != null and beginDate != ''">AND

beginDate<![CDATA[>=]]> #{beginDate}</if>

<if test="endDate != null and endDate != ''">AND endDate<![CDATA[<=]]>#{endDate}</if>

在xml文件中的查询语句中加入这两个if判断即可实现筛选一个时间段内的数据。需要注意的是，<![CDATA[<=]]>的使用是因为在使用mybatis 时我们sql是写在xml 映射文件中，如果写的sql中有一些特殊的字符的话，在解析xml文件的时候会被转义，但我们不希望他被转义，所以我们要使用<![CDATA[ ]]>来解决，或者我们对于> <的处理也可以写成&gt; &lt;

即以上的if片段还可以这样写：

<if test="beginDate != null and beginDate != ''">

AND beginDate &gt;=> #{beginDate}</if>

<if test="endDate != null and endDate != ''">

AND endDate &lt;=>#{endDate}</if>

备注：以上的beginDate/endDate都需要是一个时间类型才能够直接在mybatis中进行比较，但是当前没有一个具体的业务需求让我测试，希望后面可以有机会测试一下。

1. **总结**

我们作为java后台开发工程师，总是难免在第一时间想要通过java代码来实现业务逻辑，但事实上，我们可以让其他语言来帮助我们解决问题，例如SQL语句。

在很久以前，我觉得SQL语句很简单，但事实上还有很多复杂的SQL语句和函数能够替代我们在java后台中书写的大段代码，包括我们目前接触甚少的多表联查语句以及建表时候的规范性问题都需要不断地学习。不要被语言给局限了，每一种语言都只是一种解决方案而已。