链滴

# Java8 LocalDate 和 LocalDateTime

作者：muzeay
原文链接：https://ld246.com/article/1590456671293
来源网站：链滴

**为什么我们需要新的Java Date Time API?**

在我们开始查看Java 8 Date Time API之前，让我们看看为什么我们需要一个新的API。java中现有日期和时间相关类存在一些问题，其中一些是：

1. Java Date Time类没有一致定义，我们在包 java.util和 java.sql包中都有Date Class 。再次格化和解析类在 java.text包中定义。

2. java.util.Date包含日期和时间，而 java.sql.Date仅包含日期。java.sql包装中的这个没有意义。这个类都有相同的名称，这本身就是一个非常糟糕的设计。

3. 时间，时间戳，格式和解析没有明确定义的类。我们有 java.text.DateFormat用于解析和格式需求的抽象类。通常 SimpleDateFormat类用于解析和格式化。

4. 所有Date类都是可变的，因此它们 **不是线程安全的**。这是Java Date和Calendar类的最大问题之。

5. 日期类不提供国际化，没有时区支持。所以 java.util.Calendar和 java.util.TimeZon类进行了介，但他们也有上面列出的所有问题。

**总结起来一句话，新的Java Date Time API更安全。**

---

**Jdk8改革**

LocalDateTime获取毫秒数

```
//获取秒数
Long second = LocalDateTime.now().toEpochSecond(ZoneOffset.of("+8"));
//获取毫秒数
```

原文链接：Java8 LocalDate 和 LocalDateTime

```
Long milliSecond = LocalDateTime.now().toInstant(ZoneOffset.of("+8")).toEpochMilli();
```

LocalDateTime与String互转

```
//时间转字符串格式化
 DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyyMMddHHmmssSSS");
 String dateTime = LocalDateTime.now(ZoneOffset.of("+8")).format(formatter);
```

字符串转时间

```
String dateTimeStr = "2018-07-28 14:11:15";
DateTimeFormatter df = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss");
LocalDateTime dateTime = LocalDateTime.parse(dateTimeStr, df);
```

Date与LocalDateTime互转

```
    //将java.util.Date 转换为java8 的java.time.LocalDateTime,默认时区为东8区
    public static LocalDateTime dateConvertToLocalDateTime(Date date) {
        return date.toInstant().atOffset(ZoneOffset.of("+8")).toLocalDateTime();
    }


    //将java8 的 java.time.LocalDateTime 转换为 java.util.Date，默认时区为东8区
    public static Date localDateTimeConvertToDate(LocalDateTime localDateTime) {
        return Date.from(localDateTime.toInstant(ZoneOffset.of("+8")));
    }
```

将LocalDateTime转为自定义的时间格式的字符串

```
public static String getDateTimeAsString(LocalDateTime localDateTime, String format) {
    DateTimeFormatter formatter = DateTimeFormatter.ofPattern(format);
    return localDateTime.format(formatter);
}
```

将某时间字符串转为自定义时间格式的LocalDateTime

```
public static LocalDateTime parseStringToDateTime(String time, String format) {
    DateTimeFormatter df = DateTimeFormatter.ofPattern(format);
    return LocalDateTime.parse(time, df);
}
```

将long类型的timestamp转为LocalDateTime

```
public static LocalDateTime getDateTimeOfTimestamp(long timestamp) {
    Instant instant = Instant.ofEpochMilli(timestamp);
    ZoneId zone = ZoneId.systemDefault();
    return LocalDateTime.ofInstant(instant, zone);
}
```

将LocalDateTime转为long类型的timestamp

```
public static long getTimestampOfDateTime(LocalDateTime localDateTime) {
    ZoneId zone = ZoneId.systemDefault();
    Instant instant = localDateTime.atZone(zone).toInstant();
    return instant.toEpochMilli();
```

原文链接：Java8 LocalDate 和 LocalDateTime

```
}
```

**对于这个新API，一些最常用的类将是LocalDate和LocalDateTime。**
**非常容易使用，并且具有类似的方法，可以轻松找到特定的工作。**

原文链接：Java8 LocalDate 和 LocalDateTime