

時間相關

Java提供了許多處理日期和時間的類和方法。以下是一些基本的日期操作：

java.util.Date

創建日期對象

```
Date currentDate = new Date();
```

java.text.SimpleDateFormat

格式化日期

可以使用 `java.text.SimpleDateFormat` 類將日期格式化為特定的字符串格式。例如：

```
SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");  
String dateString = dateFormat.format(currentDate);
```

解析日期字符串

可以使用 `SimpleDateFormat` 類將字符串轉換為日期對象。例如：

```
String dateString = "2023-03-23";  
SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");  
Date date = dateFormat.parse(dateString);
```

java.util.Calendar

操作日期

可以使用 `java.util.Calendar` 類進行日期計算，例如添加或減去一定的時間。例如：

這將當前日期添加7天，得到一個未來的日期。

這僅是Java中處理日期的基本操作。還有許多其他的類和方法可以用來處理日期和時間。

```
Calendar calendar = Calendar.getInstance();  
calendar.setTime(currentDate);  
calendar.add(Calendar.DAY_OF_MONTH, 7);  
Date futureDate = calendar.getTime();
```

找出前五分鐘的日期

```
// 創建一個 Calendar 實例  
Calendar calendar = Calendar.getInstance();  
  
// 從當前日期和時間中減去 5 分鐘  
calendar.add(Calendar.MINUTE, -5);  
  
// 得到剛才減去的日期時間  
Date fiveMinutesAgo = calendar.getTime();  
  
// 格式化日期時間，如果需要的話  
SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");  
String fiveMinutesAgoString = dateFormat.format(fiveMinutesAgo);  
  
// 輸出結果  
System.out.println("前五分鐘的日期時間為：" + fiveMinutesAgoString);
```

寫一個各種常用時間日期字串轉換成日期的方法

需要注意的是，轉換時需要確保時間文字的格式與使用的 `SimpleDateFormat` 格式一致，否則會導致轉換失敗。如果轉換失敗，則會拋出 `ParseException` 異常。

```
public static Date parseDateTimeString(String dateTimeString) {
    // 定義各種常用的時間日期格式
    Map<String, String> dateTimeFormats = new HashMap<>();
    dateTimeFormats.put("yyyy-MM-dd", "\\d{4}-\\d{2}-\\d{2}$");
    dateTimeFormats.put("yyyy-MM-dd'T'HH:mm:ss", "\\d{4}-\\d{2}-\\d{2}T\\d{2}:\\d{2}:\\d{2}$");
    dateTimeFormats.put("yyyy-MM-dd'T'HH:mm:ss.SSS", "\\d{4}-\\d{2}-\\d{2}T\\d{2}:\\d{2}:\\d{2}.\\d{3}$");
    dateTimeFormats.put("yyyy-MM-dd'T'HH:mm:ss'Z'", "\\d{4}-\\d{2}-\\d{2}T\\d{2}:\\d{2}:\\d{2}Z$");
    dateTimeFormats.put("yyyy-MM-dd'T'HH:mm:ss.SSS'Z'", "\\d{4}-\\d{2}-\\d{2}T\\d{2}:\\d{2}:\\d{2}.\\d{3}Z$");
    dateTimeFormats.put("EEE MMM dd HH:mm:ss zzz yyyy", "\\w{3} \\w{3} \\d{2} \\d{2}:\\d{2}:\\d{2} \\w{3} \\d{4}$");

    // 使用各種格式進行轉換，直到轉換成功為止
    for (Map.Entry<String, String> entry : dateTimeFormats.entrySet()) {
        String format = entry.getKey();
        String regex = entry.getValue();
        if (dateTimeString.matches(regex)) {
            try {
                SimpleDateFormat dateFormat = new SimpleDateFormat(format, Locale.ENGLISH);
                Date date = dateFormat.parse(dateTimeString);
                return date;
            } catch (ParseException e) {
                // 轉換失敗，繼續使用下一個格式進行轉換
            }
        }
    }

    // 如果所有格式都無法轉換，則返回null
    return null;
}
```

好用時間Lib

在Java中，有許多好用的時間相關的函式庫，以下是一些常見且經常使用的時間函式庫推薦：

1. Joda-Time：Joda-Time是一個廣泛使用的Java日期和時間庫，它提供了比Java內置日期時間類庫更多的功能。Joda-Time包含各種類型的日期，時間和時間間隔，並提供了許多便捷的方法來進行日期計算和格式化。
2. java.time：Java 8引入了新的時間API，稱為java.time。它提供了新的日期時間類型，例如LocalDate，LocalTime和ZonedDateTime等，並提供了許多便於使用的方法來進行日期計算和格式化。
3. Date4j：Date4j是一個輕量級的日期和時間庫，它具有方便的方法和較小的庫大小。Date4j支持基本日期和時間計算，例如計算兩個日期之間的天數，以及格式化日期時間。
4. PrettyTime：PrettyTime是一個Java函式庫，用於將日期和時間轉換為易於閱讀的相對時間（例如“3天前”）。它支持多種語言和格式，可以輕鬆地集成到Java應用程序中。

🕒 修訂版本 #2

★ 由 treeman 建立於 23 🕒🕒🕒 2023 09:56:39

🔧 由 treeman 更新於 5 🕒🕒🕒 2023 10:18:39