



链滴

SpringBoot 时间类型字段指定返回格式

作者: [zhengliwei](#)

原文链接: <https://ld246.com/article/1600612066743>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

hello, 大家好, 欢迎来到银之庭。我是Z, 一个普通的程序员。今天我们来看一下如何在SpringBoot 目中, 指定时间类型字段的返回格式。

1. SpringBoot返回的默认格式

在不指定任何特殊配置的情况下, 返回的date类型的字段会自动转成UTC格式字符串, 形如2020-09-0T13:51:34.050+00:00。

2. 指定全局返回时间戳

在application.yml中增加如下配置, 可指定全局把date类型字段转换成时间戳返回, 时间戳为指定间距1970-01-01 00:00:00的毫秒数。

```
spring:
  jackson:
    serialization:
      write-dates-as-timestamps: true
```

3. 指定全局返回的时间格式

在application.yml中增加如下配置, 可指定全局date类型字段的返回格式。

```
spring:
  jackson:
    date-format: yyyy-MM-dd HH:mm:ss
    time-zone: GMT+8
```

4. 指定单个字段返回的时间格式

我们还可以只指定某个字段的返回格式, 只需要在该字段上加上com.fasterxml.jackson.annotation.JsonFormat注解, 并指定pattern和timezone即可, 示例如下:

```
package top.zhengliwei.timeFormat.model;

import com.fasterxml.jackson.annotation.JsonFormat;
import lombok.Builder;
import lombok.Data;

import java.util.Date;

@Data
@Builder
public class TimeResponseDTO {

    @JsonFormat(pattern = "yyyy/MM/dd HH:mm:ss", timezone = "GMT+8")
    private Date nowTime;

}
```

5. 指定单个字段序列化器

我们还可以指定某些字段在序列化为json时使用我们自定义的json序列化器。springboot默认使用jackson作为json序列化和反序列化组件，所以我们只要按照jackson规范实现自定义序列化器并使用即可。

5.1 编写自定义json序列化器

自定义json序列化器需要继承`com.fasterxml.jackson.databind.JsonSerializer`抽象基类，并重写抽方法`serialize`。一个自定义的时间类型序列化器如下：

```
package top.zhengliwei.timeFormat.util;

import com.fasterxml.jackson.core.JsonGenerator;
import com.fasterxml.jackson.core.JsonProcessingException;
import com.fasterxml.jackson.databind.JsonSerializer;
import com.fasterxml.jackson.databind.SerializerProvider;

import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Date;

public class DateSerializer extends JsonSerializer<Date> {

    @Override
    public void serialize(Date value, JsonGenerator gen, SerializerProvider serializers)
        throws IOException, JsonProcessingException {
        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

        String dateStr = dateFormat.format(value);
        if (!"2000-01-01 00:00:00".equals(dateStr)) {
            gen.writeString(dateStr);
        } else {
            gen.writeString("");
        }
    }
}
```

5.2 使用自定义json序列化器

在我们需要使用这个序列化器的字段上用`com.fasterxml.jackson.databind.annotation.JsonSerialize`注解，指定序列化器的类即可，如下：

```
package top.zhengliwei.timeFormat.model;

import com.fasterxml.jackson.databind.annotation.JsonSerialize;
import lombok.Builder;
import lombok.Data;
import top.zhengliwei.timeFormat.util.DateSerializer;

import java.util.Date;

@Data
@Builder
```

```
public class TimeResponseDTO {  
  
    @JsonSerialize(using = DateSerializer.class)  
    private Date nowTime;  
  
}
```