Evan
Only let oneself become strong enough, good enough, can afford the life that you want to.

SpringMVC+MyBatis 返回时间格式转换的解决方案

标签： spring mvc mybatis 时间 格式转换

2016-11-03 20:13 4437人阅读 评论(0) 收藏 举报

分类： web后台（12）

Spring MVC 4.X ResponseBody 日期类型Json 处理

摘自http://tramp-zzy.iteye.com/blog/2090330 2014-07-10

方法一：全局处理

```
<mvc:annotation-driven>
    <!-- 处理responseBody 里面日期类型 -->
    <mvc:message-converters>
        <bean class="org.springframework.http.converter.json.MappingJackson2HttpMessageConverter">
            <property name="objectMapper">
                <bean class="com.fasterxml.jackson.databind.ObjectMapper">
                    <property name="dateFormat">
                        <bean class="java.text.SimpleDateFormat">
                            <constructor-arg type="java.lang.String" value="yyyy-MM-dd HH:mm:ss" />
                        </bean>
                    </property>
                </bean>
            </property>
        </bean>
    </mvc:message-converters>
</mvc:annotation-driven>
```

方法二：局部注解处理

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

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

```
/**
 * ClassName:DateJsonSerializer <br/>
 * Function: 日期类型格式化，格式化为：yyyy-MM-dd HH:mm:ss 格式. 用法如下：<br/>
 * Reason:   @JsonSerialize(using=DateJsonSerializer.class)
 *          @Column(name="BIRTHDAY")
 *          public Date getBirthday() {
 *            return birthday;
 *          }
 *         . <br/>
 * Date:     2014年7月10日 下午1:26:08 <br/>
 * @author   zhangzhaoyu
 * @version  1.0
 * @since    JDK 1.7
 * @see
 */
public class DateJsonSerializer extends JsonSerializer<Date> {

   @Override
   public void serialize(Date value, JsonGenerator jgen,
         SerializerProvider provider) throws IOException,
         JsonProcessingException {
      SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
      String formattedDate = formatter.format(value);
      jgen.writeString(formattedDate);
   }

}
```

方法三：jackson 注解处理

@JsonIgnoreProperties

此注解是类注解，作用是json序列化时将java bean中的一些属性忽略掉，序列化和反序列化都受影响。

@JsonIgnore

此注解用于属性或者方法上（最好是属性上），作用和上面的@JsonIgnoreProperties一样。

@JsonFormat

此注解用于属性或者方法上（最好是属性上），可以方便的把Date类型直接转化为我们想要的模式，比如@JsonFormat(pattern = "yyyy-MM-dd HH-mm-ss")

@JsonSerialize

```
// 反序列化一个固定格式的Date
    @JsonDeserialize(using = CustomDateDeserialize.class)
    public void setBirthday(Date birthday) {
        this.birthday = birthday;
    }

// 序列化指定格式的double格式
    @JsonSerialize(using = CustomDoubleSerialize.class)
    public double getSalary() {
        return salary;
    }

public class CustomDateDeserialize extends JsonDeserializer<Date> {

    private SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

    @Override
    public Date deserialize(JsonParser jp, DeserializationContext ctxt)
            throws IOException, JsonProcessingException {

        Date date = null;
        try {
            date = sdf.parse(jp.getText());
        } catch (ParseException e) {
            e.printStackTrace();
        }
        return date;
    }
}
```

spring mvc4使用及json 日期转换解决方案

摘自:http://blog.csdn.net/zhanngle/article/details/24123659 2014-04-19

又到搭新开发环境的时候，总是不免去网上搜下目前最新的框架。spring是web开发必用的框架，于是乎下载了目前最新的spring4.0.3，同时越来越不想用struts2，想试试spring mvc，也将spring-webmvc4.0.3下了下来，投入两天时间学习后，发现还是挺优雅的，特别是从3.0后，spring mvc使用注解方式配制，以及对rest风格的支持，真是完美致极。

下面将这两天研究到的问题做个总结，供参考。

1.request对象的获取

方式1：在controller方法上加入request参数，spring会自动注入，如：

```
public String list(HttpServletRequest request,HttpServletResponse response)
```

方式2：在controller类中加入@Resource private HttpServletRequest request 属性，spring会自动注入，这样不知道会不会出现线
controller实例会为多个请求服务，暂未测试。

方式3：在controller方法中直接写代码获取

```
HttpServletRequest request = ((ServletRequestAttributes)RequestContextHolder.getRequestAttributes()).getRequest();
```

方式4：在controller中加入以下方法，此方法会在执行此controller的处理方法之前执行

```
@ModelAttribute
private void initServlet(HttpServletRequest request,HttpServletResponse response) {
    //String p=request.getParameter("p");
    //this.req=request;//实例变量，有线程安全问题，可以使用ThreadLocal模式保存
}
```

2.response对象的获取

可以参照以上request的获取方式1和方式4，方式2和方式3对response对象无效！

3.表单提交之数据填充

直接在方法上加入实体对象参数，**spring**会自动填充对象中的属性，对象属性名要与<input>的name一致才会填充.

如：public boolean doAdd(Demo demo)

4.表单提交之数据转换-Date类型

在实体类的属性或get方法上加入 @DateTimeFormat(pattern="yyyy-MM-dd HH:mm:ss")，那么表单中的日期字符串就会正确的转换为Date类型了。

还有@NumberFormat注解，暂时没用，就不介绍了，一看就知道是对数字转换用的。

5.json数据返回

在方法上加入@ResponseBody，同时方法返回值为实体对象，spring会自动将对象转换为json格式，并返回到客户端。如下所示：

```
@RequestMapping("/json1")
@ResponseBody
public Demo json1() {
    Demo demo=new Demo();
    demo.setBirthday(new Date());
    demo.setCreateTime(new Date());
    demo.setHeight(170);
    demo.setName("tomcat");
    demo.setRemark("json测试");
    demo.setStatus((short)1);
    return demo;
}
```

注意：spring配置文件要加上：**<mvc:annotation-driven/>**，同时还要引入jackson-core.jar,jackson-databind.jar,jackson-annotations.jar(2.x的包)才会自动转换json

这种方式是spring提供的。**我们还可以自定义输出json，以上第二条不是说了获取response对象吗，拿到response对象后，任由开发人员宰割，想怎么返回就怎么返回。**

方法不要有返回值，如下：

```
@RequestMapping("/json2")
public void json2() {
    Demo demo=new Demo();
    demo.setBirthday(new Date());
    demo.setCreateTime(new Date());
    demo.setHeight(170);
    demo.setName("tomcat");
```

```
    demo.setRemark("json测试");
    demo.setStatus((short)1);
    String json=JsonUtil.toJson(obj);//;json处理工具类
    HttpServletResponse response = //获取response对象
    response.getWriter().print(json);
}
```

OK，一切很完美。接着恶心的问题迎面而来，**date类型转换为json字符串时，返回的是long time值**，如果你想返回“yyyy-MM-dd HH:mm:ss”格式的字符串，又要自定义了。我很奇怪，不是有@DateTimeFormat注解吗，为什么不利用它。难道@DateTimeFormat只在表单提交时，将字符串转换为date类型，而date类型转换为json字符串时，就不用了。带着疑惑查源码，**原来spring使用jackson转换json字符，而@DateTimeFormat是spring-context包中的类，jackson如何转换，spring不方便作过多干涉**，于是只能遵守jackson的转换规则，自定义日期转换器。

先写一个日期转换器，如下：

```
public class JsonDateSerializer extends JsonSerializer<Date> {
  private SimpleDateFormat dateFormat=new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
  @Override
  public void serialize(Date date, JsonGenerator gen, SerializerProvider provider)  throws IOException, JsonProcessingException {
     String value = dateFormat.format(date);
     gen.writeString(value);
  }
}
```

在实体类的get方法上配置使用转换器，如下：

```
@DateTimeFormat(pattern="yyyy-MM-dd HH:mm:ss")
@JsonSerialize(using=JsonDateSerializer.class)
public Date getCreateTime() {
   return this.createTime;
}
```

OK，到此搞定。

你真的满意了吗，这么不优雅的解决方案，假设birthday属性是这样的，只有年月日，无时分秒

```
@DateTimeFormat(pattern="yyyy-MM-dd")
public Date getBirthday() {
   return this.birthday;
}
```

这意味着，又要为它定制一个JsonDate2Serializer的转换器，然后配置上，像这样

```
@DateTimeFormat(pattern="yyyy-MM-dd")
@JsonSerialize(using=JsonDate2Serializer.class)
public Date getBirthday() {
   return this.birthday;
}
```

假设还有其它格式的Date字段，还得要为它定制另一个转换器。my god，请饶恕我的罪过，不要让我那么难受

经过分析源码，找到一个不错的方案，此方案将不再使用@JsonSerialize，而只利用@DateTimeFormat配置日期格式，jackson就可以正确转换，但@DateTimeFormat只能配置在get方法上，这也没什么关系。

先引入以下类，此类对jackson的ObjectMapper类做了注解扫描拦截，使它也能对加了@DateTimeFormat的get方法应用日期格式化规则

```
package com.xxx.utils;

import java.io.IOException;
import java.lang.reflect.AnnotatedElement;
import java.text.SimpleDateFormat;
import java.util.Date;
import org.springframework.format.annotation.DateTimeFormat;
import org.springframework.stereotype.Component;
import com.fasterxml.jackson.core.JsonGenerator;
import com.fasterxml.jackson.core.JsonProcessingException;
import com.fasterxml.jackson.databind.JsonSerializer;
import com.fasterxml.jackson.databind.ObjectMapper;
import com.fasterxml.jackson.databind.SerializerProvider;
import com.fasterxml.jackson.databind.introspect.Annotated;
import com.fasterxml.jackson.databind.introspect.AnnotatedMethod;
import com.fasterxml.jackson.databind.introspect.JacksonAnnotationIntrospector;

/**
 * json处理工具类
 * @author zhangle
 */
@Component
public class JsonUtil {
    private static final String DEFAULT_DATE_FORMAT="yyyy-MM-dd HH:mm:ss";
    private static final ObjectMapper mapper;

    public ObjectMapper getMapper() {
        return mapper;
    }

    static {
        SimpleDateFormat dateFormat = new SimpleDateFormat(DEFAULT_DATE_FORMAT);

        mapper = new ObjectMapper();
        mapper.setDateFormat(dateFormat);
        mapper.setAnnotationIntrospector(new JacksonAnnotationIntrospector() {
            @Override
            public Object findSerializer(Annotated a) {
                if(a instanceof AnnotatedMethod) {
                    AnnotatedElement m=a.getAnnotated();
```

```
                    DateTimeFormat an=m.getAnnotation(DateTimeFormat.class);
                    if(an!=null) {
                        if(!DEFAULT_DATE_FORMAT.equals(an.pattern())) {
                            return new JsonDateSerializer(an.pattern());
                        }
                    }
                }
                return super.findSerializer(a);
            }
        });
    }

    public static String toJson(Object obj) {
        try {
            return mapper.writeValueAsString(obj);
        } catch (Exception e) {
            throw new RuntimeException("转换json字符失败!");
        }
    }

    public <T> T toObject(String json,Class<T> clazz) {
        try {
            return mapper.readValue(json, clazz);
        } catch (IOException e) {
            throw new RuntimeException("将json字符转换为对象时失败!");
        }
    }

    public static class JsonDateSerializer extends JsonSerializer<Date>{
        private SimpleDateFormat dateFormat;
        public JsonDateSerializer(String format) {
            dateFormat = new SimpleDateFormat(format);
        }

        @Override
        public void serialize(Date date, JsonGenerator gen, SerializerProvider provider)
                throws IOException, JsonProcessingException {
            String value = dateFormat.format(date);
            gen.writeString(value);
        }
```

```
    }
}
```

再将<mvc:annotation-driven/>改为以下配置，配置一个新的json转换器，将它的ObjectMapper对象设置为JsonUtil中的objectMapp
比spring内置的json转换器优先级更高，所以与json有关的转换，spring会优先使用它。

```
<mvc:annotation-driven>
    <mvc:message-converters>
        <bean class="org.springframework.http.converter.json.MappingJackson2HttpMessageConverter">
            <property name="objectMapper" value="#{jsonUtil.mapper}"/>
            <property name="supportedMediaTypes">
                <list>
                    <value>text/json;charset=UTF-8</value>
                </list>
            </property>
        </bean>
    </mvc:message-converters>
</mvc:annotation-driven>
```

接下来就可以这样配置实体类，jackson也能正确转换Date类型

```
@DateTimeFormat(pattern="yyyy-MM-dd HH:mm:ss")
public Date getCreateTime() {
    return this.createTime;
}
@DateTimeFormat(pattern="yyyy-MM-dd")
public Date getBirthday() {
    return this.birthday;
}
```

完毕，一切都完美了。

以下为2014/4/21 补充
写了那么多，发现白忙活了一场，原来jackson也有一个@JsonFormat注解，将它配置到Date类型的get方法上后，jackson就会按照配置的格式转换日期类型，而不自定义转换器类，欲哭无泪啊。辛苦了那么多，其实别人早已提供，只是没有发现而已。
不说了，直接上方案吧。
1.spring配置照样是这样：

```
01. <mvc:annotation-driven>
```

2.JsonUtil可以不用了，但如果要自己从response对象输出json，那么还是可以用，但改成了这样

```
package com.xxx.utils;

import java.io.IOException;
import java.text.SimpleDateFormat;
import org.springframework.stereotype.Component;
import com.fasterxml.jackson.databind.ObjectMapper;

/**
 * json处理工具类
```

```
 * @author zhangle
 */
@Component
public class JsonUtil {

    private static final String DEFAULT_DATE_FORMAT="yyyy-MM-dd HH:mm:ss";
    private static final ObjectMapper mapper;

    static {
        SimpleDateFormat dateFormat = new SimpleDateFormat(DEFAULT_DATE_FORMAT);
        mapper = new ObjectMapper();
        mapper.setDateFormat(dateFormat);
    }

    public static String toJson(Object obj) {
        try {
            return mapper.writeValueAsString(obj);
        } catch (Exception e) {
            throw new RuntimeException("转换json字符失败!");
        }
    }

    public <t> T toObject(String json,Class<t> clazz) {
        try {
            return mapper.readValue(json, clazz);
        } catch (IOException e) {
            throw new RuntimeException("将json字符转换为对象时失败!");
        }
    }
}
```

3.实体类的get方法就需要多一个@JsonFormat的注解配置

```
@DateTimeFormat(pattern="yyyy-MM-dd HH:mm:ss")
@JsonFormat(pattern="yyyy-MM-dd HH:mm:ss",timezone = "GMT+8")
public Date getCreateTime() {
return this.createTime;
}
@DateTimeFormat(pattern="yyyy-MM-dd")
@JsonFormat(pattern="yyyy-MM-dd",timezone = "GMT+8")
public Date getBirthday() {
```

```
    return this.birthday;
}
```


- 上一篇 mybatis中association和collection的column传入多个参数值
- 下一篇 田维经典语录（一）

相关文章推荐

- mysql 获取当前日期及格式化
- Mybatis中关于时间的处理
- [Mybatis + Oracle] 格式化时间
- Mybatis从数据库中取日期类型数据的方法
- JavaEE_Mybatis_SpringMVC SpringMVC日期类...
- SpringMVC配置全局日期转换器，处理日期转换...
- springmvc json日期转换解决方案(总结)
- webmagic采集CSDN的Java_WebDevelop页面
- SpringMVC+MyBatis - 12 spring mvc4返回的js...
- DubboX参数验证失败的返回错误信息的格式转换...

查看评论

暂无评论