

Spring REST 风格 Web 服务的 Json 消息封装及解析研究

丁振凡

(华东交通大学 信息工程学院, 南昌 330013)

摘要: 针对 Spring RestTemplate 和 AJAX 对 REST 风格的 Web 服务的调用处理, 讨论了 Spring MVC 编程中用 Jackson 消息变换器和用 Google 的 Gson 工具对 Json 对象进行包装处理的详细配置和使用方法, 并给出了消息转换的各自应用特点。

关键词: Spring; REST 风格 Web 服务; Json 消息; RestTemplate; AJAX

中图分类号: TP393

文献标识码: A

文章编号: 2095-2163(2012)02-0009-03

Research on Json Message Encapsulation and Analysis of Spring RESTful Web Service

DING Zhenfan

(School of Information Engineering, East China Jiao Tong University, Nanchang 330013, China)

Abstract: With the calling process to RESTful Web Service by Spring RestTemplate and AJAX, the paper discusses how to implement the Json message Encapsulation with Jackson message converter and Google Gson tool in Spring MVC, and it also gives out the related characteristics for using each convert style.

Key words: Spring; RESTful Web Service; Json Message; RestTemplate; AJAX

0 引言

在面向服务的计算中, 基于 REST 风格的 Web 服务, 因其定义、调用的高效和简洁性, 成为当前服务计算的新热点。Spring MVC 支持 REST 风格的 Web 服务的定义, Spring 同时也定义了 RestTemplate 以配合 Web 服务的调用。另一方面, 在 B/S 应用中, 浏览器可通过 AJAX 技术调用 Web 服务, 并根据服务的返回结果动态更新页面内容^[1,2]。REST 服务的响应消息的数据对象包装形式使用较多的是 XML 和 Json。Json 作为一种易被 JavaScript 解析的对象消息格式, 其在消息传递、解析和处理效率方面均具有一定的优势, 因此受到越来越多应用开发者的欢迎。

在 Spring REST 风格的 Web 服务应用开发中, 对返回给调用者的数据对象进行 Json 消息包装存在两种形式。一种是转化为 Json 字符串, 其特点是在客户端被 Javascript 的 eval 函数解析转换为 Javascript 对象格式。另一种是 Jackson 数据对象格式, 要用相应的消息变换器进行识别。本文就使用 RestTemplate 和 AJAX 调用 REST 服务所涉及的 Json 消息处理的相关问题进行讨论。

1 基于消息转换器的 Json 对象包装

在 MVC 控制器的 Mapping 处理方法中可以通过 @ResponseBody 注解定义方法的返回对象为服务的响应消息^[3]。该方法在返回数据时, 将借助 HttpMessageConverter 实现消息的转换处理。为了将 Java 对象变换为 Json 消息格式, 需要在工程中引入 jackson 的 jar 包。其中包含 MappingJacksonHttpMessageConverter 这个实现 Json 消息包装的转换器。在 Bean 的配置文件中, 可设置该转换器的 Bean 的 supportedMediaTypes 属性值为 "application/json"。

1.1 REST 风格的 Web 服务端的编程处理

1.1.1 Bean 的依赖注入配置

在 MVC 配置文件中需要对相关 Bean 进行配置。配置的关键是给注解方法的调用处理适配器 (AnnotationMethodHandlerAdapter) 注入 Json 消息转换器, 这样 @ResponseBody 注解返回的数据将用 Json 进行包装。具体配置为:

```
<bean class="org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter">
    <property name="messageConverters">
        <list>
            <ref bean="jsonConverter" />
        </list>
    </property>
</bean>
<bean id="jsonConverter"
class="org.springframework.http.converter.json.MappingJackson
HttpMessageConverter">
    <property name="supportedMediaTypes" value="application/json" />
</bean>
```

1.1.2 REST 风格的 Web 服务的控制器设计

以下为一个简单 MVC 控制器实现代码, @ResponseBody 注解定义返回数据为响应消息, 该消息将根据配置定义的 Json 转换器进行数据包装。这里仅返回字符串对象, 实际允许任何可串行化的 Java 对象。

```
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
```

收稿日期: 2012-03-20

作者简介: 丁振凡 (1965-) 男, 江西丰城人, 学士, 教授, 硕士生导师, 主要研究方向: 语义 Web、分布式计算、计算机辅助教学。

```
ing;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.ResponseBody;
@Controller
public class RESTController {
    @RequestMapping( value = "/show/{person}", method = RequestMethod.GET)
    @ResponseBody
    public String show( @PathVariable( "person" ) String me ) {
        return "hello ." + me; // 返回消息
    }
}
```

1.2 REST 服务的客户调用端的处理

对于用 `@ResponseBody` 返回的响应消息, 服务调用者端要用相同的消息变换器来识别消息并转化为 Java 对象。Spring 框架中提供了 `RestTemplate` 来访问此类 Web 服务, 并提取服务的返回结果, 从而为应用系统间服务功能的互相调用提供了支持。

1.2.1 REST 模板的注入配置

在配置 `RestTemplate` 时, 如果需要解析特定格式的消息, 可通过 `messageConverters` 属性列出使用的消息转换器。以下配置中引用了“`jsonConverter`”Bean 定义的转换器, 将返回的 Json 消息进行转换, 解析成 `JsonObject`。

```
<bean id="restTemplate" class="org.springframework.web.client.RestTemplate">
    <property name="messageConverters">
        <list>
            <ref bean="jsonConverter" />
        </list>
    </property>
</bean>
<bean id="jsonConverter"
class="org.springframework.http.converter.json.MappingJackson2HttpMessageConverter">
    <property name="supportedMediaTypes" value="application/json" />
</bean>
```

1.2.2 通过 REST 模板调用 REST 服务

在应用环境中, 可以用 `getBean` 方法获取 `RestTemplate` 对象, 并借助对象的方法调用 REST 服务。`RestTemplate` 的 `getForObject` 方法完成 get 请求, `postForObject` 方法完成 post 请求, `put` 方法对应地完成 put 请求, `delete` 方法完成 delete 请求等。以下为 `getForObject` 的使用代码:

```
RestTemplate template=appContext.getBean( "restTemplate");
String x= (String) template.getForObject( url + "/show/{person}", String.class, new Object[] {"john" });
```

其中 `getForObject` 方法的第 1 个参数是 http 请求的地址,

第 2 个参数是响应返回的数据的类型, 第 3 个参数是 URL 请求中需要设置的参数。

2 基于字符串格式的 Json 包装

基于字符串格式的 Json 对象包装适用于浏览器通过 AJAX 访问 Web 服务, 因为 Javascript 脚本通过 `eval` 函数可将 Json 消息很容易地转化为 JavaScript 对象格式。

2.1 Web 服务端的处理

将 Java 对象包装为 Json 字符串的工具几种, 本文选用 Google 提供的 JSON 转换工具包, 其 jar 包为 `gson-1.7.1.jar`, 利用 `Gson` 对象的 `toJson` 方法可将对象数据转换为 Json 字符串格式。发给调用者的为普通文本串, 以下为代码样例:

```
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import com.google.gson.Gson;
import javax.servlet.http.HttpServletResponse;
import java.util.*;
@Controller
public class monitorController {
    @RequestMapping( value="/lookfor",method=RequestMethod.GET)
    public void display( HttpServletResponse response ){
        response.setHeader( "ContentType", "text/plain");
        Gson gson = new Gson();
        try {
            List<String> message=Search.getMessage();
            //调业务逻辑获取数据
            String listToJson = gson.toJson( message);
            //将对象转换为 Json 格式
            response.getWriter().write( listToJson);
        } catch (Exception e) { e.printStackTrace(); }
    }
}
```

另一个使用较多的是 `org.json` 公司的工具, 对应的 JAR 包为 `json-lib-2.4-jdk15.jar`, 其中提供了 `JSONArray` 和 `JSONObject` 两个类, 可用其提供的 `fromObject()` 方法将数据对象转换为 JSON 格式。

2.2 浏览器客户方的编程

浏览器客户方采用 AJAX 技术实现与服务端的通信。用 `XMLHttpRequest` 对象发送 URL 请求访问服务端的 Web 服务, 通过 `XMLHttpRequest` 对象的 `responseText` 属性提取来自服务端的响应消息。调用 JavaScript 的 `eval` 函数将服务器返回的 JSON 文本转化为对象形式, 利用 DHTML 技术改变页面显示。JavaScript 脚本代码如下:

```
var xmlhttp= new XMLHttpRequest( "Microsoft.XMLHTTP");
myurl="lookfor"; // 服务器的 URI
xmlhttp.Open( "GET",myurl, false);
xmlhttp.send(); //发送请求
```

(下转第 17 页)

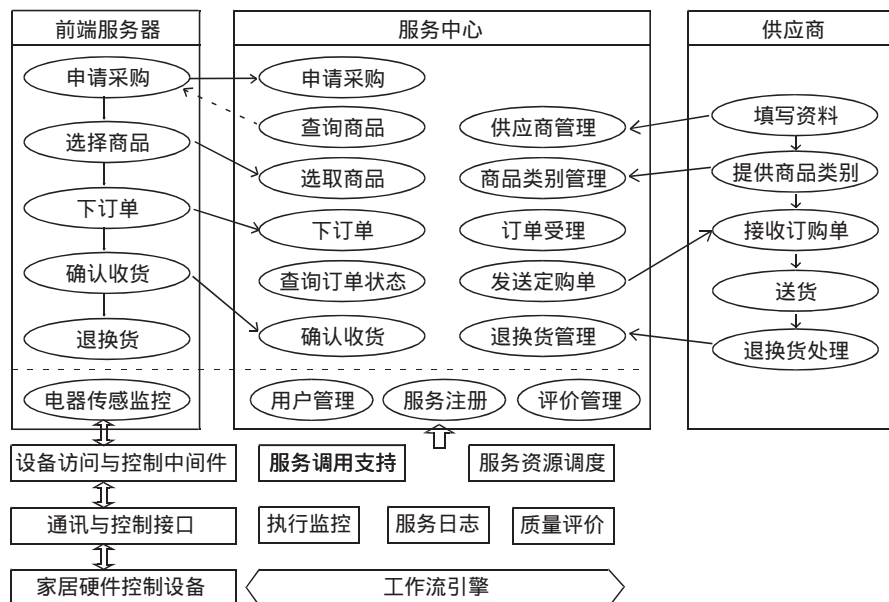


图7 采购服务业务功能设计图



图8 采购服务业务界面设计图

基于 BPMN 建模语言的家庭服务过程建模方法,并以具有典型性的采购业务为例,给出了建模实例、系统设计和实现。

本文方法是针对家庭服务过程建模的通用方法,适用于智慧家庭的各服务主题,但由于服务主题间存在差异,后期工作中需要针对特殊服务过程需求进行完善和优化,同时加入对价值、性能、质量等方面因素的考虑。

参考文献:

- [1] "Introduction |Sofia". <http://www.sofia-project.eu/packages>.
- [2] "SENSEI-Project Description". <http://www.ict-sensei.org/index>.

(上接第10页)

```
var c = xmlhttp.responseText; //获取响应 Json 串
var jsonObj=eval("(" + c + ")");
for ( i=0;i<jsonObj.length;i++){
    document.getElementById("disp").innerHTML="" +jsonObj[i]; //DHTML 显示消息 }
```

3 结束语

文章分析了 Spring MVC 的 REST 服务对响应消息的两种 Json 封装形式。一种是基于 Jackson 消息转换器的处理形式,另一种是 Json 字符串形式。这两种方式各有特点,实际应用中可根据需要具体选择。基于 Json 消息转换的服务可借助 Spring REST 模板来访问,适合于分布式计算中

php?option=com_content&task=view&id=12&Itemid=65.

- [3] "The Hydra project". http://www.hydramiddleware.eu/articles.php?cat_id=9&rowstart=0.
- [4] "Despre Project" - Website Project FCINT. http://www.fcint.ro/Despre_Project.aspx.
- [5] XU J, Lee Y, TSAI W et al, "Ontology-based smart home solution and service composition". International Conferences on Embedded Software and Systems 2009.
- [6] MESHKOVA E, RIIHIHJJARVI J, MAHONEN P. "Modeling the home environment using Ontology with applications in software configuration management". International Conferences on Telecommunications (ICT'08).
- [7] TSAI W, LEE Y, WIEZEL A et al. "Ontology-based service composition framework for syndicating building intelligence". Conference on Commerce and Enterprise Computing 2009 IEEE.
- [8] 中国智能家居网_解读智能家居—什么是智能家居 2007-0-3-22 <http://www.smarthomecn.com/html/2007-03/3647.html>.
- [9] Introduction to BPMN, Stephen A. White - IBM, May 2004, http://www.omg.org/bpmn/Documents/Introduction_to_BPMN.pdf.

对来自其他计算机的服务的调用,从而实现应用集成整合或应用间的信息传递。基于 Json 字符串形式的消息可用于浏览器与 Web 服务器间的数据交换。客户端通过 AJAX 技术访问服务并将根据结果动态改变页面,可用于当前流行的胖客户端应用中。

参考文献:

- [1] 屈展,李婵. JSON 在 Ajax 数据交换中的应用研究[J]. 西安石油大学学报(自然科学版) 2011(1):95-98.
- [2] 谭力,杨宗源,谢瑾奎. Ajax 技术的数据响应优化[J]. 计算机工程 2010(4):52-54.
- [3] 张宇,等. 基于 Spring 的 MVC 框架设计与实现[J]. 计算机工程 2010(2):59-62.