

json对象在前后端数据交互的作用

郑 雄 冉启亚 杨真珍 周维敏 彭芳策
兴义民族师范学院 兴义市 562400

摘 要: 在信息平台“大数据”的时代,传统关系数据库无法满足大量非结构化数据的有效管理。由此诞生了附加多个不同数据源的方式,这个方式极大地改善了因特网上不同数据的管理性能。但是,在异种数据源的数据相互作用中,需要对不同的数据源进行描述,为了提高数据源在系统中的数据相互作用的开发效率,本文研究了json对象在前后端数据交互的作用,从而达到减少开发代码的目的,实现软件开发的高效。

关键词: 大数据; 异构数据库; JSON; API

The interaction of JSON objects in front and back end data

Xiong Zheng, Qiya Ran, Zhenzhen Yang, Weimin Zhou, Fangce Peng
Xingyi Normal University for nationalities, Xingyi 562400

Abstract: in the era of “big data” of information platform, the traditional relational database can not meet the effective management of a large number of unstructured data. This gives birth to the way of attaching multiple different data sources, which greatly improves the management performance of different data on the Internet. However, in the data interaction of heterogeneous data sources, different data sources need to be described. In order to improve the development efficiency of the data interaction of data sources in the system, this paper studies the interaction of JSON objects in the front and rear data, so as to reduce the development code and achieve the efficiency of software development.

Keywords: big data, heterogeneous database, JSON, API

1 基于JSON的异构数据源数据交互方案的提出

1.1 异构数据源数据交互模型的内部框架设计

1.1.1 点对点模型

当联合数据库系统使用点对点模型构建不均匀的多数据源数据库系统时, 通过从数据库的源代码移植到目标数据库的数据, 在不同的数据库之间进行解析, 现数据交互的接口。

1.1.2 星型模型

通过临时逻辑数据介质完成中间数据与不同数据源之间的数据交互, 从而实现不同数据源之间的数据交互。

1.2 基于JSON的异构数据源交互模型内部转换方案设计

不同的数据源相互作用必须考虑源数据的妥当性测试、特殊数据的处理、源数据与JSON数据之间的映射关系的设计与实施。在将不同源数据转换为目标数据源的过程中, java对象模型广泛用于连接两个数据。(见图1)

1.3 数据交互中异构数据源的选择

针对不同数据源的数据相互作用模型的框架, 选择

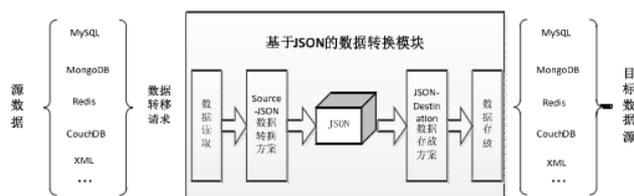


图1 基于JSON的数据转换模型

中间数据形式。该模型的创意是实现不同数据源与json之间的转换, 添加新数据源就可以顺利地添加新数据源, 唯一的变换接口功能将被添加到API, 不改变以前的代码。

1.4 基于JSON的数据交互模型的API接口标准设计

基于json的不同数据源的数据相互作用模型进行封装, 只有最外侧的接口对外开放。程序开发者根据设计的数据相互作用方案, 输入适当的参数, 只需调用对应的接口函数, 就能够实现不同数据源之间的数据交互。

(1) 接口名称: Verify, 该接口用于验证待处理的数据的有效性。

●输入参数:

1) 字符串数据、源数据、JSON数据等。

2) 用于验证数据方案的有效性时匹配的模式文件的地址。

输出结果：布尔，判断数据是否合法。

在图2为验证器接口实现的流程图。

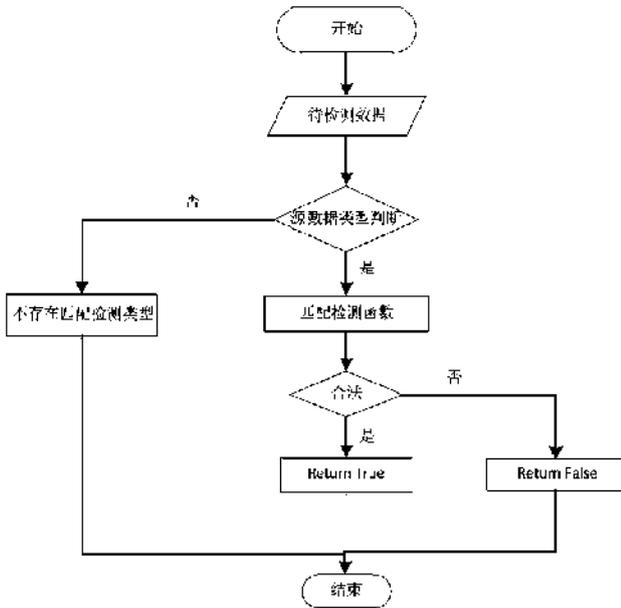


图2 Verify接口的实现流程图

首先，获取转换对象的源数据，参照对应的模式文件，验证数据的妥当性。如果返回了true，则源数据是正当的。否则源数据不正确，返回false。

(2) 将源数据转换为JSON数据，作为外部从包含时间和日期的JSON的不同数据源的转换功能，对外部暴露具有数据处理功能的接口。

●输入参数：

1) 对象SourceName、数据源类型（MySQL、Redis、MongoDB、XML）用于确定要获取的数据地址。

2) 字符串、列表、地图等用于JSON转换的数据结构体集合

3) string jsonfilename，保存地址JSON数据的JSON文件。

●输出结果：boolean，判断转换是否成功。

图3为SourceToJSON接口的实现流程图，

首先，取得变换对象的源数据，调用参数对应的数据存储结构，使调用源的函数匹配。转换后的数据存储参数内指定的物理位置中，返回为true，源数据将被正式转换为JSON数据格式。否则，返回false，显示数据转换失败。

实现向目标数据库插入JSON数据。作为曝光接口，它将把包含时间和日期数据处理功能的数据源转换功能作为目标的JSON封装。

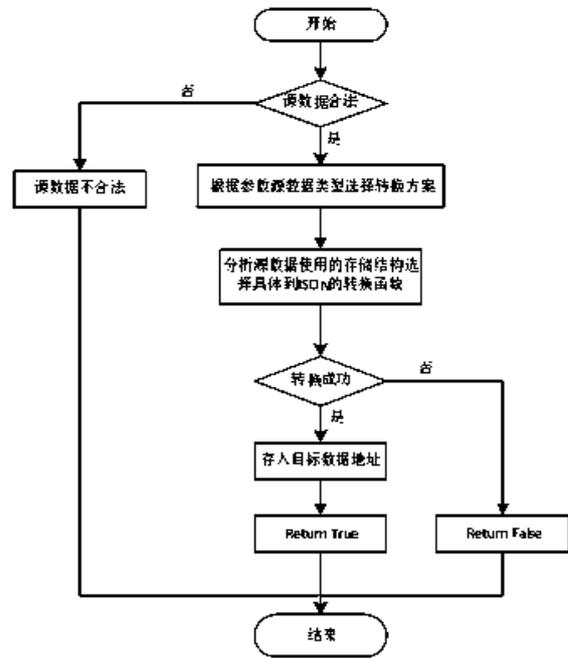


图3 SourceToJSON接口实现流程图

●输入参数：

string jsonfilename，用于存储JSON数据的文件地址。解决对象类名、JSON数据的所有字段、关键类名、选项参数。

字符串SourceName、MySQL、Redis、MongoDB等分析后的JSON的对话型数据原型。

输出结果：对象、数据存储位置。

在图4为实现JontoOdestination接口的流程图，

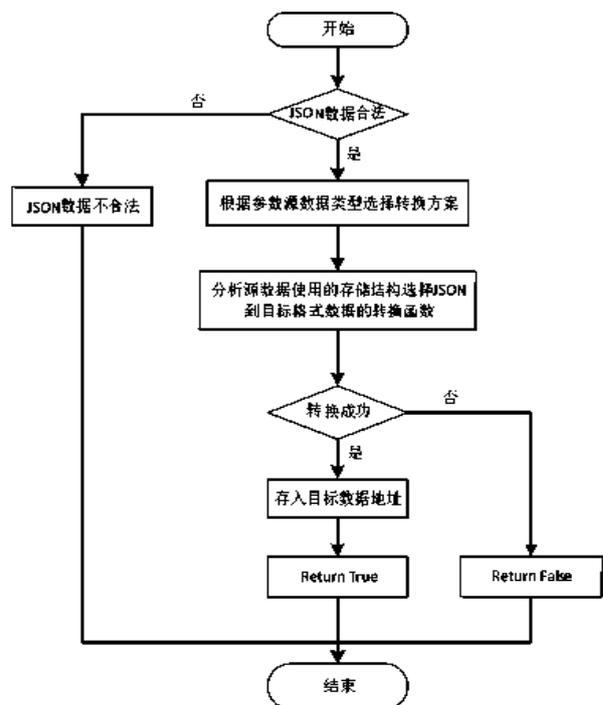


图4 JontoDest ination接口实现的流程图

首先，取得应转换的JSON数据，使参数的对应的数据存储结构一致，使调用源的函数一致。转换后的数据存储在参数指定的Java存储结构中，结果集合正常返回，表示JSON数据格式被正常转换。除此以外，返回返回false，表示JSON数据转换失败。

2 通用数据交互接口的设计与实现

对基于json的不同数据源的数据相互作用方式的整体实施的技术渠道，导入了各模块的层次结构和功能设计。

2.1 异构数据源模型的总体设计目标和结构

2.1.1 总体设计要求

面对Web应用的复杂数据环境，根据不同功能数据库之间的数据传输模型，开发用于异功能数据库之间的数据传输的API的大规模结构化的数据、数据持续性和其他要求。

1) 数据源选择mysql、mongodb、redis，以实现不同数据源数据和json格式数据之间的转换方式。不同类型的数据库转换方案之间的兼容性要求尽可能低以满足模型设计的高可缩放性。

2) 使用JAR包进行接口封装和Java语言编码，系统开发者只需要调用API的相应函数来完成异种数据的转换。

2.1.2 模型的详细结构设计

1) 基本客户端请求服务端数据的数据模型：

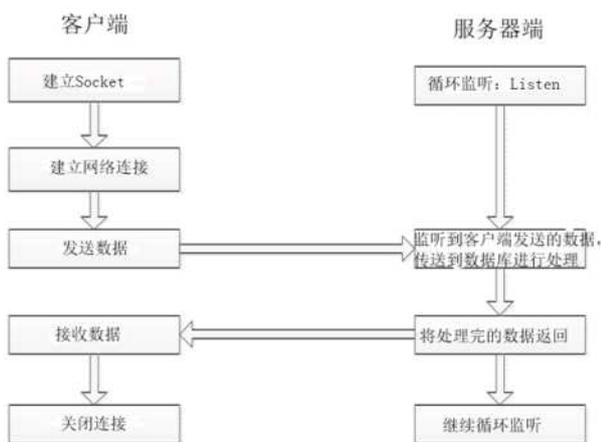


图5 C/S数据模型图

客户端请求数据的步骤是：服务器循环监听客户端连接请求，监听到查询请求后建立连接，客户端发送查找关键字作为参数到服务器端，服务器端执行数据库查询语句查找数据，将数据进行解析，得到指定格式的数据并向客户端展示。

2) 加入数据库中间件技术后的模型：

C-M-C架构指客户端、中间件、服务器三层结构。

这是由客户端/服务器结构开发的。图6示出了其基本框架。

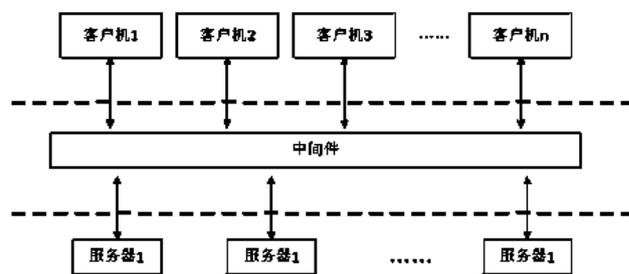


图6 C-M-C架构图

客户端：请求应用程序，显示数据的一部分。中间件：在相同或不同环境中格式化数据的数据处理中间件，可以双重与客户端和服务端对话。

服务器：数据的物理存储侧。基于多个不同数据库的系统可包括数据库集群和分布技术，以管理数据和持续。

3) 本文实现的具体交互模型：

当客户端应用触发用于数据处理的请求时，核心创意可以通过数据交互接口定义的功能来完成触发数据源的数据查询转换。可以将数据源最终存储到目标数据源。新操作还可以旁路此文档的数据交互模块，并执行目标数据库数据更新文本以直接完成数据库更新。您必须选择另一个方案的商业方案。以下概括详细的模型设计和分析：

在异种数据源系统中，系统的不均匀性决定了客户端的性能。数据请求可能在数据库系统的不同数据库中分发。为了提高数据利用效率和透明度，需要完成api接口中的数据格式转换，以提高数据传输效率。

在模型设计中应该解决的问题是图表和json之间的4条线的实现。界面函数的定义是关键字。

通过取得上述两点平衡，合理的接口功能系统的设计不仅可以确保API的封装，还能保证系统开发的可操作性。

2.2 MySQL与JSON数据交互方案实现

关键函数实现

实现MySQL数据库中的数据读取，转换成JSON格式。参数被设置为在MySQL数据库中读取的数据存储位置 and JSON文件输出的名称。

使用Object Mapper的Write Value方法转换Java对象。此方法可以将Java对象转换为无JSO的Write Value方法。输出流、输出转换后的JSON内容、或者指定存储转换后的JSON内容的输出文件的两个参数。第二个参数是需要转换的Java对象。

对于日期类型的数据，定义串行化和逆串行化功能，并调用日期Serializer()和日期Siderializer()。在实施中，必须指定杰克逊的JSONGenerator对象的实例。

当两个表链接在一起，包括彼此的部分时，死循环问题经常发生。在这个例子中，存在逆串行化的双向转换。SFJSONJSONException: 楼层有周期!

选定的处理方式是使用JSONCONFIG的setCycleDetectionStrategy()来格式化Dead Loop的值、Jackson的parse()格式转换设置，日期串行化的所有对象由日期串行化或逆串行化处理器处理。

JMySQL接口的内部函数继承了Add、Delete、Change、Check等MySQL连接器接口的基本功能，并能执行一些基本操作。

2.3 Redis 与 JSON 相互转换方案实现

Redis 与 JSON 转换函数的实现

(1) 关键函数实现:

JSONTretoreString(): JSON数据以独立的键/值形式存储在Redis String数据结构体中。函数的参数是覆盖JSON DATA键的类对象类名称，函数存储的是JSON数据的JSON文件名。

函数描述: 将循环加载到JSON文件中的JSON数据，然后根据参数中的类对象属性串行化，并将该循环存储在图中以获得结果，并调用Jedis对象的方法完成循环中的数据存储。

redistojson(): 从REDIS数据库读取数据，转换为JSON数据，保存到指定文件。其主要参数是存储Jedis的操作数据和JSON数据的JSON文件。

函数说明: 参数读取READIS数据，转换JSON的写入*.JSON数据。

(2) Redis 与 JSON 转换的类图 7 所示:

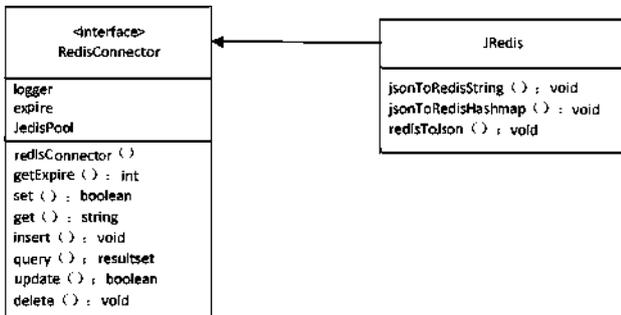


图7 JRedis类图

Jredis接口的内部函数继承了RedisConnector接口的基本功能的一部分，如GET、SET、INSERT等。

(3) 通过redis管道技术优化jsonToRedis接口函数

REDIS是基于C/S模型和要求/回答协议的TCP服务。

每次客户向服务器发送命令请求时，它都会被屏蔽，降低了使用REDIS快速访问数据的意义。

管道技术能很好地改善这个状况。这将发送，而不依赖于redis来封装“批判”工具并执行多个命令。此性能适用于通过REINIS管道执行插入REDIS数据库命令。

3 基于JSON异构数据源数据交换方案的验证与实例应用

为了确保API密封的正当性，需要确认不同数据源与JSON数据之间的转换功能，以确定功能的准确性和执行效率。

3.1 接口函数功能和转换效率测试

测试过程需要准备不同的存储结构数据，例如用于存储JSON数据的列表、set、hashmap和json文件的源数据。转换效率测试主要用于测试不同数据的转换时间。

指定函数的功能测试应确保数据库的正确连接、I/O流的正确封装、数据库的正确操作文本及例外处理。测试需要设计更多的测试案例，以确保测试的综合性。

不同数据源可以采用不同的测试用例，比如MySQL的Bean, List, Set, Map, Redis的HashMap, String, MongDB的Document等，每种数据结构随机设计50组测试数据对对应接口函数进行测试，测试流程8所示:

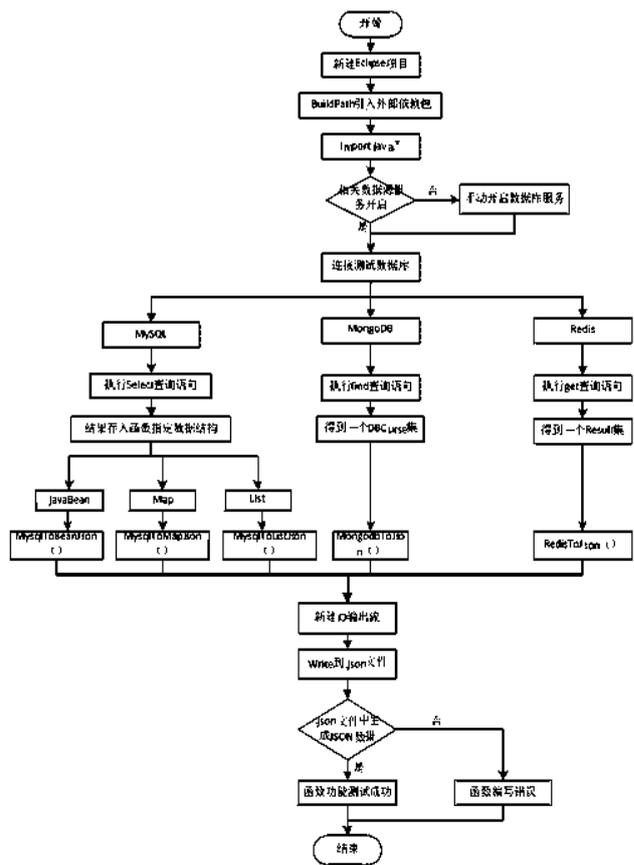


图8 SourceToJson 函数测试流程

测试用例组	测试用例结构
测试用例 1 组	{"key":"value"}
测试用例 2 组	[{"key":"value"}]
测试用例 3 组	[{"key":"value"}, {"key":"value"}]
测试用例 4 组	{"key":[{"key":"value"}]}
测试用例 5 组	[{"key":[{"key":"value"}, {"key":"value"}]}, {...}]

图9 不同测试用例结构

通过对功能测试结果的解析, 所完成的接口基本能够完成不同数据源与json数据之间的转换。由于功能测试保证接口的正当性, 需要测试设计函数接口的数据转换效率。这决定了接口的实用性。如果效率低, 表示转换方式没有实用的应用价值, 并且设计不合理。

3.2 实验应用平台环境搭建

图10以图示促进读取器确认设计计划的硬件和软件配置。

相关环境	具体名称
操作系统	32 位 Windows 7.1
集成开发环境	IntelliJ、navicat、RedisDesktopManager、MongoVUE、Eclipse
平台框架	Jersey+Mybatis+ Spring
项目管理工具	Maven
编程语言	Java、JavaScript
外部开发包	Redis的 jedis-2.1.0、jackson-1.9、google 的 Gson

图10 风险管理平台环境搭建表

4 结束语

随着数据时代和计算机技术的高速发展, 应用程序面临的数据也更加巨大, 数据结构也越来越复杂, 单一结构数据库系统难以满足多应用方案的数据积累和处理。在数据库框架中导入了多数据库系统。在互联网上非结构化数据的增加比例中, 非结构化数据的比例正在增加。更多的非结构化数据同时访问, 更好地发展NOSQL数据库技术。加入了异种多数据库系统的框架。不同数据库之间的数据相互作用需要在多应用多平台系统开发的编码过程中频繁解决。被封装了的异种数据库数据相互作

用API, 成为值得研究的话题。在互联网平台系统中客户端请求与不同数据源之间的不同数据库与数据请求之间的数据交互的当前商务背景下, 完成了基于json中间数据格式的数据库不同数据相互作用模型与api的设计。

参考文献:

[1]魏志伟.一种基于json数据协议的Web管理系统数据交互方法.

[2]叶小谋, 王蒙, 唐新余, et al.基于JSON的机构养老数据交互方法及系统.

[3]陈郁欣.Arduino客户端与服务器交互数据以JSON格式传输的实现[J].电子世界, 2017(24): 132-133.

[4]邢四为.基于JSON的信息交互系统的研究与实现[D].安徽大学, 2013.

[5]龚成莹, 邢敬宏, 胡银保.基于JSON的Android移动终端与PHP及MySQL数据通信[J].工业仪表与自动化装置, 2013(1): 63-65.

[6]李志军.基于JSON的Android与SQLServer数据通信[J].电脑与信息技术, 2015, 23(4): 51-53.

[7]朱峰.基于JSON的互联网异构数据整合的应用研究[D].2016.

[8]王魁生, 王晓波.利用JSON进行网站客户端与服务端数据交互[J].软件导刊, 2010, 09(3).

[9]谭翔纬.利用JSON实现Android客户端与Web服务器间的数据交互[J].福建电脑, 2013, 29(2): 166-167.

[10]王华志.基于JSON的异构数据源数据交互技术研究[D].

[11]周兵.基于Android数据交互系统的研究与实现[J].湖南工程学院学报: 自然科学版, 2015(3): 46-50.

[12]刘静.脑卒中云平台数据交互与存储研究[D].2016.

[13]程桂花.MVVM前后端数据交互中安全机制的研究与实现[D].