

文章编号: 2095-2163(2020)01-0156-04

中图分类号: TP311.52

文献标志码: A

# 基于 Web Service 的油气田信息服务构建

范伦杰<sup>1</sup>, 张余强<sup>2</sup>, 罗彬彬<sup>2</sup>

(1 四川大学 电子信息学院, 成都 610064; 2 成都西图科技有限公司, 成都 610065)

**摘要:** 传统的油气田信息服务是以数据库服务的标准进行数据的传输与展示。由于数据库信息在网络上传输和加密的需要, 本文探讨并实现了一种基于 Http 协议和 SOAP 协议的 Web Service 数据传输方案。该方案针对油气田信息传输的保密和权限验证需求, 具体设计并实现了一套完整的 Web Service 数据传输方案。该方案实现了大批量数据库信息的统一传输和控制, 并将身份认证功能从数据库中解放出来, 在 Web Service 中完成了权限校验功能, 为油田信息化的发展提供了有益的参考和借鉴作用。

**关键词:** 在线浏览; Web Service; WSDL; SOAP

## Construction of oil and gas field information service based on Web Service

FAN Lunjie<sup>1</sup>, ZHANG Yuqiang<sup>1</sup>, LUO Binbin<sup>2</sup>

(1 College of Electronic Information and Engineering, Sichuan University, Chengdu 610064, China;

2 Chengdu Xitu Technology Co. Ltd., Chengdu 610065, China)

**[Abstract]** The traditional oil and gas field information service is based on the standard of database service for data transmission and display. Because of the need for database information to be transmitted and encrypted on the network, this paper discusses and implements a Web Service data transfer scheme based on Http protocol and SOAP protocol. According to the requirements of confidentiality and authority verification of oil and gas field information transmission, a complete Web Service data transmission scheme is designed and implemented. The scheme realizes the unified transmission and control of large-volume database information, and liberates the authentication function from the database, completes the permission verification function in Web Service, which provides helpful reference for the development of oilfield information industry.

**[Key words]** online browse; Web Service; WSDL; SOAP

### 0 引言

随着油田信息产业的发展, 油气田管理人员需要完成对中石油数据库信息的统一化管理工作。已有的油气田数据是以各种方式分散在各台服务器上存储起来的, 需要一个统一的服务去支持第三方开发组件调用, 防止第三方机构构建的展示平台污染数据源, 配有对第三方的身份信息验证功能, 保证信息的数据安全。基于以上需求, 调研了网络上常见的数据资源的共享方式, 其中包括公网上广泛流行的 Restful 微服务架构, 较为传统严谨的 Web Service 数据传输服务。Restful 架构仅是基于 Http 协议来设计运行, 相较而言 Web Service 具有更严格的语义描述, 对数据的交互有着严格的 WSDL 文件描述。相对于 REST 架构来说, 虽然缺少了系统的延展性和伸缩性, 但是在安全和规范方面更为标准。因此采用 Web Service 协议来实现数据的共享<sup>[1]</sup>。

### 1 需求分析及服务总体架构设计

本次研究中, 根据现有数据库中的数据表结构,

将数据分为 5 个子模块来传输, 分别为: 生产动态、生产成果、储量成果、井筒曲线和统计卡片, 不同勘探生产模块, 存储的主要信息不同。各个模块的主要数据内容见表 1。

表 1 数据模块及其主要内容

Tab. 1 Data module and its main content

功能模块	功能模块包括的内容
生产动态	生产动态的数据是每天的勘探生产成果的一个简要汇报
生产成果	生产成果内包括设计、钻井、路井、测井、试油、物探和化验共 7 个模块, 这些模块存放在数据库数十张表中, 主要是近年来勘探生产过程中产生的大部分信息
储量成果	储量成果中的数据是历史上数年的数据整理和积累
井筒曲线	综合前面所有的数据, 在网页上绘制图形
统计卡片	以表格形式汇总报告数据

在梳理阐述前文功能模块的前提下, 还需要设

**作者简介:** 范伦杰(1996-), 男, 硕士研究生, 主要研究方向: 通信与信息系统; 张余强(1985-), 男, 硕士, 主要研究方向: 软件工程; 罗彬彬(1990-), 男, 学士, 主要研究方向: 软件工程。

收稿日期: 2019-11-28

计权限控制、文件服务功能。所有油气田信息的传输均以 Web Service 协议作为基础,而在此通用的数据交换协议下,拟对协议细节进行细化定制,同时构建数据方和需求方的交互语言格式,另外还要规范各个子模块的交互进程。系统功能模块的整体设计结构即如图 1 所示,不同的项目子结构,需要编写大量的业务代码以及封装公用逻辑业务,成功运行后就会展现出一个完整的项目服务。

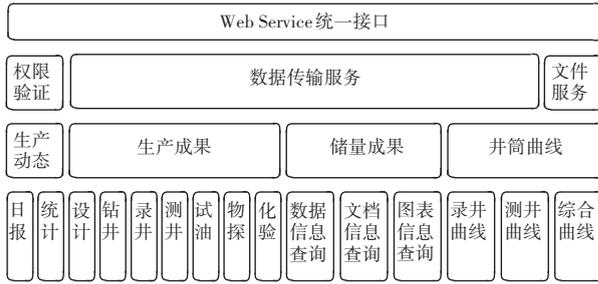


图 1 Web Service 总体功能模块

Fig. 1 Overall function module of Web Service

## 2 功能模块设计和实现

各个功能模块都依赖于 Web Service 实现的远端调用。Web Service 服务主要由 WSDL 文件进行描述, WSDL 文件将服务描述为一系列 request、response、绑定参数,绑定的数据传输协议的集合<sup>[2]</sup>。WSDL 以 XML 格式进行描述。通过 XML 格式的自描述特性,将 Web Service 各种端口的各方面描述信息和数据传输协议进行了规范化加工。具体的 WSDL 设计端口如图 2 所示。由于 WSDL 文件的语义描述准确,客户端在通过网络传输获得 WSDL 文件后,即可依此开发 Web Service 客户端。选择 Spring 集成 Apache CXF 框架是目前最为普遍和成熟的 Web Service 客户端开发方式。

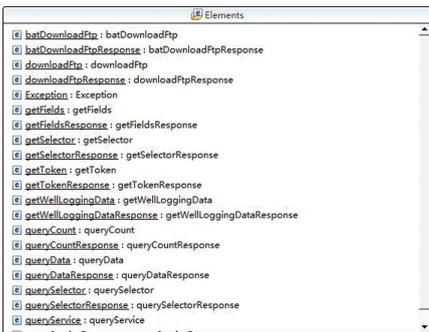


图 2 WSDL 端口设计图

Fig. 2 Port design drawing of WSDL

CXF 是 Apache 软件基金会开源的顶级孵化项目 Celtie 和 Xfire 的结合,而且高度兼容现有的主流后端开发 Spring 框架。通过配置 Spring.xml 文件,注册 CXF 的代理类到 Spring IOC 容器中,即可做到

JAX-WS 协议的部分客户端代码自动生成,代理类的设计主要对着 request 请求中 SOAP 协议自动化拆包和封包操作,这也极大地简化了开发人员的工作量<sup>[3]</sup>。

### 2.1 权限验证模块分析设计

油田权限认证信息存放在本地的数据服务器上。利用 Web Service 协议进行权限校验前,在数据调用方访问数据库时,需要加上一个对于数据调用平台的身份验证功能,该功能不能是利用 IP 地址方式对来访者进行限制。参考常见的网络鉴权思路,在服务端依据用户信息生成 Token,再依据 Token 进行身份验证<sup>[4]</sup>。

在 Web Service 平台下,数据绑定的消息格式一般都是 SOAP。Token 可以编码存放在 SOAP 请求中,但是由于在 CXF 框架中,SOAP 拆包封包等工作是由自动生成的代码代理实现,需要在 SOAP 包中添加自定义 Token 就需要创建自定义 SoapAuthHandler 类实现 SOAPHandler<SOAPMessageContext>接口,重写其中的 handleMessage()方法并配置该过滤类在 Spring.xml 文件中。

在 handleMessage()方法中,需要依据 Token 值和命名空间 namespaceURI,生成 QName,将其对象序列化,以字符串的形式添加到 SOAPHeader 中,并且设置过期时间为 6 个小时。本次研发的 Token 生成方案的原理公式可写为如下形式:

$$Token = SHA512(SHA512(UserInfo) + salt). \quad (1)$$

由式(1)解析可知,该方案就是依据 SHA512 哈希算法对用户信息进行加密,加盐,再加密生成<sup>[5]</sup>。

通过上面的哈希算法,对 UserInfo 信息进行加密传输,salt 是以一定函数对应于 UserInfo 的字符串。如此一来,即使用户 Token 被截取,也无法逆向 SHA512 哈希函数获取 UserInfo。研究给出的总体验证流程如图 3 所示。

在后续访问数据信息接口时,如果出现 Token 过期的情况,重新验证即可。该方案可以避免每次请求数据都发送用户名和密码,避免监听程序发现数据包中的身份信息,而且对于服务端而言,对该 Token 进行验证和比较后就能够确定访客身份信息。

### 2.2 数据传输模块设计

数据传输需要设计各种有针对性的 WSDL 中的模块。对于调用方面言,需要不同形式数据。参考常见的数据库查询语句,一次数据库查询就要制定准确的数据库表、条目数、排序方式、分页条件、字

字符串匹配等参数。把数据库中的数据集成到第三方页面中去,可选择的方式就包括将这些参数存储到 XML 文件中进行传输。

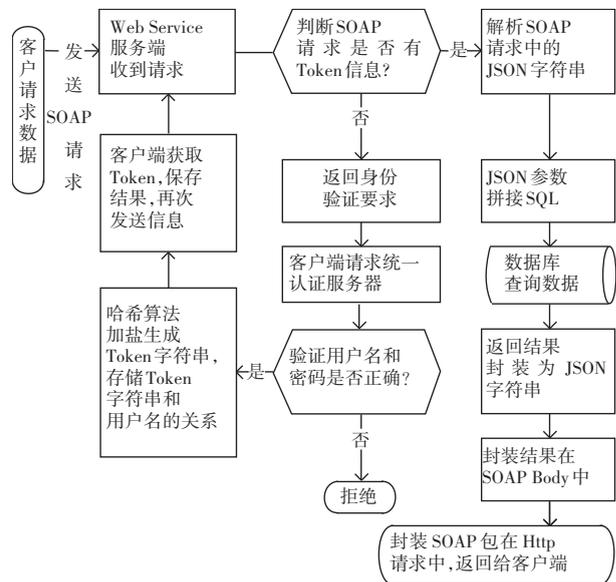


图 3 Web Service 身份验证流程图

Fig. 3 Authentication flow chart of Web Service

相较于 JSON 文件,XML 文件信息较为繁琐,在传输相同信息量的文本时会增大带宽占用,因此在数据库参数传输设计上本文选用了 JSON 格式作为替代。

研究可知,JSON(JavaScript Object Notation)是一种轻量的数据交换格式。这是 ECMAScript 的一个子集,采用完全独立于编程语言的文本格式来存储和表示数据。具有清晰明了的层次结构和逻辑关系,对其进行序列化和反序列化也有很多成熟的工具可供调取使用。

将 JSONObject 对象序列化后,将字符串封装到 SOAP 包的 Envelope 消息体内,而后再将该 SOAP

包封装到 Http 请求中完成数据的交换。以上思想,借鉴了网络传输过程中的 5 层传输模型,一层一层地封装信息,保证下层协议对上层协议的穿透性。

对于每个模块的不同数据库表,定义不同的 ServiceID 号、条目数以及排序方式等参数信息,对于查询 SQL 中较为复杂的级联查询,在 JSON 对象中通过在键值对的键值中再次包含一个 JSON 对象来实现。这种参数传递方式,不会受到传统的函数调用过程中函数参数表参数个数的限制,也能够对所表达的数据请求含义有着明确的表示。

本文研发设计的 JSON 文本信息示例详见如下。

```
{ serviceID : 1535,
  pageIndex : 1,
  pageSize : 50,
  search : [ { field : "RQ",
              type : 2,
              condition : 7,
              value : [ \ "2008-01-01" ]
            },
            { ... } ],
  sort : [ { field : "BH", order : 0 },
           { ... } ] }
```

该示例明确描述了需要信息的参数。在服务端 SOAP 拆包获取 JSON 字符串文本后,对其进行 JSON 反序列化和文本提取,提取出关键的查询信息后拼接 SQL。在数据库命令控制台中执行 SQL 获得 Result 结果集,对结果集进行再次封装,存入 SOAP Body 中返回给 Ajax 请求。前端在 HTML 页面中对 Ajax 请求的返回结果进行局部重新渲染,保证页面的展示效果。至此,数据传输后页面展示的部分结果图即如图 4 所示。

原油深度	原油深度		日产油	日产气	日产水	累计产油量	累计产水量	
928	936							01日11
2494	2514				11.44		202.96	01日11
3411.5	3444		13.84			56.65		01日11
1919.5	1935							01日11
2190	2213		0.36			3.97		02日01
4557	4593							01日11
4195	4203		12.05			67.13		01日11
3879	3886				16.02		39.86	01日11
1256	1267		6.22			37.54		01日21
					15.84			01日11
2581	2612.5							01日11
2672	2709							01日11
2976	3004				33.79		102.84	01日11
3228	3262		20.41	0.388	99.62	196.39	594.21	01日11

图 4 数据结果展示页面

Fig. 4 Data results display page

### 2.3 文件服务

油田系统已经存储的数据文件格式多样,包括大量的常规 Office 文档、油气田专用的 716 格式文件、地理信息 ArcGIS 文件和各种格式的图片文件。考虑到其与数据库内规范化的数据不同,本文就研创开发了一套 FTP 服务运行在文件存储的服务器上,使用 FTP 服务器进行文件下载。同时为了能让第三方开发通过 Web Service 方式获得文件,在 Web Service 接口中定制了一个文件下载的接口,该接口以 FTP 路径为参数,先在 Web Service 服务器以 FTP 方式下载文件到本地服务器,再以字节流的方式将数据发送到调用方。当调用方获取到文件后,对各种格式文件可以做相应处理,图片可直接嵌入 Img 标签内进行预览,Office 文档可在文件格式转码后在 Web 浏览器中用 PDF.JS 插件进行预览,ArcGIS 文件信息可用于开发三维地形服务等。

### 3 接口压测

在进行了前文各项子模块的设计后,接下来就要对每个模块进行压力测试。由于 Web Service 服务器的承载能力受到多种因素限制,包括硬件性能、

网络 IO 吞吐能力、操作系统、工业软件的架构设计等等在内,后续还需对该工程的具体负载能力进行科学客观的评估。

针对 Web Service 的接口服务的压力测试有多种常用工具,诸如 Apache JMeter 和 SOAPUI 等软件,也可以直接用 JAVA 网络 IO 请求写代理脚本去进行压力测试<sup>[6]</sup>。在全面调研过后,最终选取 JMeter 的压力测试工具集测试,相对其它几种方案更为简洁,具有直观的图形化界面和完备的参数输入。本实验的硬件条件为 CPU I5-4460 3.20 GHz, 16 GBDDR3 内存,WIN10 企业版操作系统,100 M 局域网。测试时,先配置 Http 请求头的参数,再将请求的 SOAP 消息体拷贝到 JMeter 的 Http 请求的 body 中,发送请求给 Web Service 接口。

本次研究中,设置并发的线程数为 500,Ramp-Up 时间为 1 s,用来模拟同时有 500 个人去访问同一接口,并设置请求 Loop 次数为 10,预估为 5 000 个请求样本。由于部分的请求被服务器端阻塞,对应的 Jmeter 请求线程也被暂停了,实际统计出的样本数量为 4 481 个。实验测试结果数值见表 2。

表 2 请求响应汇总表

Tab. 2 Request response summary form

Lable	样本	平均值/ms	中位数	最小值	最大值	异常/%	吞吐量/sec	90%百分位	99%百分位
Http 请求	4 481	25 796	19 738	598	457 082	3.59	8.5	40 797	330 572

分析表 2 数据可以看出,一共发送了 4 481 个 Http 请求给 Web Service 接口,有着 3.59% 的请求错误率,平均能在 2 s 左右获得正确的反馈数据。90% 百分位的值在高强度并发面前也出现在合理范围内,99% 百分位因为属于错误响应,33 s 的响应时间已经不可接受。虽然传输会出现错误,但是通过合理的设计请求重传机制,依旧能够保证数据的正确性,满足现有的调用需求。

### 4 结束语

本文针对油田数据信息的展示以及安全需求,采用 Web Service 协议平台设计架构,设计并实现了一套基于该平台协议的数据传输系统。本次研究的成果主要在于成功利用该自描述协议完成了油气田勘探生产环境下的生产数据传输。其中包括从原始数据库中提取信息进行封装,对接口模块的规划形成完备的 WSDL 文件,对接口参数的合理设计、统一的权限管理以及 SOAP 拆包封包等技术的客户

端/服务器端实现。该方案不仅能够满足石油公司的需求,在实际的生产环境中,也可以按照该方案修改某些内容扩大使用场景,将该方案移植到其它的需求环境中去。因此该方案具有一定的工程效益。

### 参考文献

- [1] 曹军, 李林, 王正阳, 等. 基于嵌入式系统的 IEC61400-25-4 通信服务 Webservice 实现[J]. 价值工程, 2019, 38(28): 282-287.
- [2] 徐晓琴. 基于 Web Service 技术的 SAP 接口实现[J]. 电脑知识与技术, 2018, 14(14): 30-31, 34.
- [3] 朱钰鹏. 基于 Webservice 的城市管理信息交换平台设计与实现[D]. 武汉: 华中科技大学, 2017.
- [4] 于俊祥. 基于 Web Service 的企业应用集成研究与应用[D]. 杭州: 浙江大学, 2006.
- [5] 项武铭, 鲍亮, 俞少华. 基于 JWT 的 RESTful API 角色权限验证方案设计[J]. 现代计算机(专业版), 2018(34): 82-85.
- [6] 武金梅. 对缩短步数的 HASH 函数算法 SHA-256, SHA-512 的分析[D]. 济南: 山东大学, 2008.
- [7] 周宇. 基于 JMeter 的 Web 服务质量分析[J]. 中国新通信, 2015, 17(8): 99-100.