

四川省政务信息资源共享平台
接口技术规范
(试行)

四川省经济信息中心

2018年3月

目 录

一、指南说明.....	2
（一）目的.....	2
（二）内容概述.....	2
（三）术语和定义.....	2
（四）适用对象.....	3
二、WEB 接口服务简介.....	4
（一）四川省信息资源共享平台简介.....	4
（二）WEB 接口共享方式简介.....	5
三、WEB 接口服务规范.....	5
（一）接口类型规范.....	5
（二）接口地址规范.....	6
（三）接口请求方式规范.....	7
（四）接口请求参数规范.....	7
（五）接口应答内容规范.....	9
（六）其他约定.....	10
四、WEB 接口服务示例.....	11
（一）REST 接口示例.....	11
（二）WebService 接口示例.....	13

一、指南说明

（一）目的

为加快推动四川省基于电子政务外网的政务信息系统整合共享，规范服务提供方开发接口服务内容，服务需求方调用接口服务，特制定本规范。

（二）内容概述

本指南主要包括 WEB 接口服务介绍、WEB 接口服务规范、WEB 接口服务示例三部分。

WEB 接口服务简介描述了四川省政府信息资源共享平台中 WEB 接口服务基本情况。WEB 接口服务规范描述了实现接口的具体要求。WEB 接口服务示例简单编写了 WebService 和 REST 的样例，开发人员可以仿照此部分内容进行开发。

（三）术语和定义

本文涉及到的专用名词及术语定义如下：

- 政务外网局域网：接入政务外网的逻辑隔离区域，与部门其他网络之间有明确边界。
- 省级共享平台：四川省政府信息资源共享平台（政务

外网）。

- 资源提供方：通过省级共享平台提供资源的部门、地方政务部门。
- 资源需求方：通过省级共享平台获取资源的部门、地方政务部门。
- XML：扩展标记语言（Extensible Markup Language）。
- SOAP：简单对象访问协议（Simple Object Access Protocol）。
- WSDL：网络服务描述语言（Web Service Description Language）。
- REST：表述性状态传递（Representational State Transfer）。
- JSON：JS 对象标记（JavaScript Object Notation），一种轻量级的数据交换格式。

（四）适用对象

本指南适用于各部门和地市州，进行接口服务开发的系统设计人员和程序开发人员。

二、WEB 接口服务简介

（一）四川省信息资源共享平台简介

四川省政府信息资源共享平台由省级共享平台、市州级共享平台等多级平台组成。平台依托电子政务外网建设，是开展非涉密政务信息资源共享交换的窗口和枢纽，满足部门之间、市州之间及跨市州数据共享的需求。

平台定位为目录中心、资源中心、交换中心。目录中心汇聚全省的政务信息资源目录，支撑基于资源目录的共享交换工作。资源中心实现归集到省级共享平台的政务信息资源的加工、处理与集中管理。交换中心提供集中的共享交换服务，支持库表、文件、服务接口三种数据共享交换方式。下属各级共享平台横向对接所辖区域政务部门、基础信息资源库、主题信息资源库及其他社会信息库，纵向多级连通，形成横向联动、纵向贯通的全省数据共享交换体系部门参与信息共享的应用系统，根据实际需要选择合适的方式接入省级共享平台，实现与其他外部系统的信息共享和业务协同。

（二）WEB 接口共享方式简介

WEB 接口共享方式是交换中心 3 种共享方式之一，提供 WEB 接口方式进行数据共享，提供服务方可以从自己的业务系统直接封装，也可以将数据推送到前置机后再进行封装，封装完毕后，通过目录中心和交换中心进行发布，发布成功后，服务使用方就可以根据发布信息进行使用。

三、WEB 接口服务规范

为确保服务接口的标准性和可用性，服务接口应满足以下协议规范与技术要求。

（一）接口类型规范

接口类型采用以下 2 种方式：

➤ REST

REST 服务消息封装协议应采用 HTTP 1.0/1.1 标准。

➤ Web Service

Web Service 服务消息封装协议应采用 SOAP 1.1/1.2 标准。

服务提供方可以选择其中一种方式进行开发，返回的结

果都采用 JSON 格式。

服务提供方在发布服务的时候需要注明接口采用哪种方式进行实现，服务调用方根据该类型进行调用。

（二）接口地址规范

接口地址表示的是具体网址 URL，每个 URL 代表一种可以使用的资源，为了便于管理，URL 在发布的时候采用如下格式进行：

http://ip:port/部门拼音简写/系统英文名称/模块英文名称/业务操作英文名称/版本号

部门拼音简写：部门拼音的简写，用每个中文拼音的第一个字母，如省发展改革委就是sfzggw。

系统英文名称：系统的英文缩写，由服务提供者自行定义。

模块英文名称：模块的英文缩写，由服务提供者自行定义。

业务操作英文名称：业务操作处理的英文缩写，由服务提供者自行定义。

版本号：该服务的版本号，约定用vn进行定义，如该服务为第一次发布使用，命名为v1，以后有修改，命名为v2，以此类推。

如省发展改革委的统一管理系统的用户管理模块提供了一个获取用户详情的接口服务，那么命名就为：

`/sfzggw/ums/user/getDetail/v1`

（三）接口请求方式规范

REST接口类型只使用HTTP请求方式中的GET和POST。

GET：如果服务使用方要获取服务提供方的信息，采用GET请求方式。

POST：如果服务需求方要提交信息给服务提供方，采用POST请求。

（四）接口请求参数规范

为了系统安全和系统控制，请求参数除了业务系统自身规定的参数外，还需包含以下表格内容，发布的时候需要把所有请求参数在统一目录中进行发布，服务使用方根据这些参数进行调用。

请求参数	说明	是否必须	类型
deptName	请求部门名称,用于获知该服务被那个部门调用。	是	字符串
deptCode	请求部门编码,用于获知该服务被那个部门调用。	否	字符串
programName	请求程序名称,用于获知该服务被那个业务程序调用。	是	字符串
username	用户名,用于安全认证,如果业务系统对安全有要求,可以让需求方提供认证信息,通过认证后返才回业务数据。	否	字符串
password	用户密码,用于安全认证,如果业务系统对安全有要求,可以让需求方提供认证信息,通过认证后才返回业务数据。	否	字符串
pageNo	当前页(分页参数),需要获取分页数据的时候使用,如果不是分页数据,可以不提供。如果是分页数据,默认为 1。	否	数字
pageSize	每页记录数(分页参数),需要获取分页数据的时候使用,如果不是分页数据,可以不提供。如果是分页数据,默认为 10。	否	数字

服务提供方收到上述参数后，可以用于用户认证和记录调用日志处理。

其他请求参数由服务提供者根据业务系统需要自行定义，自定义参数命名不能和上述参数重复，用每个中文拼音的首字母相连接进行命名或者用英文进行命名，不要用无意义的字符进行命名。

（五）接口应答内容规范

为了便于系统统一管理，对返回内容做如下约定：

返回内容	说明	是否必须	类型
code	执行结果代码，详情请看说明。	是	字符串
message	执行结果信息，详情请看说明。	是	字符串
data	返回结果，详情请看说明。	否	数组对象
records	总记录数，详情请看说明。	否	数字
pages	总页数，详情请看说明。	否	数字

说明：

code: 执行结果代码，0为成功，其他为失败，其他错误代码请根据实际情况进行定义，并且需要在目录中心进行登记。

message: 执行结果信息，如果失败需要返回失败的详细情况。

data:返回的业务结果数据,类型是一个数组,单条记录和分页数据都须封装在该数组中。里面的数据项根据实际情况进行定义,并且需要在目录中心进行登记。数据项名称用每个中文拼音的首字母相连进行命名或者用英文进行命名,不要用无意义的字符进行命名。

records:总记录数,如果不是分页数据,不必返回。

pages:总页数,如果不是分页数据,不必返回。

为了便于服务使用方对数据的处理,服务提供方在统一目录中发布的时候需要把上述返回内容进行详细填写。

当内容中包含图片和文件,需要把图片和文件放到电子政务外网可以访问的服务器上,统一采用URL方式进行处理。

（六）其他约定

- 服务描述内容应包含服务调用者使用的具体访问接口内容,主要包括数据类型定义、操作、交互消息格式和内容、端口类型及绑定等内容,内容格式应按照 WSDL 1.1、SOAP 1.1/1.2 版本标准。
- 应按照 XML Schema 1.0 规范的规定使用; Message 应采用类型定义整个消息的数据结构。消息类型定义应按照 XMLSchema 1.0 规范的规定使用; 消息样式

Document Literal/Wrapped文档风格，禁止使用RPC Encoding及Document Encoded形式；消息字符集和编码应保持一致，应采用 GBK 或 UTF-8。PortType 应定义 Web 服务的操作，及操作中对应的输入输出参数和消息交互模式。

四、WEB 接口服务示例

（一）REST 接口示例

➤ 应答示例

```
18- /**
19  * 项目名称：四川省政务信息资源共享平台
20  * 类描述：REST 服务演示
21  * 创建人：xxxx
22  * 作者单位：省发展改革委
23  * 联系方式：xxxx
24  * 创建时间：2018-01
25  * @version 1.0
26  * 注意：本内容仅限于四川省政务信息资源共享平台，禁止外泄以及用于其他的商业目
27  */
28 @WebServlet({"rest/module1/getDetail/v1"})
29 public class RestDemoService extends HttpServlet {

30-     protected void doGet(HttpServletRequest request,
31         HttpServletResponse response) throws ServletException, IOException {
32         String deptName = request.getParameter("deptName");
33         String programName = request.getParameter("programName");
34         String username = request.getParameter("username");
35         String password = request.getParameter("password");
36         String arg1 = request.getParameter("arg1");
37
38         /*验证服务接口输入信息*/
39         if (deptName == null || deptName.trim().length() < 1) {
40             ResponseUtils.responseResult(response, new Result("1", "调用部门不能为空"));
41             return;
42         }

```

```

44     if (programName == null || programName.trim().length() < 1) {
45         ResponseUtils.responseResult(response, new Result("1", "调用程序不能为空"));
46         return;
47     }
48
49     /**
50     * 此处根据需要做验证处理
51     */
52
53     /**
54     * 此处根据需要做认证处理
55     */
56
57     /*下面是业务逻辑的处理，请根据业务进行编码*/
58     try{
59         MyBeanManager demoManager = new MyBeanManager();
60         List<MyBean> data = demoManager.getMyBean(arg1);
61         ResponseUtils.responseResult(response, new Result("0", "", data));
62
63         /**
64         * 此处根据需要做日志处理
65         */
66     }
67     catch(Exception ex){
68         ex.printStackTrace();
69         //代码根据实际情况返回
70         ResponseUtils.responseResult(response, new Result("2", ex.getMessage()));
71         /**
72         * 此处根据需要做日志处理
73         */
74     }
75 }

```

➤ 请求示例

```

20- /*
21- * 调用演示
22- */
23- public void getModuleDetail(){
24-     //服务发布者提供的URL地址
25-     String url = "http://localhost:8888/sfzggw/rest/module1/getDetail/v1";
26-
27-     //服务接口需要的参数
28-     Map<String, String> args = new HashMap<String, String>();
29-     args.put("deptName", "省发展改革委");
30-     args.put("programName", "统一管理系统");
31-     args.put("username", "johnny");
32-     args.put("password", "666666");
33-     args.put("arg1", "1");
34-
35-     //调用服务接口
36-     String result = HttpUtils.get(httpClient, url, args);
37-
38-     //根据需要处理返回数据
39-     System.out.println("result : " + result);
40- }

```

（二）WebService 接口示例

➤ 应答示例

```
6- /**
7  * 项目名称：四川省政务信息资源共享平台
8  * 类描述：WebService服务演示
9  * 创建人：xxxx
10 * 作者单位：省发展改革委
11 * 联系方式：xxxx
12 * 创建时间：2018-01
13 *
14 * @version 1.0 注意：本内容仅限于四川省政务信息资源共享平台，禁止外泄以及用于其他的商业目
15 */
16
17 @WebService
18 public interface WsDemoService {
19
20     /**
21      * 定义接口
22      */
23     public abstract String getDetail(
24         @WebParam(name = "deptName") String deptName,
25         @WebParam(name = "programName") String programName,
26         @WebParam(name = "username") String username,
27         @WebParam(name = "password") String password,
28         @WebParam(name = "arg1") String arg1);
29 }
```

```
10- /**
11 * 项目名称：四川省政务信息资源共享平台
12 * 类描述：REST 服务演示
13 * 创建人：xxxx
14 * 作者单位：省发展改革委
15 * 联系方式：xxxx
16 * 创建时间：2018-01
17 * @version 1.0
18 * 注意：本内容仅限于四川省政务信息资源共享平台，禁止外泄以及用于其他的商业目
19 */
20 public class WsDemoServiceImpl implements WsDemoService{
21
22     /**
23      * 获取单条数据演示。
24      * @param name 输入参数-名称
25      */
26     @Override
27     public String getDetail(String deptName, String programName,
28         String username, String password, String arg1) {
29
30         Gson gson = new Gson();
31         /*验证服务接口输入信息*/
32         if (deptName == null || deptName.trim().length()<1) {
33             return gson.toJson(new Result("1", "调用部门不能为空"));
34         }
```

```
36         if (programName == null || programName.trim().length()<1) {
37             return gson.toJson(new Result("1", "调用程序不能为空"));
38         }
39
40         /**
41          * 此处根据需要验证处理
42          */
43
44         /**
45          * 此处根据需要认证处理
46          */
47
48         /*下面是业务逻辑的处理，请根据业务进行编码*/
49         try{
50             MyBeanManager demoManager = new MyBeanManager();
51             List<MyBean> data = demoManager.getMyBean(arg1);
```

```

53      /**
54       * 此处根据需要做日志处理
55       */
56       return gson.toJson(new Result("0", "", data));
57     }
58     catch(Exception ex){
59       ex.printStackTrace();
60     }
61     /**
62     * 此处根据需要做日志处理
63     */
64     //代码根据情况返回
65     return gson.toJson(new Result("2", ex.getMessage()));
66   }
67 }
68 }

```

```

localhost:8888/sfzggw/webservice/module2/getDetail/v1?wsdl
<?xml:definition xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wedl="http://schemas.xmlsoap.org/wsdl/" xmlns:tns="http://ws.demo.fv.com/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns1="http://schemas.xmlsoap.org/soap/http" name="WsDemoServiceImplService" targetNamespace="http://ws.demo.fv.com/">
  <wedl:types>
    <xs:schema xmlns:xsi="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://ws.demo.fv.com/" elementFormDefault="unqualified" targetNamespace="http://ws.demo.fv.com/" version="1.0">
      <xs:element name="getDetail" type="tns:getDetail"/>
      <xs:element name="getDetailResponse" type="tns:getDetailResponse"/>
      <xs:complexType name="getDetail">
        <xs:sequence>
          <xs:element minOccurs="0" name="deptName" type="xs:string"/>
          <xs:element minOccurs="0" name="progrName" type="xs:string"/>
          <xs:element minOccurs="0" name="username" type="xs:string"/>
          <xs:element minOccurs="0" name="password" type="xs:string"/>
          <xs:element minOccurs="0" name="arg1" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="getDetailResponse">
        <xs:sequence>
          <xs:element minOccurs="0" name="return" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
  </wedl:types>
  <wedl:message name="getDetail">
    <wedl:part element="tns:getDetail" name="parameters"/>
  </wedl:message>
  <wedl:message name="getDetailResponse">
    <wedl:part element="tns:getDetailResponse" name="parameters"/>
  </wedl:message>
  <wedl:portType name="WsDemoService">
    <wedl:operation name="getDetail">
      <wedl:input message="tns:getDetail" name="getDetail"/>
      <wedl:output message="tns:getDetailResponse" name="getDetailResponse"/>
    </wedl:operation>
  </wedl:portType>
  <wedl:binding name="WsDemoServiceImplServiceSoapBinding" type="tns:WsDemoService">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wedl:operation name="getDetail">
      <soap:operation soapAction="" style="document"/>
      <wedl:input name="getDetail">
        <soap:body use="literal"/>
      </wedl:input>
      <wedl:output name="getDetailResponse">
        <soap:body use="literal"/>
      </wedl:output>
    </wedl:operation>
  </wedl:binding>
  <wedl:service name="WsDemoServiceImplService">
    <wedl:port binding="tns:WsDemoServiceImplServiceSoapBinding" name="WsDemoServiceImplPort">
      <soap:address location="http://localhost:8888/sfzggw/webservice/module2/getDetail/v1"/>
    </wedl:port>
  </wedl:service>
</wedl:definitions>

```

➤ 请求示例

```

7=  /**
8=  * 调用演示
9=  */
10= public void getModuleDetail(){
11=     //服务类是由wsimport工具生成的
12=     WsDemoServiceImplService factory = new WsDemoServiceImplService();
13=     WsDemoService wsImpl = factory.getWsDemoServiceImplPort();
14=
15=     //调用服务接口方法，并传入参数
16=     String result = wsImpl.getDetail("省发展改革委", "统一管理系统", "johnny", "666666", "1");
17=
18=     //根据需要处理返回数据
19=     System.out.println(result);
20= }

```