

生态环境数据接入规范

Access specification of ecology and environment data

2021 - 12 - 14 发布

2022 - 06 - 01 实施

江西省市场监督管理局

发布

目 次

前言 II

1 范围 1

2 规范性引用文件 1

3 术语和定义 1

4 接入范围 1

5 接入方式 1

6 接入要求 2

7 接入流程 2

附录 A（资料性）接口示例 3

前 言

本文件按照GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由江西省生态环境厅提出并归口。

本文件起草单位：江西省生态环境科学与规划研究院、江西省生态环境监测中心、江西省质量和标准化研究院、江西省华赣环境集团有限公司。

本文件主要起草人：江驰、谭文津、雷宁宁、谢雷、淦鑫、徐靓、曾建华、张仰硕、吴仁都、汪秋雨、余君、冷滨、黄振、曹卓冰、陈雪威、付忠诚、李苏发、陈英、谢岩、胡昊旻、李雪。

生态环境数据接入规范

1 范围

本文件规定了生态环境数据的术语和定义、接入范围、接入方式、接入要求、接入流程。
本文件适用于江西省内生态环境数据的接入与应用。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 22239 信息安全技术 网络安全等级保护基本要求
HJ 719 环境信息系统数据库访问接口规范
HJ 727 环境信息交换技术规范
DB36/T 981 电子政务共享数据统一交换平台技术规范
DB36/T 1179 政务数据共享技术规范

3 术语和定义

下列术语和定义适用于本文件。

3.1

数据提供方 data provider

在一次数据接入过程中，负责提供生态环境数据的组织或单位。

3.2

数据需求方 data demander

在一次数据接入过程中，负责接收生态环境数据的组织或单位。

4 接入范围

4.1 生态环境数据平台接入国家、省、设区市、县区生态环境行政主管部门和其他各级机构生态环境相关的数据。

4.2 省、设区市、县区生态环境数据应逐级接入或回流。

4.3 其他各级机构数据接入同级生态环境数据平台。

5 接入方式

5.1 类型

数据接入方式有数据文件、数据中间库和数据接口三种，针对不同的数据及需求采用相应的接入方式。

5.2 数据文件

通过电子邮件或FTP方式传输生态环境数据文件。

5.3 数据中间库

通过数据中间库交换生态环境数据，数据交换应符合HJ 719、HJ 727、DB36/T 981、DB36/T 1179等文件要求。

5.4 数据接口

5.4.1 数据获取

按照数据提供方发布的数据接口要求，调用接口服务，获取生态环境数据。

5.4.2 接口要求

提供数据接口应符合以下要求（接口示例见附录A）：

- a) 接口名称：此处描述接口中文含义；
- b) 接口地址：此处描述接口服务地址；
- c) 接口说明：对接口含义进行详细描述；
- d) 接口协议：http、WebService等；
- e) 备注：接口需备注的内容；
- f) 输入参数：包括参数名称、参数值、验证码等；
- g) 返回结果：包括参数名称、描述、状态等。

6 接入要求

6.1 对数据接入应进行身份及权限验证。

6.2 对数据分类分级，明确接入范围。

6.3 接入应符合 GB/T 22239、DB36/T 981、DB36/T 1179 等文件要求。

6.4 对数据进行校验，保证数据的准确。

6.5 通过数据中间库或数据接口方式进行接入的，应对数据增加主键、时间戳。

7 接入流程

7.1 数据需求方提出数据接入申请。

7.2 数据提供方根据申请分析接入的数据项及用途，确认接入方式，并提供相应的数据字典、数据接口说明文档等。

7.3 数据提供方对提供的数据接口、数据中间库及数据文件进行监控、维护，确保数据接入持续有效。

附 录 A
(资料性)
接口示例

A.1 接口示例一

A.1.1 接口描述

接口名称	通用接口
接口地址	http://{ip}:{port}/service/ws/jsonWebService?wsdl
接口协议	WebService
接口方法	RunJsonResult
备注	所有的接口返回的都是最新时间的数据

A.1.2 输入参数

参数名称	描述	类型	是否必填	示例
TOKEN	授权码	String	是	"ynhbt"
INTERFACEID	接口 ID	String	是	"8aaab3db5e97eecb015e993bb2d40122"
PAGER	分页; page: 当前页数 (从 1 开始); pageSize: 每页条数	String	是	{"pager":{"page":"","pageSize":""}}
PARAMS	fieldName: 参数名称; value: 查询参数值; operator: 匹配关系默认为 "="。 注: 凡是返回的字段都可以作为查询条件使用, 按照这种格式直接拼接	String	是	{"params":[{"fieldName":"行政区", "value":"02700001", "operator":"="}, {"fieldName":"企业名称", "value":"南昌", "operator":"="}]}

A.1.3 返回结果

参数名称	描述	示例
head	total: 总数; message: 响应信息; state: 状态 (true 成功; false 失败);	{"data":[{"STANDENTERID":"100",...}], "head":{"total":5612, "message":null, "interfaceId":"200", "state":true}}
data	数据信息; 格式: 字段名: 值	

A.1.4 WebService调用示例

调用 java 类示例:
//服务地址

```
String serverUrl="http://{ip}:{port}/service/ws/jsonWebService?wsdl";
//params参数
String params="{\"pager\":{\"page\":\"1\",\"pageSize\":\"10\"},\"params\": [{\"value\":\"\",\"fieldName\":\"POINTTYPE\",\"operator\":\"=\"}, {\"value\":\"360200000000\",\"fieldName\":\"CODE_REGION\",\"operator\":\"=\"}]}";
//接口方法
String method = "RunJsonResult";
//接口ID
String interfaceId = "8aaab3db5e97eecb015e993bb2d40122";
//授权码
String token = "ynhbt"
Client c=new Client(new URL(serverUrl));
Object[] o=c.invoke(method, new String[] {interfaceId, token, params});
System.out.println(o[0].toString());
```

A.2 接口示例二

A.2.1 接口描述

接口名称	县级饮用水站监测小时数据接口
接口地址	http://{ip}:{port}/{context}/serviceinterface/search/run.action
接口协议	WebService
接口方法	getMonitoringWaterStation
备注	县级饮用水站监测小时数据

A.2.2 输入参数

参数名称	描述	类型	是否必填	示例
TOKEN	授权应用	String	是	"ynhbt"
INTERFACEID	接口 ID	String	是	"8aaab3db5e97eecb015e993bb2d40122"
POINTCODE	测点编码	String	否	
ENDTIME	结束时间	String	是	"yyyy-MM-dd HH:mm:ss"
CODEREGION	行政区域代码	String	否	
STARTTIME	开始时间	String	是	"yyyy-MM-dd HH:mm:ss"

A.2.3 返回结果

参数名称	描述	示例
head	total: 总数; message: 响应信息; state: 状态 (true 成功; false 失败);	{"head":{"total":"0","state":"true/false","message":""},"data":[{"POINTCODE":"","PKID":"","RegionName":"","POLLUTENAME":"","POLLUTEVALUE":"","CODE_

参数名称	描述	示例
data	PKID: 主键 POINTCODE: 测点编码 POINTNAME: 测点名称 CODE_REGION: 行政区域代码 RegionName: 所在地市 MONITORTIME: 监测时间 CODE_POLLUTE: 污染物编码 POLLUTENAME: 污染物名称 POLLUTEVALUE: 监测值	REGION:"", "CODE_POLLUTE": "", "POINTNAME": "", "MONITORTIME": ""}}

A.2.4 WebService调用示例

```
public void wsTest() throws Exception{
    String param =
    "{\"data\": {\"TOKEN\": \"xxxx\", \"POINTCODE\": \"xxxx\", \"ENDTIME\": \"xxxx\", \"CODEREGION\": \"xxxx\", \"STARTTIME\": \"xxxx\"}}";
    Client c = new Client(new URL("http://{ip}:{port}/service/ws/jsonWebService?wsdl"));
    Object[] o = c.invoke("RunJsonResult", new String[] {"模板id", "TOKEN", param});
    System.out.println(o[0].toString());
}
```

A.3 接口示例三

A.3.1 接口描述

接口名称	公共代码接口
接口地址	http://{ip}:{port}/{context}/serviceinterface/search/run.action
接口协议	WebService
接口方法	getPublicCode
备注	公共代码数据

A.3.2 输入参数

参数名称	描述	类型	是否必填	示例
TOKEN	授权应用	String	是	"ynhbt"
INTERFACEID	接口 ID	String	是	"8aaab3db5e97eecb015e993bb2d40122"
REGIONCODE	行政区代码	String	否	

A.3.3 返回结果

参数名称	描述	示例
head	total: 总数; message: 响应信息; state: 状态 (true 成功; false 失败);	
data	REGIONCODE: 行政区代码 REGIONNAME: 所属行政区名称 PARENTCODE: 父级行政区编码 PROVINCECODE: 所属省份代码 CITYCODE: 所属地市代码 COUNTYCODE: 所属区县代码 TOWNCODE: 所属镇乡代码 ALLREGINNAME: 行政区全名	{ "head": { "total": "0", "state": "true/false", "message": "" }, "data": [{ "TOWNCODE": "", "PROVINCECODE": "", "ALLREGINNAME": "", "CITYCODE": "", "COUNTYCODE": "", "PARENTCODE": "", "REGIONCODE": "", "REGIONNAME": "" }] }] }

A.3.4 WebService调用示例

```
public void wsTest() throws Exception{  
    String param = "{\\\"data\\\":{\\\"TOKEN\\\":\\\"xxx\\\",\\\"REGIONCODE\\\":\\\"xxx\\\",}}\"";  
    Client c = new Client(new URL("http://{ip}:{port}/service/ws/jsonWebService?wsdl"));  
    Object[] o = c.invoke("RunJsonResult", new String[] {"模板id","TOKEN",param});  
    System.out.println(o[0].toString());  
}
```