**HCSPXTGL**

**水循环模拟平台系统工程标准**

**Q/SL 100.03-2016**

**水循环模拟平台业务信息交换标准**

**2019-06-01发布 2019-06-01实施**

**南京水利科学研究院**

**前言**

参考软件工程信息交换方面的相关标准规范并结合水循环模拟平台系统的实际情况，制定出《水循环模拟平台业务信息交换标准》。

本标准体系主要包括以下内容：

——信息交换协议，包括：请求报文和响应报文；

——信息交换格式结构；

——信息交换报文定义。

**目次**

[1 总则 4](#_Toc26278095)

[1.1 目的 4](#_Toc26278096)

[1.2 范围 4](#_Toc26278097)

[2 引用标准 4](#_Toc26278098)

[3 术语和定义 5](#_Toc26278099)

[3.1 JSON 5](#_Toc26278100)

[3.2 HTTP 5](#_Toc26278101)

[4 信息交换协议 5](#_Toc26278102)

[4.1 请求报文 5](#_Toc26278103)

[4.2 响应报文 8](#_Toc26278104)

[5 信息交换格式结构 11](#_Toc26278105)

[5.1 JSON语法规则 11](#_Toc26278106)

[5.1.1 对象结构 11](#_Toc26278107)

[5.1.2 数组结构 12](#_Toc26278108)

[6 信息交换格式定义 13](#_Toc26278109)

[6.1 JSON报文格式定义 13](#_Toc26278110)

[6.2 JSON报文格式描述： 13](#_Toc26278111)

[附录A信息交换协议报文示例 14](#_Toc26278112)

[附录B JSON报文示例 15](#_Toc26278113)

# 总则

## 目的

制订信息交换与共享的标准格式，能够屏蔽应用系统之间或外部模型各自之间的差异性，制定模型接口、系统内部信息交换体系，解决了不同时空尺度的水循环专业模型无缝链接，实现了系统的开放性和可扩展性，使得水循环模拟平台系统适应我国不同区域。

## 范围

信息交换技术，是数据交换平台的模型集成中间件的核心关键技术之一，解决多应用系统和模型服务之间根据相互关系和访问需求建立有序的信息交换流程的问题。规范统一的开发和运行架构，为系统与模型整合奠定技术基础，同时为满足未来业务发展提供技术保障。数据资源分析、规划、详细设计和系统实施过程中制定与本项目有关的业务数据标准。需要向多应用系统和模型服务制定信息传输的要求和信息交换的格式, 包括报文结构、传输约定、数据定义和数据组成。

本文主要定义系统内部程序信息交换标准，文档数据结构标准可参考数据结构标准文档。

# 引用标准

下列文件中的条款通过本规范的引用而成为本规范的条款。

《计算机软件文档编制规范》（GB8567-2006）

《标准化工作导则第一单元：标准的起草与表述规则》（GB-T 1 3-1997）

《标准化工作导则符号、代号标准编号规定》（GB/1.5）

凡是注日期的引用文件，其随后所有的修改单（不包括勘误的内容）或修订版均不适用于本规范。凡是不注日期的引用文件，其最新版本适用于本规范。

# 术语和定义

下列术语和定义适用于本文件。

## JSON

[JavaScript](https://baike.baidu.com/item/JavaScript) Object Notation，是一种轻量级的数据交换格式。它基于 [ECMAScript](https://baike.baidu.com/item/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。简洁和清晰的层次结构使得 JSON 成为理想的数据交换语言。

## HTTP

Hypertext Transfer Protocol，超文本传输协议是应用层协议。HTTP 是一种请求/响应式的协议，即一个客户端与[服务器](http://server.chinabyte.com/)建立连接后，向服务器发送一个请求;服务器接到请求后，给予相应的响应。

# 信息交换协议

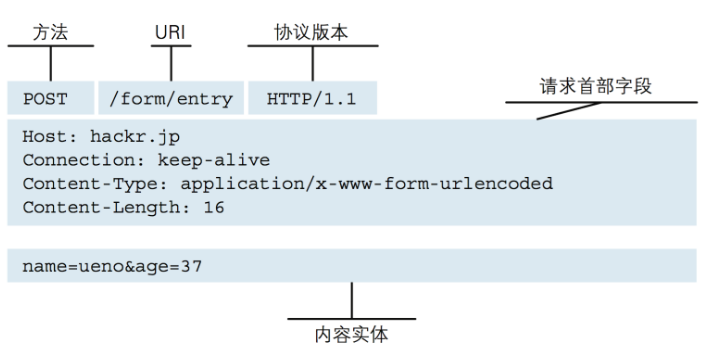
信息交换规则遵从HTTP协议，共分为请求报文、响应报文两部分。

## 请求报文

HTTP请求报文由**请求行、请求头部、空行和请求体**四个部分组成，如下图所示：



图表 1请求报文格式



图表 2请求报文的构成

报文描述：

请求行：请求行由方法字段、URL 字段 和HTTP 协议版本字段 3 个部分组成，他们之间使用空格隔开。常用的 HTTP 请求方法有 GET、POST、HEAD、PUT、DELETE、OPTIONS、TRACE、CONNECT;

**● GET：当客户端要从服务器中读取某个资源时，使用GET 方法。**GET 方法要求服务器将URL 定位的资源放在响应报文的部分，回[送给](http://www.chinabyte.com/keyword/%E9%80%81%E7%BB%99/)客户端，即向服务器请求某个资源。使用GET 方法时，请求参数和对应的值附加在 URL 后面，利用一个问号(“?”)代表URL 的结尾与请求参数的开始，传递参数长度受限制。例如，/index.html?id=100&op=bind。

**● POST：当客户端给服务器提供信息较多时可以使用POST 方法，**POST 方法向服务器提交数据，比如完成表单数据的提交，将数据提交给服务器处理。**GET 一般用于获取/查询资源信息，POST 会附带用户数据，一般用于更新资源信息。**POST 方法将请求参数[封装](http://www.chinabyte.com/keyword/%E5%B0%81%E8%A3%85/)在HTTP 请求数据中，以名称/值的形式出现，可以传输大量数据;

　　请求头部：请求头部由关键字/值对组成，每行一对，关键字和值用英文冒号“:”分隔。请求头部通知服务器有关于客户端请求的信息，典型的请求头有：

　　● User-Agent：产生请求的[浏览器](http://www.chinabyte.com/keyword/%E6%B5%8F%E8%A7%88%E5%99%A8/)类型;

　　● Accept：客户端可识别的响应内容类型列表;星号 “ \* ” 用于按范围将类型分组，用 “ \*/\* ” 指示可接受全部类型，用“ type/\* ”指示可接受 type 类型的所有子类型;

　　● Accept-Language：客户端可接受的自然语言;

　　● Accept-Encoding：客户端可接受的编码压缩格式;

　　● Accept-Charset：可接受的应答的字符集;

　　● Host：请求的主机名，允许多个[域名](http://www.chinabyte.com/keyword/%E5%9F%9F%E5%90%8D/)同处一个IP 地址，即虚拟主机;

　　● connection：连接方式(close 或 keepalive);

　　● Cookie：[存储](http://storage.chinabyte.com/)于客户端扩展字段，向同一域名的服务端发送属于该域的cookie;

　　空行：最后一个请求头之后是一个空行，发送回车符和换行符，通知服务器以下不再有请求头;

　　请求体：请求包体不在 GET 方法中使用，而是在POST 方法中使用。POST 方法适用于需要客户填写表单的场合。与请求包体相关的最常使用的是包体类型 Content-Type 和包体长Content-Length;

## 响应报文

**HTTP 响应报文由状态行、响应头部、空行和响应体**四个部分组成，如下图所示：



图表 3响应报文格式

报文描述：

状态行：状态行由 HTTP 协议版本字段、状态码和状态码的描述文本三个部分组成，他们之间使用空格隔开;

　　● 状态码由三位数字组成，第一位数字表示响应的类型，常用的状态码有五大类如下所示：

　　1xx：表示服务器已接收了客户端请求，客户端可继续发送请求;

　　2xx：表示服务器已成功接收到请求并进行处理;

　　3xx：表示服务器要求客户端重定向;

　　4xx：表示客户端的请求有非法内容;

　　5xx：表示服务器未能正常处理客户端的请求而出现意外错误;

　　● 状态码描述文本有如下取值：

　　200 OK：表示客户端请求成功;

　　400 Bad Request：表示客户端请求有语法错误，不能被服务器所理解;

　　401 Unauthonzed：表示请求未经授权，该状态代码必须与 WWW-Authenticate 报头域一起使用;

　　403 Forbidden：表示服务器收到请求，但是拒绝提供服务，通常会在响应正文中给出不提供服务的原因;

　　404 Not Found：请求的资源不存在，例如，输入了错误的URL;

　　500 Internal Server Error：表示服务器发生不可预期的错误，导致无法完成客户端的请求;

　　503 Service Unavailable：表示服务器当前不能够处理客户端的请求，在一段时间之后，服务器可能会恢复正常;

　　响应头部：响应头可能包括：

　　Location：Location响应报头域用于重定向接受者到一个新的位置。例如：客户端所请求的页面已不存在原先的位置，为了让客户端重定向到这个页面新的位置，服务器端可以发回Location响应报头后使用重定向语句，让客户端去访问新的域名所对应的服务器上的资源;

　　Server：Server 响应报头域包含了服务器用来处理请求的软件信息及其版本。它和 User-Agent 请求报头域是相对应的，前者发送服务器端软件的信息，后者发送客户端软件(浏览器)和[操作系统](http://soft.chinabyte.com/os/)的信息。

　　Vary：指示不可缓存的请求头列表;

　　Connection：连接方式;

　　对于请求来说：close(告诉[WEB服务器](http://www.chinabyte.com/keyword/web%E6%9C%8D%E5%8A%A1%E5%99%A8/)或者[代理服务器](http://www.chinabyte.com/keyword/%E4%BB%A3%E7%90%86%E6%9C%8D%E5%8A%A1%E5%99%A8/)，在完成本次请求的响应后，断开连接，不等待本次连接的后续请求了)。keepalive(告诉WEB服务器或者代理服务器，在完成本次请求的响应后，保持连接，等待本次连接的后续请求);

　　对于响应来说：close(连接已经关闭); keepalive(连接保持着，在等待本次连接的后续请求); Keep-Alive：如果浏览器请求保持连接，则该头部表明希望WEB 服务器保持连接多长时间(秒);例如：Keep-Alive：300;

　　WWW-Authenticate：WWW-Authenticate响应报头域必须被包含在401 (未授权的)响应消息中，这个报头域和前面讲到的Authorization 请求报头域是相关的，当客户端收到 401 响应消息，就要决定是否请求服务器对其进行验证。如果要求服务器对其进行验证，就可以发送一个包含了Authorization 报头域的请求;

　　空行：最后一个响应头部之后是一个空行，发送回车符和换行符，通知服务器以下不再有响应头部。

　　响应体：服务器返回给客户端的文本信息;

# 信息交换格式结构

## JSON语法规则

JSON是一个序列化的对象或数组。

### 对象结构

对象结构是使用大括号“{}”括起来的，大括号内是由0个或多个用英文逗号分隔的“关键字:值”对（key:value）构成的。

语法：

|  |
| --- |
| var jsonObj =  {  "键名1":值1,  "键名2":值2,  ……  "键名n":值n  } |

图表 4对象结构

### 数组结构

JSON数组结构是用中括号“[]”括起来，中括号内部由0个或多个以英文逗号“,”分隔的值列表组成。

语法：

|  |
| --- |
| var jsonArr =  [  {  "键名1":值1,  "键名2":值2  },  {  "键名3":值3,  "键名4":值4  },  ……  ] |

图表 5数组结构

# 信息交换格式定义

## JSON报文格式定义

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **属性** | **名称** | **类型** | **是否为空** | **说明** |
| result | 处理结果 | Number | 是 |  |
| data | 数据对象 | Object | 否 |  |
| msg | 描述信息 | String | 否 |  |

图表 6JSON报文格式定义

## JSON报文格式描述：

|  |
| --- |
| {  "result" : 0,  "data" : {},  "msg" : ""  } |

图表 7JSON报文内容

JSON报文由三部分组成处理结果、数据对象和描述信息。

处理结果的值由数字组成，1表示调用接口成功,0表示调用失败,-1表示服务器异常,-2表示输入参数有误,-3 表示token校验失败；

数据对象的值可以是一个对象、字符串、数组，当处理结果的值为非1时，该对象的值返回空字符串；

描述信息的值是选填项，值是字符串类型的字段，主要用来描述接口调用描述的信息。

以上三部分共同组成范文的数据结构，报文必须包含这三个属性。

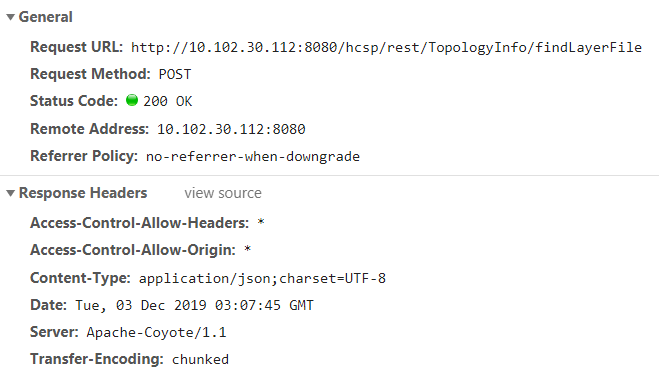
# 附录A信息交换协议报文示例

请求报文示例如下图：



图表 8请求报文示例

响应报文示例如下图：



图表 9响应报文示例

# 附录B JSON报文示例

JSON格式的响应报文如下表所示：

|  |
| --- |
| {  "result" : 1,  "data" : {  "XAJ1" : {  "fileTypeCode" : 5,  "axes" : [ {  "paramattr" : "999",  "locations" :  "119.98961092449909,31.992246765041305;119.99785067984402,32.01916026596977;119.99098422019465,32.0682881600977;119.955278621221,32.122066970235565;119.94703886899809,32.16763494093086;119.94429228524857,32.19099406535421;119.94566557840179,32.21084422574403;119.9415457024781,32.2388610877344;119.93605253423998,32.268036153312195;119.93467924337003,32.30536591041559;119.93467924415553,32.325190966350824;119.93467924553993,32.36016567026508;119.9374258308141,32.37415148135741;119.93330595446895,32.39279653420861;119.9113332773343,32.406779454555206;119.87837426160624,32.41843144296841;119.85365499985197,32.42076316649463;119.81657610728743,32.41261216529122;119.79597672276925,32.407954017359884;119.76851087705401,32.403296113128725;119.7492847851176,32.396307073797885;119.73829844671758,32.384655867724305;119.69435309545183,32.40679654719334;119.65452762094559,32.410294167179565;119.62843506871145,32.402140906629604;119.6201953150297,32.38233339581434;119.63530152781243,32.31938367093907;119.64216798820593,32.29255829196386;119.66002078676362,32.28555797953451;119.65727420165243,32.25755620654976;119.65727420105179,32.233047338506665;119.66276736925617,32.20736385286331;119.67100712214327,32.192183465920316;119.69709967422267,32.18984608236748;119.71907234978048,32.1863410355799;119.72868539529657,32.18166880952103;119.74379160980861,32.17699593094816;119.76301770076307,32.15947318738913;119.75203136198904,32.13260138771542;119.74791148471733,32.11390298050333;119.75340465310582,32.09403095351499;119.78499037434106,32.07766045639304;119.80009658886047,32.06713534613168;119.82618914161353,32.05660811616438;119.8344288950338,32.04958999750165;119.86052144826854,32.04490924832647;119.87150778560411,32.01683163340588;119.89760033785872,31.97820978086209;119.92369289125651,31.965330629640572;119.94291898366546,31.96298745932471;119.97313141538702,31.97117946419031;119.98961092403684,31.98171298024452;119.98823763204129,31.98990621033436",  "id" : "wxsm-5c90ee6c3dc747ed9203d98b4e8d248c",  "mainid" : "wxsm-886f63cd997543fea51a0c23b4e91e8f",  "blockid" : "layer",  "alias" : "",  "elementName" : "区块",  "elementId" : "1980",  "styleId" : "winfo\_bd4baf92e714ca11e993b054e1add85a5d"  } ]  }  },  "msg" : "查询成功"  } |

图表 10响应JSON报文示例