**xService开放式服务技术指南**

**(xService OPEN SERVER GUIDE)**

**Release V1.6**

目 录

前言2

第一章 服务框架3

1.1xService服务框架3

1.2xService服务流程4

第二章 资源配置6

2.1xService服务配置6

2.2xService服务目录6

第三章 服务接口9

3.1xService接口概览9

3.2xService公共HTTP头定义9

3.3xService错误消息16

3.4xService审计服务17

3.5xServiceWEB服务18

3.6xService文件服务20

3.7xService云存储代理31

3.8xService数据库网关33

3.9xService短信网关49

3.10xService设备网关52

3.11xService支付网关52

**前 言**

本文档是xService系列开放式服务的技术指南，描述了xService的服务框架、资源管理、服务接口。文档的主要阅读对象为从事构建基于WEB API服务应用的开发商，目的在于协助其建立安全、可靠以及具备良好响应性能的通用http服务和企业应用服务。

xService可以根据应用场合需要配置成并发线程服务，或者配置成为并发进程服务。在某些特殊的情况下，如在64位的系统中，第三方只有32位的例程，xService也可以配置成64位守护进程＋32位服务进程。并发线程服务理论上具有调度方面的优势，如果应用需要配置为线程服务，xService建议组织在其服务目录中的例程采用xdl for c\c++ library开发，xdl实现了内存堆的私有化操作和线程崩溃保护。当工作线程结束后，私有堆会自动释放，不会因为程序申请和释放内存不匹配而导致内存泄漏的严重问题。如果工作线程意外终止，崩溃保护机制会隔离问题线程，确保其不影响主线程或进程的运行，并有机会最后释放问题线程的资源。在xService中，并发进程服务则更为健壮，在复杂的环境中，特别是在xService中部署了多个第三方服务例程的情况下，xService并不能防护第三方应用的内存泄漏和程序崩溃问题，因此采用子进程来调度第三方服务是安全的做法。

发布版的xService作为系统服务运行，可以使用Service Manger来管理。xService也可以编译为桌面应用，当作为普通桌面程序运行时，xService的功能与服务态下运行无异，但更容易跟踪调试，为开发阶段服务例程的调试提供便利。

xService不是面向浏览器的服务例程，而是面向应用的服务例程，因此xService不会主动返回html网页，除非是应用需要并特别声明。在企业应用中，xService返回的数据类型总是尽量按场景需要以及安全原则返回，比如对于请求同一个敏感资源，在安全的交易场合，xSerivice会返回application/xml原生数据供请求方处理，但在非安全的场合，可能会返回image/jpg或image/svg+xml数据仅供请求方浏览，主要是防止原生数据在非可信场合落地。

xService是跨平台和开源的应用。

项目地址：https://sourceforge.net/projects/easily-sdk-4/

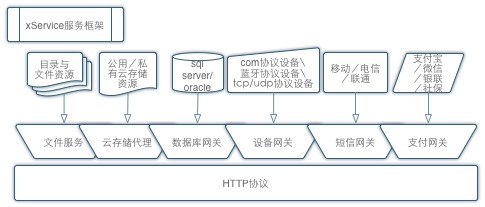
下载项目：git clone git://git.code.sf.net/p/easily-sdk-4/code easily-sdk-4-code

第一章 服务框架

xService是基于http协议构建的网络服务例程，各可用资源被组织在xService的服务目录中，客户端通过http的请求／应答方式，调用其公布的WEB API。

# ****xService服务框架****

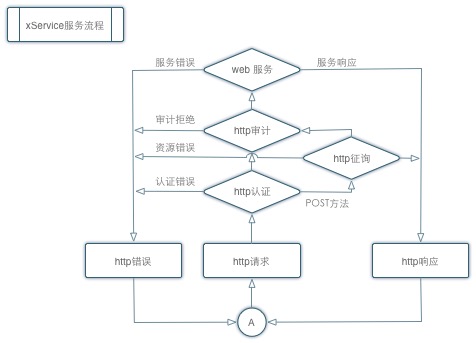
图例展示了xService支持的一些通用服务，专业服务可以参考其他以xService实现的web service服务平台的技术指南。



* 文件服务：xService可以将本地、共享网络、其他http文件目录，组织在文件服务的入口中，为客户端提供目录枚举、文件读写服务。
* 云存储代理：xService为第三方提供云存储的读写功能，实现的方式有两种，（1）转移授权给第三方，使其可以直接访问云存储资源。（2）为云存储缓存读写，第三方通过间接方式使用云存储资源。
* 数据库网关：xService可以将多个数据源组织在服务入口中，并为客户端提供一致的访问接口。客户端无需安装专有的数据库访问接口或工具，可以在同一例程中访问多个异构数据源。
* 设备网关：对于注册及组织在xService中的设备，xService将其虚拟成为ip设备，客户端可以通过ip地址，采用http请求／应答方式来访问或控制设备。xService负责将来自http请求中的设备控制指令转换为设备直连对话／应答协议，也可以将设备应答的内容转换为适合网络传输的文档（如xml文档），以此简化客户端与设备的交互过程，并使客户端具备远程控制设备的能力。
* 短信网关：多个运营商的短信服务接口可以组织在xService服务入口中，xService根据客户收发短信的号段来调用相应的运营商短信服务接口。
* 支付网关：xService支持点对点支付（一个运营商支付一次交易请求）和多对单支付（多个运营商支付一次交易请求）。xService为交易行为提供数字凭证和可视凭证，凭证可用于客户端通关验证。

# ****1.2 xService服务流程****

图例展示了xService的请求／应答交互例程。



* http请求：客户端根据xService服务地址和服务入口发出http请求。
* http认证：xService收到http请求后，先验证其Authorization字段，认证不匹配则返回认证错误。
* http征询：如果采用post方法调用服务，并且客户端请求中包含了Expect，xService在调度入口资源时，如果资源可用则先返回100 Continue响应状态，服务继续，否则返回资源错误，服务终止。
* http审计：xService根据请求方的服务入口，调用相应的审计例程，如果客户请求在审计策略中被判为违规，则返回审计拒绝。
* web 服务：xService根据请求方的服务入口，调用相应的服务例程，例程内部错误则返回服务错误，例程正确执行结束后返回http 200响应状态。

第二章 资源配置

xService服务框架中，可受托管理的资源有：文件资源、云存储资源、数据库资源、短信服务资源、支付服务资源、设备资源、专业服务资源等。资源均配置在xService的服务目录中，请求方根据不同的服务入口来调用xService的资源服务。

# ****2.1 xService服务配置****

xService一般安装于｛$Program Files｝\｛xService｝目录。xService的守护进程（xService）和服务进程（xhttps）通常安装于该目录下，运行xService和xhttps的动态库，均安装于该目录的子目录（api）之下，在系统环境变量$PATH中，需设置该搜索路径：｛$xService｝\api。

httpd.config是xService的初始化配置文件，以下描述了有关xService的启动配置项。

|  |  |  |
| --- | --- | --- |
| 名称 | 描述 | 选项 |
| port | 服务端口名，未定义值则按80端口启动服务。 |  |
| mode | 服务方式，定义了thread，xhttps以并发线程方式运行，未定义或定义为process，xhttps则以并发进程方式运行。 | ｛thread|  process} |
| module | 指定了并发进程方式下，xhttps的运行位置。 |  |
| root | 指定了xService的服务根目录。 |  |

# ****2.2 xService服务目录****

｛$xService｝目录下通常有如下服务目录：

* cfg：为xService服务入口的配置文件目录，每个服务入口都有其对应的配置文件｛$ServerName｝.config。xService在收到服务入口请求时，先从cfg目录下读取相应的入口配置文件，以获取相应服务例程的位置，并调用其声明的入口函数。

|  |  |  |
| --- | --- | --- |
| 名称 | 描述 | 选项 |
| module | 入口服务例程名 | 如loc\_api |
| entry | 入口函数 | 如xhttp\_invoke |
| path | 入口服务路径 | 如｛$xService｝/loc |
| trace | 入口服务例程的日志路径 | 如｛$xService｝/loc/log |

* lic：为xService接入许可证的文件目录，xService在收到http请求时，根据Authorization访问键查找该目录下的许可证文件，对请求方进行接入认证。

|  |  |  |
| --- | --- | --- |
| 名称 | 描述 | 选项 |
| PUBLIC-KEY | 公共密钥，即请求方的访问键 |  |
| PRIVATE-KEY | 私有密钥，请求方和响应方用此作为生成签名的密钥。 |  |

* api：运行xService所需要的动态库以及安装于xService的服务例程均存储在该目录下。
* loc：xService文件服务的入口，该目录下的loc.config配置文件中指明了读写文件的具体路径。

|  |  |  |
| --- | --- | --- |
| 名称 | 描述 | 选项 |
| path | xService文件读写服务的具体路径 | ｛本地路径｜  共享网络路径｜  HTTP网络路径｝ |

* oss：xService云存储代理的服务入口，该目录下的oss.config定义了访问云存储资源的参数配置。

|  |  |  |
| --- | --- | --- |
| 名称 | 描述 | 选项 |
| location | 云存储的内链或外链访问地址 |  |
| secrect-id | 云存储的访问键 |  |
| secrect-key | 云存储的访问密钥 |  |

* xdb：xService数据库网关的服务入口，子目录xdb\odbc和xdb\oci分别定义了odbc和oci数据源的访问入口，各自目录下的｛$dbname｝.dsn定义了可访问数据源的参数配置。

|  |  |  |
| --- | --- | --- |
| 名称 | 描述 | 选项 |
| DRIVER | ODBC数据库访问接口的驱动名 |  |
| DATABASE | 数据库名 |  |
| SERVER | 服务名或IP地址 |  |
| UID | 数据库用户名 |  |
| PWD | 数据库用户的密码 |  |

* sms：xService短信网关的服务入口，其目录下｛$运营商名｝.config定义了短信运营商的参数配置。

|  |  |  |
| --- | --- | --- |
| 名称 | 描述 | 选项 |
| address | 运营商的短信服务地址 |  |
| account | 运营商分配的短信服务账号 |  |
| password | 短信服务账号的密码 |  |

* dev：xService设备网关的服务入口，该目录下的com、bluetooth、socket子目录分别对应使用com通讯协议、蓝牙通讯协议、tcp/udp通讯协议的设备服务入口，各目录下｛$devname｝.ini定义了各设备的连接参数。
* pay：xService支付网关的服务入口，该目录下｛$运营商｝.config定义了各自支付服务的参数配置。
* ｛$web service name｝：xService中提供专业服务的web 服务入口，目录下包含各自定义的配置文件。

第三章 服务接口

# ****3.1接口概览****

* HTTP头定义：xServie所支持的HTTP头的定义以及适用的场景描述。
* xService错误消息：描述xService定义的错误处理例程返回给请求方的消息格式。
* xService审计服务：由xService和安装于xService的web服务共同实现的服务成本、数据安全方面的审计服务。
* xService WEB服务：在xService中可以注册安装各种基于xml+http的专业服务，并通过xService展示其服务接口。
* xService文件服务：由xService实现的基于http的文件读写服务。
* xService云存储代理：由xService为第三方提供云存储访问的转移授权和读写缓存服务。
* xService数据库网关：由xService实现的与数据库访问接口无关的数据库资源访问功能。
* xService短信网关：由xService实现的与运营商访问接口无关的短信收发功能。

# ****3.2公共HTTP头定义****

# ****3.2.1 HTTP请求头****

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 描述 | 场景 | 选项 |
| Accept | 请求方声明可接受的文档类型 | 用于请求返回文件或交易数据的业务场景 | ｛application/xml｜  application/pdf｜  application/soap+xml｜  text/xml｜  text/html｜  text/plain｜  image/jpeg｜  image/png｜  image/gif｜  image/bmp｜  image/svg+xml｜  video/mpeg｜  video/avi｜  audio/mp3｝ |
| Accept-Encoding | 请求方声明可接受的压缩传输编码类型 | 用于请求返回文件或交易数据的业务场景 | ｛gzip｜  deflate｝ |
| Accept-Charset | 请求方声明可接受的文档字符集编码类型 | 用于请求返回交易数据的业务场景 | ｛gb2312｜  utf-8｜  utf-16｝ |
| Authorization | 请求方的认证串 | 请求xService服务 | ｛XDS｝ {PUBLIC-KEY} {SINGNATURE} |
| Content-Encoding | 请求方采用的压缩传输编码类型 | 用于向服务端传送文件或交易数据的业务场景 | ｛gzip｜  deflate｝ |
| Content-Length | 请求方向服务端传送的数据实体的字节数 | 用于向服务端传送文件或交易数据的业务场景 |  |
| Content-Range | 请求方向服务端传送的数据实体的字节范围 | 用于向服务端上传文件的业务场景 | ｛from｝－｛to｝／｛total｝ |
| Content-Type | 请求方向服务端传送的数据实体的类型 | 用于向服务端传送文件或交易数据的业务场景 | ｛application/xml｜  application/pdf｜  application/soap+xml｜  text/xml｜  text/html｜  text/plain｜  image/jpeg｜  image/png｜  image/gif｜  image/bmp｜  image/svg+xml｜  video/mpeg｜  video/avi｜  audio/mp3｝ |
| Date | 请求方的时间戳 |  | UTC/GMT |
| Expect | 请求方征询服务或资源是否可用 | 用于使用POST方法向服务端传送文件或交易数据的业务场景 | ｛100-Continue｝ |
| Host | 请求方请求服务的域名/IP地址及端口号 |  | ｛xxx.xxx.xxx.xxx｝:｛nnnn｝ |
| If-Match | 请求方声明的文件ETag匹配条件，匹配则服务的接受请求，不匹配则拒绝请求 | 用于向服务端上传/下载文件业务场景 | ｛ETag｝ |
| If-Modified-Since | 请求方声明的文件最后修改时间的匹配条件，成立则服务的接受请求，不成立则拒绝请求 | 用于向服务端上传/下载文件业务场景 | ｛UTC/GMT｝ |
| If-None-Match | 请求方声明的文件ETag匹配条件，不匹配则服务的接受请求，匹配则拒绝请求 | 用于向服务端请求上传/下载文件业务场景 | ｛ETag｝ |
| Last-Modified | 请求方声明的文件修改日期 | 用于向服务端上传文件需要同步文件最后修改时间的业务场景 | ｛UTC/GMT｝ |
| Range | 请求方声明的文件字节区段 | 用于向服务端请求下载文件业务场景 | ｛bytes｝=｛from｝－｛to｝ |
| Transfer-Encoding | 请求方声明交易数据传送采用chunk方式 | 用于向服务端传送交易数据的业务场景 | ｛chunked｝ |
| xds-action | 请求方声明的WEB服务的动作（由xService扩展） | 请求xService服务 | ｛命名空间｝／｛动作名｝ |
| xds-landmark | 请求方声明的地标标识符（由xService扩展） | 请求xService服务。（xService使用地标识别来提供个性化或差异化服务） |  |

# ****3.2.2 HTTP响应头****

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 描述 | 场景 | 选项 |
| Accept-Ranges | 服务方指明是否返回了数据区段 | 用于服务端响应下载文件的业务场景 | ｛bytes｜none｝ |
| Content-Encoding | 服务方指明压缩传输编码类型 | 用于服务端响应下载文件的业务场景 | ｛gzip｜  deflate｝ |
| Content-Length | 服务方指明返回数据实体的字节数 | 用于服务端响应传送文件或返回交易数据的业务场景 |  |
| Content-Range | 服务方指明返回数据实体的字节范围 | 用于服务端响应下载文件的业务场景 | ｛from｝－｛to｝／｛total｝ |
| Content-Type | 服务方指明返回数据实体的类型 | 用于服务端响应交易返回数据的业务场景 | ｛application/xml｜  application/pdf｜  application/soap+xml｜  text/xml｜  text/html｜  text/plain｜  image/jpeg｜  image/png｜  image/gif｜  image/bmp｜  image/svg+xml｜  video/mpeg｜  video/avi｜  audio/mp3｝ |
| Connection | 服务方指明连接是否关闭 | 用于服务端响应传送文件或返回交易数据的业务场景 | ｛Close｝ |
| Date | 服务方的时间戳 |  | UTC/GMT |
| ETag | 服务方返回的文件ETag | 用于服务端响应传送文件的业务场景 | ｛ETag｝ |
| Last-Modified | 服务方返回的文件最后修改时间 | 用于服务端响应传送文件的业务场景 | UTC/GMT |
| Transfer-Encoding | 服务方采用chunk方式返回交易数据 | 用于服务端返回交易数据的业务场景 | ｛chunked｝ |
| Server | 服务方名称 |  |  |
| xds-trace | 服务方返回的交易标识符（由xService扩展） | 请求方可以使用交易标识符获取交易日志，追溯交易细节 |  |

# ****3.2.3 URL 转义****

请求方的url串中query部分在发送请求之前对以下字符需要进行转义。

|  |  |  |  |
| --- | --- | --- | --- |
| 字符 | 十进制 | 十六进制 | 转义串 |
| 空格 |  | 20 | %20 |
| “ |  | 22 | %22 |
| # |  | 23 | %23 |
| % |  | 25 | %25 |
| & |  | 26 | %26 |
| ( |  | 28 | %28 |
| ) |  | 29 | %29 |
| + |  | 2B | %2B |
| , |  | 2C | %2C |
| / |  | 2F | %2F |
| : |  | 3A | %3A |
| ; |  | 3B | %3B |
| < |  | 3C | %3C |
| = |  | 3D | %3D |
| > |  | 3E | %3E |
| ? |  | 3F | %3F |
| @ |  | 40 | %40 |
| \ |  | 5C | %5C |
| | |  | 7C | %7C |

# ****3.2.4 HTTP认证****

请求方在调用xService服务时，必须填写Authorization字段，认证串由（标志，访问健，签名值）组成。服务主管机构向接入方颁发（公用密钥、私有密钥），请求方使用公用密钥作为访问键，使用私有密钥对HTTP头部进行签名。xService根据公用密钥获取已颁发的私有密钥，对请求方的HTTP头部重新计算签名值，并与请求方传入的签名值进行比对，不匹配则拒绝服务。

**签名值（Signature）的算法**：

signature = base64(hmac\_sha1(PrivateKey,VERB + \n + Content-Type + \n + Date + \n + xds-action + \n + xds-landmark + \n + object))

**签名示例：**

服务方颁发的密钥（Public Key，Private Key）为：（a1c604122f29a147e6f8ba318b31da5e，c18cd8f512dc56941cb96cda3dbf4944）

请求方调用xml web服务：<HTTP://192.168.0.1:8000/myserver>，此时纳入签名计算各项的键值为：

**PrivateKey**：c18cd8f512dc56941cb96cda3dbf4944

**VERB**：POST

**Content-Type**：application/xml

**Date**：17,Nov 2015 18:49:58 GMT

**xds-action**：www.namespace.com/myaction

**xds-landmark：**

**object**：/myserver

签名值计算：

mysignature = base64(

hmac\_sha1(

“c18cd8f512dc56941cb96cda3dbf4944”,

“POST\napplication/xml\n17,Nov 2015 18:49:58 GMT\nwww.namespace.com/myaction\n\n/myserver”

)

)

组合签名串为：｛XDS｝｛空格｝｛Public Key｝｛冒号｝｛Signature｝

Authorization: XDS a1c604122f29a147e6f8ba318b31da5e:mysignature

# ****3.2.5 HTTP征询****

请求方在使用POST方法调用xService服务时，可以填写Expect字段，表明请求方希望先征询xService服务资源是否可用。如果请求方填写了Expect字段，xService在资源可用时返回100 Continue响应头，否则返回错误响应，请求方应在收到100 Continue状态后开始上传数据。

请求方先发送请求头：

*POST / HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS \*\*\*\*: ####*

*Content-Length:100*

*Expect: 100-Continue*

服务方首先返回：

*HTTP/1.1 100 Continue*

请求方上传数据实体：

｛数据实体｝

服务方最后返回：

*HTTP/1.1 200 Ok*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Connection: close*

*Server: xService*

# ****3.3 xService 错误消息****

HTTP应答码为2##的情况下，表明业务处理成功，HTTP消息体返回业务数据包，否则消息体返回错误消息包。以下为错误消息包格式。

<?xml version="1.0" encoding="UTF-8" ?>

<fault>

<fault-code>my fault code</fault-code>

<fault-string> my fault message</fault-string>

</fault>

请求示例：

*GET / HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS a1c604122f29a147e6f8ba318b31da5e:*

返回认证失败示例：

*HTTP/1.1 401 Unauthorized*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Content-type: application/xml*

*Content-Length: 111*

*Connection: close*

*Server: xService*

*xds-trace: 20151213012359*

<?xml version="1.0" encoding="UTF-8" ?>

<fault>

<fault-code>-1</fault-code>

<fault-string> The signature is: abcdefghigklmnokqrstuvwxwz=</fault-string>

</fault>

在出现交易错误的后续处理中，请求方可以根据xService在响应头中返回的xds-trace来获取交易日志，以追溯交易细节。xService的文件服务的根目录为LOC，交易日志一般存储在该/LOC/LOG/{服务入口名}/的目录下。

请求交易日志示例：

*GET /LOC/LOG/myservice/20151213012359.log HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS \*\*\*\*: ####*

交易日志返回示例：

*HTTP/1.1 200 Ok*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Content-type: text/plain; charset=gb2312;*

*Content-Length: 111*

*Connection: close*

*Server: xService*

｛日志文档实体｝

* **有关文件服务的请参见3.6 xService文件服务**

# ****3.4 xService 审计服务****

xService审计服务包括成本审计和安全审计，成本审计由xService的http服务实现，用于评估服务成本。安全审计由安装于xService的各web服务自主实现，主要用于隐私数据的保护。

审计服务通常返回放行／警告／拒绝三种判定。

审计服务由服务方实现，不需要请求方调用，在服务方判定审计拒绝的情况下，请求方将收到审计错误的响应。

请求示例：

*GET / HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS a1c604122f29a147e6f8ba318b31da5e:*

返回审计失败示例：

*HTTP/1.1 500 Internal Server Error*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Content-type: application/xml*

*Content-Length: 111*

*Connection: close*

*Server: xService*

*xds-trace: 20151213012359*

<?xml version="1.0" encoding="UTF-8" ?>

<fault>

<fault-code>-1</fault-code>

<fault-string> The aduiting rejected </fault-string>

</fault>

# ****3.5 xService WEB服务****

xService web服务访问入口定义为：｛http:｝//｛地址｝:｛端口号｝/｛服务名｝。

在xService的web服务体系中，服务方在xService的根目录中定义web服务入口，并注册相应服务供请求方调用，请求方在调用web服务时，需在http头xds-action项中指明动作。

xService的web服务一般采用xml表征业务交互数据。

xService web调用示例：web服务地址 <http://192.168.0.1:8000/myservice>

请求消息体：

*POST /myservice HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS \*\*\*\*: ####*

*Content-Type: application/xml*

*Content-Length: 111*

*Accept: application/xml*

*Accept-Charset: utf-8*

*xds-action:* [*www.namespace.com/myaction*](http://www.namespace.com/myaction)

<?xml version="1.0" encoding="UTF-8" ?>

<rowset>

<row>

<param1>-1</param1>

<param2> 2</param2>

</row>

<row>

<param1>-3</param1>

<param2> 4</param2>

</row>

<rowset>

返回消息体：

*HTTP/1.1 200 Ok*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Content-type: application/xml*

*Content-Length: 111*

*Connection: close*

*Server: xService*

*xds-trace: 20151213012359*

<?xml version="1.0" encoding="UTF-8" ?>

<rowset>

<row>

<result1>-1</result1>

<result2> 2</result2>

</row>

<row>

<result1>-3</result1>

<result2> 4</result2>

</row>

</rowset>

# ****3.6 xService 文件服务****

xService文件服务访问入口定义为：｛http:｝//｛地址｝:｛端口号｝/｛loc｝/｛对象名｝。

xService提供目录列举、文件属性、文件读写服务，对于大容量文件（8M以上）xService支持区段读写，请求方可以籍此实现断点续传功能。

xService文件服务中的ETag的计算方法不针对文件进行静态计算，而是根据具体操作（HEAD/GET/PUT）采用动态计算，这一特性是为了在文件断点续传中，使得请求方读写文件更为确切。

ETag = STRING(MD5(文件名，文件大小，文件修改时间))

在断点续传中，网络两端的文件大小和文件写入时间存在差异，请求方可以在读写操作过程中，修改文件大小和文件写入时间，生成动态的ETag，并结合If\_None\_Match/If\_Match，以保证文件读写次序和写入时间正确。

# ****3.6.1 xService 获取文件属性****

请求方使用HEAD方法获取xService存储目录中文件的大小、写入时间、ETag基本属性。HEAD方法不会返回任何数据实体。

相关HTTP请求头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| 参见HTTP头定义 |  |  |

相关HTTP响应头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| Content-Length | 文件的字节数 |  |
| Last-Modified | 文件的修改时间 |  |
| ETag | 文件的ETag |  |

请求示例：

*HEAD /LOC/LOG/myservice/20151213012359.log HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS \*\*\*\*: ####*

响应示例：

*HTTP/1.1 200 Ok*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Last-Modified: Fri, 24 Feb 2012 02:58:28 GMT*

*ETag: “abcdefg”*

*Content-Length: 1111*

*Connection: close*

*Server: xService*

# ****3.6.2 xService 获取目录列表****

请求方使用LIST方法获取xService存储目录中文件及子目录的列表，列表信息包含：是否为文件、文件的大小、修改时间、ETag基本属性。LIST方法返回XML数据包。

相关HTTP请求头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| 参见HTTP头定义 |  |  |

相关HTTP响应头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| 参见HTTP头定义 |  |  |

相关xml节点的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 描述 | 选项 |
| name | 文件名 |  |
| write-time | 文件修改时间 |  |
| size | 文件字节数 |  |
| ETag | 文件的ETag |  |

请求示例：

*LIST /LOC/LOG/ HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS \*\*\*\*: ####*

响应示例：

*HTTP/1.1 200 Ok*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Content-Type: application/xml*

*Content-Length: 1111*

*Connection: close*

*Server: xService*

<?xml version="1.0" encoding="UTF-8" ?>

<file-list>

<file name=”demo.xml” size=”100” write-time=” *Fri, 24 Feb 2015 02:58:28 GMT*” ETag=”abcdef”/>

<directory name=”myservice”/>

</file-list>

# ****3.6.3 xService 下载文件****

请求方使用GET方法获取xService存储目录中的文件，响应头中包含：文件的大小、修改时间、ETag基本属性。数据体中返回文件内容。

相关HTTP请求头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| Accept-Encoding | 请求方声明可接受的压缩传输编码类型。  如果声明了此项，服务方对于大于8M的数据体予以压缩后返回。 | ｛gzip｜  deflate｝ |
| If-Match | 请求方声明的文件ETag匹配条件。  如果声明了此项，服务方将计算文件的ETag，与请求方的ETag匹配则服务的接受请求，不匹配则拒绝请求。 | ｛ETag｝ |
| If-None-Match | 请求方声明的文件ETag匹配条件。  如果声明了此项，服务方将计算文件的ETag，与请求方的ETag不匹配则服务的接受请求，匹配则拒绝请求。 | ｛ETag｝ |
| If-Modified-Since | 请求方声明的文件时间戳。  如果声明了此项，服务方将取文件的修改时间，大于此时间则服务的接受请求，否则拒绝请求。 | ｛UTC/GMT｝ |

相关HTTP响应头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| Content-Length | 服务方指明返回数据实体的字节数。  当服务方数据体非压缩返回，此项等同于文件字节数，压缩返回，则等同于文件压缩后的字节数。 |  |
| ETag | 服务方返回的文件ETag | ｛ETag｝ |
| Last-Modified | 服务方返回的文件最后修改时间 | UTC/GMT |
| Content-Encoding | 服务方指明压缩传输编码类型。  如果服务方的数据体采用压缩返回，此项指明了压缩方式。 | ｛gzip｜  deflate｝ |
| Content-Range | 服务方指明返回数据实体的字节范围。  如果服务方的数据体采用压缩返回，请求方可以从此项获取文件的实际大小。 | ｛from｝－｛to｝／｛total｝ |

非压缩传输请求示例：

*GET /LOC/LOG/demo.xml HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS \*\*\*\*: ####*

非压缩传输响应示例：

*HTTP/1.1 200 Ok*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Content-Length: 1111*

*ETag: “abcdefgh”*

*Last-Modified: Fri, 24 Feb 2015 02:58:28 GMT*

*Connection: close*

*Server: xService*

｛非压缩的文件实体｝

压缩传输请求示例：

*GET /LOC/LOG/demo.pdf HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS \*\*\*\*: ####*

*Accept-Encoding: gzip*

压缩传输响应示例：

*HTTP/1.1 200 Ok*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Content-Length: 555*

*ETag: “abcdefgh”*

*Last-Modified: Fri, 24 Feb 2015 02:58:28 GMT*

*Content-Encoding: gzip*

*Content-Range: 0-1110/1111*

*Connection: close*

*Server: xService*

｛压缩的文件实体｝

# ****3.6.4 xService 区段下载文件****

请求方使用GET方法按区段获取xService存储目录中的文件，区段下载同样支持压缩传送。区段下载需要结合If-None-Match和If-Match使用，If-None-Match总是以请求方文件的ETag为基准，If-Match总是以服务方文件的ETag为准。在首次区段下载时请求方以本地文件的ETag填写If-None-Match，如果本地文件不存在可以填入空值，在后续区段下载时If-Match均使用服务方返回的ETag。

* **有关文件压缩下载的请参见3.6.3 xService下载文件**

相关HTTP请求头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| Range | 请求方声明的文件字节区段。  from指明以零为起始的开始位置，to指明截止位置，请求字节数为：(to – from + 1)。 | ｛bytes｝=｛from｝－｛to｝ |
| If-None-Match | 请求方声明的需要比对的ETag。用于首次区段下载。 | {ETag} |
| If-Match | 请求方声明的需要匹配的ETag。用于后续区段下载。 | {ETag} |

相关HTTP响应头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| Content-Range | 服务方指明返回数据实体的字节范围。  from指明以零为起始的开始位置，to指明截止位置，返回字节数为：(to – from + 1)，total指明文件的总字节数。 | ｛from｝－｛to｝／｛total｝ |
| ETag | 服务方返回的文件ETag | ｛ETag｝ |

以下示例服务方200字节的文件分二次下载：

区段下载请求示例：（首次下载）

*GET /LOC/LOG/demo.pdf HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS \*\*\*\*: ####*

*Range: bytes=0-*

*If-None-Match:*

区段下载响应示例：（首次下载）

*HTTP/1.1 200 Ok*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Content-Length: 100*

*ETag: “abcdefgh”*

*Last-Modified: Fri, 24 Feb 2015 02:58:28 GMT*

*Content-Range: 0-99/200*

*Connection: close*

*Server: xService*

｛文件实体｝

区段下载请求示例：（第二次下载）

*GET /LOC/LOG/demo.pdf HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS \*\*\*\*: ####*

*Range: bytes=100-199*

*If-Match: “abcdefgh”*

区段下载响应示例：（第二次下载）

*HTTP/1.1 200 Ok*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Content-Length: 100*

*ETag: “abcdefgh”*

*Last-Modified: Fri, 24 Feb 2015 02:58:28 GMT*

*Content-Range:100-199/200*

*Connection: close*

*Server: xService*

｛文件实体｝

# ****3.6.5 xService 上传文件****

请求方使用PUT方法向xService上传文件。

相关HTTP请求头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| Content-Encoding | 请求方声明压缩传输编码类型。  如果声明了此项，请求方以压缩的方式上传文件。 | ｛gzip｜  deflate｝ |
| Content-Length | 请求方声明上传数据实体的字节数。  当请求方数据体非压缩上传，此项等同于文件字节数，压缩上传，则等同于文件压缩后的字节数。 |  |
| Content-Range | 请求方声明上传数据实体的字节范围。  如果声明了此项，服务方可以籍此获取文件的实际大小。 | ｛from｝－｛to｝／｛total｝ |
| If-Match | 请求方声明的文件ETag匹配条件。  如果声明了此项，服务方将计算文件的ETag，与请求方的ETag匹配则服务的接受请求，不匹配则拒绝请求。 | ｛ETag｝ |
| If-None-Match | 请求方声明的文件ETag匹配条件。  如果声明了此项，服务方将计算文件的ETag，与请求方的ETag不匹配则服务的接受请求，匹配则拒绝请求。 | ｛ETag｝ |
| If-Modified-Since | 请求方声明的文件时间戳。  如果声明了此项，服务方将取文件的修改时间，大于此时间则服务的接受请求，否则拒绝请求。 | ｛UTC/GMT｝ |
| Last-Modified | 请求方声明的文件最后修改时间。  如果声明了此项，服务方将以此修改文件的最后写入时间。 | UTC/GMT |

相关HTTP响应头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| 参见HTTP头定义 |  |  |

非压缩传输请求示例：

*PUT /LOC/LOG/demo.xml HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS \*\*\*\*: ####*

*Content-Length: 1111*

*Last-Modified: Fri, 24 Feb 2015 02:58:28 GMT*

｛文件实体｝

非压缩传输响应示例：

*HTTP/1.1 200 Ok*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Connection: close*

*Server: xService*

压缩传输请求示例：

*PUT /LOC/LOG/demo.pdf HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS \*\*\*\*: ####*

*Content-Encoding: gzip*

*Content-Length: 555*

*Content-Range: 0-1110/1111*

*Last-Modified: Fri, 24 Feb 2015 02:58:28 GMT*

｛文件实体｝

压缩传输响应示例：

*HTTP/1.1 200 Ok*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Connection: close*

*Server: xService*

# ****3.6.6 xService 区段上传文件****

请求方使用PUT方法按区段向xService上传文件，区段上传同样支持压缩传送。区段上传需要结合If-None-Match和If-Match使用，If-None-Match总是以请求方文件的ETag为基准，If-Match总是以服务方文件的ETag为准。在首次区段上传时请求方以本地文件的ETag填写If-None-Match，在后续区段上传时If-Match If-Match均使用服务方返回的ETag。

* **有关文件压缩上传的请参见3.6.5 xService上传文件**

相关HTTP请求头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| Content-Range | 请求方声明上传数据实体的字节范围，服务方按此区段写入文件。 | ｛from｝－｛to｝／｛total｝ |
| If-None-Match | 请求方声明的需要比对的ETag | {ETag} |
| If-Match | 请求方声明的需要匹配的ETag | {ETag} |

相关HTTP响应头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| Last-Modified | 服务方返回的文件最后修改时间。 | UTC/GMT |

以下示例请求方将300字节的文件分三次上传：

区段上传请求示例：（首次上传）

*PUT /LOC/LOG/demo.pdf HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS \*\*\*\*: ####*

*Content-Range: 0-99/300*

*If-None-Match: “abcdefgh”*

*Last-Modified: Fri, 24 Feb 2015 02:58:28 GMT*

｛文件实体｝

区段上传响应示例：（首次上传）

*HTTP/1.1 200 Ok*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Last-Modified: Fri, 24 Feb 2015 02:58:28 GMT*

*Etag: “abcd1”*

*Connection: close*

*Server: xService*

区段上传请求示例：（第二次上传）

*PUT /LOC/LOG/demo.pdf HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS \*\*\*\*: ####*

*Content-Range: 100-199/300*

*If-Match: “abcd1”*

*Last-Modified: Fri, 24 Feb 2015 02:58:28 GMT*

区段上传响应示例：（第二次上传）

*HTTP/1.1 200 Ok*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Last-Modified: Fri, 24 Feb 2015 02:58:28 GMT*

*Etag: “abcd2”*

*Connection: close*

*Server: xService*

区段上传请求示例：（第三次上传）

*PUT /LOC/LOG/demo.pdf HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS \*\*\*\*: ####*

*Content-Range: 200-299/300*

*If-Match: “abcd2”*

*Last-Modified: Fri, 24 Feb 2015 02:58:28 GMT*

区段上传响应示例：（第三次上传）

*HTTP/1.1 200 Ok*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Last-Modified: Fri, 24 Feb 2015 02:58:28 GMT*

*Etag: “abcd3”*

*Connection: close*

*Server: xService*

# ****3.6.7 xService 删除文件****

请求方使用DELETE方法删除服务端的文件。

相关HTTP请求头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| 参见HTTP头定义 |  |  |

相关HTTP响应头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| 参见HTTP头定义 |  |  |

请求示例：

*DELETE /LOC/LOG/demo.xml HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS \*\*\*\*: ####*

响应示例：

*HTTP/1.1 200 Ok*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Connection: close*

*Server: xService*

# ****3.7 xService 云存储代理****

* 云存储代理为非授权直接访问云存储的用户提供访问云资源的动态授权。
* 云存储代理为访问云资源的用户提供高速缓存功能。
* 云存储代理可以建立多个云存储的访问入口，xService负责向用户提供一致的WEB API访问接口。

# ****3.7.1 xService 云存储访问入口****

xService云存储访问入口定义为：｛http:｝//｛地址｝:｛端口号｝/｛oss｝/｛bucket｝/｛对象名｝。

示例URL：<http://192.168.0.1:8000/oss/mybucket/myfile.doc>，其中**oss**是xService的web访问入口，**mybucket**是云存储（如阿里云）中创建的bucket。当通过云存储（如阿里云）外链访问时，被映射为如下URL：<http://mybucket.oss-cn-aliyuncs.com/myfile.doc>。

# ****3.7.2 xService 云存储转移授权****

请求方使用GRANT方法，要求xSercice授予直接访问云存储的权限。

获取访问文件属性权限：http://192.168.0.1:8000/oss/mybucket/myfile.doc?HEAD

获取读取文件权限：<http://192.168.0.1:8000/oss/mybucket/myfile.doc?GET>

获取写入文件权限：<http://192.168.0.1:8000/oss/mybucket/myfile.doc?PUT>

获取删除文件权限：<http://192.168.0.1:8000/oss/mybucket/myfile.doc?DELETE>

授权请求示例：

*GRANT /oss/mybucket/myfile.doc HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS \*\*\*\*: ####*

响应示例：

*HTTP/1.1 200 Ok*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Connection: close*

*Server: xService*

*Authorization: OSS* abcdefg*:* hijklmnopq

请求方在xService响应头中获取OSS accesskey和singature，将其包入云存储访问的URL串中，如阿里云存储访问：<http://mybucket.oss-cn-aliyuncs.com/myfile.doc>?OSSAccessKeyId=abcdefg&Expires=1141889120&Signature=hijklmnopq，其中Expires由请求方自己计算，其值为：自 UTC 时间 1970-01-01开始的秒数。这样，第三方用户就可以通过xService获得授权，从外链直接访问云存储。

# ****3.7.3 xService 云存储缓存服务****

云存储代理缓存中的文件读写的方法与xService文件服务基本相同，不同之处在于访问入口（oss）和bucket的映射。xService负责缓存文件与云存储文件的同步。

* **有关文件读写方法的请参见3.6 xService文件服务**
* **有关云存储访问入口的请参见3.7.1 xService云存储访问入口**

# ****3.7.4 xService 云存储同步服务****

xService虽然负责对缓存中文件与云存储进行异步方式的同步，但请求方也可以通过调用SYNC方法对指定文件进行即时同步。同步遵循master-slave规则，以缓存为master，云存储为slave。

* 如果云存储文件比缓存文件更新，则获取云存储文件覆盖缓存文件。
* 如果云存储文件旧于缓存文件，则将缓存文件覆盖云存储文件。
* 如果云存储中的文件在缓存中不存在，则删除云存储中的文件。

同步请求示例：

*SYNC /oss/mybucket/myfile.doc HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS \*\*\*\*: ####*

同步响应示例：

*HTTP/1.1 200 Ok*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Connection: close*

*Server: xService*

# ****3.8 xService 数据库网关****

xService数据库网关通过提供一组WEB API为用户提供访问数据库资源的接口。使用网关访问数据库的几个特点：

* 用户端无需安装数据库访问客户端工具。
* 用户端无需通过数据库定义的用户访问数据库。数据库的位置对用户是不可见的，xService只为认证通过的特定应用和特定用户提供数据库访问能力。
* 用户通过短时连接访问数据库资源，在网络切换频繁的移动环境中用户无需担心以往采用持久连接的方式会导致连接失效的问题。
* xService可以注册多种／多个数据库访问入口，并为用户保持访问方法的一致。
* xService支持以流的方式处理数据记录，以缓解大数据集存取时对内存使用的压力。

# ****3.8.1 xService 数据库网关访问入口****

数据库网关访问入口的URL形如：｛http:｝//｛地址｝:｛端口号｝/｛xdb｝/｛连接类型｝/｛数据库dsn｝

如：通过odbc访问sqlserver pubs数据库<http://192.168.0.1:8000/xdb/odbc/pubs.dsn>

如：通过oci访问oracle orcl数据库<http://192.168.0.1:8000/xdb/oci/orcl.dsn>

以下创建测试表用于示例：

create table mydb.dogs{

did nvarchar(10) not null,

dname nvarchar(50) not null,

dage date null

dprice numeric(10,2),

primary key (pid)

}—用于演示存取常用数据类型

create table mydb.cats{

cid nvarchar(10) not null,

cdoc nvarchar(max) null,

primary key (cid)

}—用于演示存取长文本数据类型

create table mydb.pigs{

pid nvarchar(10) not null,

pdoc varbinary(max) null,

primary key (pid)

}—用于演示存取二进制数据类型（图像）

create table mydb.docs{

xid narchar(10),

xdoc xml null,

primary key (xid)

}—用于演示存取xml文档

# ****3.8.2 xService 数据库网关获取数据模式****

请求方使用**xdb-schema**获取查询语句的数据模式。

相关HTTP请求头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| xds-action | xdb-schema |  |

相关HTTP响应头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| 参见HTTP头定义 |  |  |

相关xml节点的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 描述 | 选项 |
| type | 数据类型 | ｛string|  short|  long|  numeric|  date|  time|  datetime|  complextType｝ |
| id | 节点标识符 |  |
| name | 节点名称 |  |
| precision | 数据长度 |  |
| scale | 小数位数 |  |

URL查询串的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 描述 | 格式 |
| /?query | Sql查询语句 | ｛sql｝=｛查询语句｝ |

**注意**：以下演示url查询串的字符未进行转义，在实际传输前应先进行url转义。

* **有关URL查询串字符的转义请参见3.2.3URL转义**

请求示例：

*GET /xdb/odbc/mydb.dsn?sql=select \* from dogs HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS \*\*\*\*: ####*

*xds-action: xdb-schema*

响应示例：

*HTTP/1.1 200 Ok*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Content-Type: application/xml*

*Content-Length: 100*

*Connection: close*

*Server: xService*

<?xml version="1.0" encoding="UTF-8" ?>

<schema>

<declare type="complexType" id="row" name="row">

<element type="string" id="did" name="did" precision=20/>

<element type="string" id="dname" name="dname" precision=50/>

<element type="date" id="dage" name="dage"/>

<element type="numeric" id="dprice" name="dprice" precision=10 scale=2/>

</declare>

<element type="complexType" id="rowset" name="rowset">

<element ref="row"/>

</element>

</schema>

# ****3.8.3 xService 数据库网关查询数据集****

请求方使用**xdb-query**获取查询语句的数据集合。

相关HTTP请求头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| xds-action | xdb-query |  |

相关HTTP响应头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| 参见HTTP头定义 |  |  |

URL查询串的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 描述 | 选项 |
| /?query | Sql查询语句 | ｛sql｝=｛查询语句｝ |

**注意**：以下演示url查询串的字符未进行转义，在实际传输前应先进行url转义。

* **有关URL查询串字符的转义请参见3.2.3URL转义**

请求示例：

*GET /xdb/odbc/mydb.dsn?sql=select \* from dogs HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS \*\*\*\*: ####*

*xds-action: xdb-query*

响应示例：

*HTTP/1.1 200 Ok*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Content-Type: application/xml*

*Content-Length: 100*

*Connection: close*

*Server: xService*

<?xml version="1.0" encoding="UTF-8" ?>

<rowset>

<row>

<did>1</did>

<dname>旺财</dname>

<dage>2014-01-01</dage>

<dprice>100.25</dprice>

</row>

<row>

<did>2</did>

<dname>来福</dname>

<dage>2015-01-01</dage>

<dprice>200.50</dprice>

</row>

</rowset>

# ****3.8.4 xService 数据库网关导出数据集****

请求方使用**xdb-export**获取查询语句的数据集合。服务方先返回的第一行为列名集合，之后为多行列值集合，列名和列值由\t分隔，行由\n分隔，数据传输方式采用chunked。

相关HTTP请求头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| xds-action | xdb-export |  |

相关HTTP响应头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| Transfer-Encoding | chunked |  |

URL查询串的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 描述 | 选项 |
| /?query | Sql查询语句 | ｛sql｝=｛查询语句｝ |

**注意**：以下演示url查询串的字符未进行转义，在实际传输前应先进行url转义。

* **有关URL查询串字符的转义请参见3.2.3URL转义**

请求示例：

*GET /xdb/odbc/mydb.dsn?sql=select \* from dogs HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS \*\*\*\*: ####*

*xds-action: xdb-export*

响应示例：

*HTTP/1.1 200 Ok*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Content-Type: text/plain;charset=utf-8;*

*Transfer-Encoding: chunked*

*Connection: close*

*Server: xService*

{chunk head}did\tdname\tdage\tdprice\n{chunk tail}

{chunk head}1\t旺财\t2014-01-01\t100.25\n{chunk tail}

{chunk head}1\t来福\t2015-01-01\t200.50\n{chunk tail}

{chunk head} {chunk tail}

# ****3.8.5 xService 数据库网关导入数据集****

请求方使用**xdb-import**将数据集合导入指定表中。请求方先上传一行列名集合，然后上传多行列值集合，列名和列值由\t分隔，行由\n分隔，数据传输方式采用chunked。

相关HTTP请求头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| xds-action | xdb-import |  |
| Transfer-Encoding | chunked |  |

相关HTTP响应头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| 参见HTTP头定义 |  |  |

URL查询串的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 描述 | 选项 |
| /?query | 数据表名称 | ｛table｝=｛表名｝ |

**注意**：以下演示url查询串的字符未进行转义，在实际传输前应先进行url转义。

* **有关URL查询串字符的转义请参见3.2.3URL转义**

请求示例：

*POST /xdb/odbc/mydb.dsn?table=dogs HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS \*\*\*\*: ####*

*Content-Type: text/plan; charset=utf-8;*

*Transfer-Encoding: chunked*

*xds-action: xdb-import*

{chunk head}did\tdname\tdage\tdprice\n{chunk tail}

{chunk head}1\t旺财\t2014-01-01\t100.25\n{chunk tail}

{chunk head}1\t来福\t2015-01-01\t200.50\n{chunk tail}

{chunk head} {chunk tail}

响应示例：

*HTTP/1.1 200 Ok*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Connection: close*

*Server: xService*

# ****3.8.6 xService 数据库网关更新数据集****

请求方使用**xdb-batch**批量执行sql语句，所有的更新语句将放在一个事务中处理。数据传输方式采用chunked。

相关HTTP请求头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| xds-action | xdb-batch |  |
| Transfer-Encoding | chunked |  |

相关HTTP响应头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| 参见HTTP头定义 |  |  |

请求示例：

*POST /xdb/odbc/mydb.dsn HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS \*\*\*\*: ####*

*Content-Type: text/plan; charset=utf-8;*

*Transfer-Encoding: chunked*

*xds-action: xdb-batch*

{chunk head}delete from dogs\n {chunk tail}

{chunk head}insert into dogs(did,dname,dage,dprice) values(“1”,”旺财”,”2014-01-01”,”100.25”)\n{chunk tail}

{chunk head}insert into dogs(did,dname,dage,dprice) values(“1”,”来福”,”2015-01-01”,”200.50”)\n{chunk tail}

{chunk head}update dogs set dprice=300 where did = ‘1’\n {chunk tail}

{chunk head} {chunk tail}

响应示例：

*HTTP/1.1 200 Ok*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Connection: close*

*Server: xService*

# ****3.8.7 xService 数据库网关执行存储过程****

请求方使用**xdb-proc**调用数据库存储过程。

相关HTTP请求头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| xds-action | xdb-proc |  |

相关HTTP响应头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| 参见HTTP头定义 |  |  |

请求示例：

*POST /xdb/odbc/mydb.dsn HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS \*\*\*\*: ####*

*Content-Type: application/xml*

*Content-length: 100*

*xds-action: xdb-proc*

<?xml version="1.0" encoding="UTF-8" ?>

<func name="myfunc" data-type="integer">

<param param-type="[in]" data-type="string" name="param1">input1</param>

<param param-type="[in,out]" data-type="string" name="param2">input2</param>

<param param-type="[out]" data-type="string" name="param3"/>

</func>

响应示例：

*HTTP/1.1 200 Ok*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Content-Type: application/xml*

*Content-Length: 100*

*Connection: close*

*Server: xService*

<?xml version="1.0" encoding="UTF-8" ?>

<func name="myfunc" return="1">

<param name="param2">output2</param>

<param name="param3">output3</param>

</func>

# ****3.8.8 xService 数据库网关读取BLOB****

请求方使用**xdb-readblob**获取查询语句检索返回的blob数据，数据传输方式采用chunked。

相关HTTP请求头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| xds-action | xdb-readblob |  |

相关HTTP响应头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| Transfer-Encoding | chunked |  |

URL查询串的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 描述 | 选项 |
| /?query | Sql查询语句 | ｛sql｝=｛查询语句｝ |

**注意**：以下演示url查询串的字符未进行转义，在实际传输前应先进行url转义。

* **有关URL查询串字符的转义请参见3.2.3URL转义**

请求示例：

*GET /xdb/odbc/mydb.dsn?sql=select pdoc from pigs where pid = ‘1’ HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS \*\*\*\*: ####*

*xds-action: xdb-readblob*

响应示例：

*HTTP/1.1 200 Ok*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Transfer-Encoding: chunked*

*Connection: close*

*Server: xService*

{chunk head}#########{chunk tail}

{chunk head}############{chunk tail}

{chunk head}################{chunk tail}

{chunk head} {chunk tail}

# ****3.8.9 xService 数据库网关写入BLOB****

请求方使用**xdb-writeblob**将blob数据写入指定字段中，数据传输方式采用chunked。

相关HTTP请求头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| xds-action | xdb-writeblob |  |
| Transfer-Encoding | chunked |  |

相关HTTP响应头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| 参见HTTP头定义 |  |  |

URL查询串的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 描述 | 选项 |
| /?query | Sql执行语句，占位符?表示blob字段值 | ｛sql｝=｛执行语句｝ |

**注意**：以下演示url查询串的字符未进行转义，在实际传输前应先进行url转义。

* **有关URL查询串字符的转义请参见3.2.3URL转义**

请求示例：

*POST /xdb/odbc/mydb.dsn?sql=insert into pigs(pid,pdoc) values(‘1’,?) HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS \*\*\*\*: ####*

*Transfer-Encoding: chunked*

*xds-action: xdb-writeblob*

{chunk head}##########{chunk tail}

{chunk head}##################{chunk tail}

{chunk head}#########################{chunk tail}

{chunk head} {chunk tail}

响应示例：

*HTTP/1.1 200 Ok*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Connection: close*

*Server: xService*

# ****3.8.10 xService 数据库网关读取CLOB****

请求方使用**xdb-readclob**获取查询语句检索返回的clob数据。

相关HTTP请求头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| xds-action | xdb-readclob |  |

相关HTTP响应头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| 参见http定义 |  |  |

URL查询串的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 描述 | 选项 |
| /?query | Sql查询语句 | ｛sql｝=｛查询语句｝ |

**注意**：以下演示url查询串的字符未进行转义，在实际传输前应先进行url转义。

* **有关URL查询串字符的转义请参见3.2.3URL转义**

请求示例：

*GET /xdb/odbc/mydb.dsn?sql=select cdoc from cats where cid = ‘1’ HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS \*\*\*\*: ####*

*Accept: text/plain*

*Accept-Charset: utf-8*

*xds-action: xdb-readclob*

响应示例：

*HTTP/1.1 200 Ok*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Content-Type:text/plain;charset=utf-8;*

*Content-Length: 100*

*Connection: close*

*Server: xService*

{文本数据实体}

# ****3.8.11 xService 数据库网关写入CLOB****

请求方使用**xdb-writeclob**将clob数据写入指定字段中。

相关HTTP请求头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| xds-action | xdb-writeclob |  |

相关HTTP响应头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| 参见HTTP头定义 |  |  |

URL查询串的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 描述 | 选项 |
| /?query | Sql执行语句，占位符?表示clob字段值 | ｛sql｝=｛执行语句｝ |

**注意**：以下演示url查询串的字符未进行转义，在实际传输前应先进行url转义。

* **有关URL查询串字符的转义请参见3.2.3URL转义**

请求示例：

*POST /xdb/odbc/mydb.dsn?sql=insert into cats(cid,cdoc) values(‘1’,?) HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS \*\*\*\*: ####*

*Content-Type:text/plain;charset=utf-8;*

*Content-Length: 100*

*xds-action: xdb-writeclob*

｛文本数据实体｝

响应示例：

*HTTP/1.1 200 Ok*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Connection: close*

*Server: xService*

# ****3.8.12 xService 数据库网关读取XML****

请求方使用**xdb-readxdoc**获取查询语句检索返回的xml数据。

相关HTTP请求头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| xds-action | xdb-readxdoc |  |

相关HTTP响应头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| 参见http定义 |  |  |

URL查询串的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 描述 | 选项 |
| /?query | Sql查询语句 | ｛sql｝=｛查询语句｝ |

**注意**：以下演示url查询串的字符未进行转义，在实际传输前应先进行url转义。

* **有关URL查询串字符的转义请参见3.2.3URL转义**

请求示例：

*GET /xdb/odbc/mydb.dsn?sql=select xdoc from docs where xid = ‘1’ HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS \*\*\*\*: ####*

*Accept: text/xml*

*Accept-Charset: utf-16*

*xds-action: xdb-readxdoc*

响应示例：

*HTTP/1.1 200 Ok*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Content-Type:text/xml;charset=utf-16;*

*Content-Length: 100*

*Connection: close*

*Server: xService*

<?xml version="1.0" encoding="UTF-16" ?>

<document>

<item/>

<item/>

</document>

# ****3.8.13 xService 数据库网关写入XML****

请求方使用**xdb-writexdoc**将xml数据写入指定字段中。

相关HTTP请求头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| xds-action | xdb-writexdoc |  |

相关HTTP响应头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| 参见HTTP头定义 |  |  |

URL查询串的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 描述 | 选项 |
| /?query | Sql执行语句，占位符?表示xml字段值 | ｛sql｝=｛执行语句｝ |

**注意**：以下演示url查询串的字符未进行转义，在实际传输前应先进行url转义。

* **有关URL查询串字符的转义请参见3.2.3URL转义**

请求示例：

*POST /xdb/odbc/mydb.dsn?sql=update docs set xdoc=? where xid=’1’ HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS \*\*\*\*: ####*

*Content-Type:text/xml;charset=utf-16;*

*Content-Length: 100*

*xds-action: xdb-writexdoc*

<?xml version="1.0" encoding="UTF-16" ?>

<document>

<item/>

<item/>

</document>

响应示例：

*HTTP/1.1 200 Ok*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Connection: close*

*Server: xService*

# ****3.9 xService 短信网关****

用户可以通过xService短信网关收发短信。短信网关的入口定义为：｛http:｝//｛地址｝:｛端口号｝/｛sms｝。

# ****3.9.1 xService 短信网关发送短信****

请求方使用**sms-send**发送消息。

相关HTTP请求头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| xds-action | sms-send |  |

相关HTTP响应头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| 参见HTTP头定义 |  |  |

相关xml节点的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 描述 | 选项 |
| message-src | 短信源地址 |  |
| message-dest | 短信目的地址 |  |
| message-text | 短信文本 |  |
| message-id | 短信标识符（用于取回短信回复） |  |

请求示例：

*POST /sms HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS \*\*\*\*: ####*

*Content-Type: application/xml*

*Content-length: 100*

*xds-action: sms-send*

<?xml version="1.0" encoding="UTF-8" ?>

<message>

<message-src>135\*\*\*\*\*\*\*\*</message-src>

<message-dest>130\*\*\*\*\*\*\*\*</message-dest>

<message-text>测试消息</message-text>

</message>

响应示例：

*HTTP/1.1 200 Ok*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Content-Type: application/xml*

*Content-Length: 100*

*Connection: close*

*Server: xService*

<?xml version="1.0" encoding="UTF-8" ?>

<message>

<message-id>########</message-id>

</message>

# ****3.9.2 xService 短信网关收取回复****

请求方使用**sms-recv**取回消息回复。

相关HTTP请求头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| xds-action | sms-recv |  |

相关HTTP响应头的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 场景描述 | 格式选项 |
| 参见HTTP头定义 |  |  |

相关xml节点的含义

|  |  |  |
| --- | --- | --- |
| 名称 | 描述 | 选项 |
| message-src | 短信源地址 |  |
| message-dest | 短信目的地址 |  |
| message-text | 短信文本 |  |
| message-id | 短信标识符（用于取回短信回复） |  |

请求示例：

*POST /sms HTTP/1.1*

*Host: 192.168.0.1:8000*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Authorization: XDS \*\*\*\*: ####*

*Content-Type: application/xml*

*Content-length: 100*

*xds-action: sms-recv*

<?xml version="1.0" encoding="UTF-8" ?>

<message>

<message-src>135\*\*\*\*\*\*\*\*</message-src>

<message-dest>130\*\*\*\*\*\*\*\*</message-dest>

<message-id>#######</message-id>

</message>

响应示例：

*HTTP/1.1 200 Ok*

*Date: Fri, 24 Feb 2015 02:58:28 GMT*

*Content-Type: application/xml*

*Content-Length: 100*

*Connection: close*

*Server: xService*

<?xml version="1.0" encoding="UTF-8" ?>

<message>

<message-text>回复测试</message-text>

</message>

# ****3.10 xService 设备网关****

（暂略）

# ****3.11 xService 支付网关****

（暂略）