

# 学生信息查询协议

## 1. 介绍

学生信息查询协议被设计用来进行学生信息查询、修改、添加、删除的协议，他与具体的信息存储方式无关，也与具体的语言实现无关。

## 2. 结构

### 2.1 客户端/服务器消息传递

学生信息查询协议的工作方式是在可信通道上交换消息。客户端是指建立在可信通道上的，与服务端连接的程序，用于向服务端发送一个或一组请求消息。服务端是指开放服务端口，并通过响应消息对请求消息进行回应的程序。

客户端与服务端仅指相关程序部分在具体通信中的功能。同一个程序可能既是服务端又是客户端。学生信息查询协议依赖 URI 标识目标资源，同时也用来表示资源之间的相互联系。

## 3. 消息格式

每一个学生信息查询协议消息都以一个统一“起始行”作为标记，之后每一行都对应一条头部数据，最后已一个空行作为消息头部的结尾，在头部结束后可以有选择的决定是否在之后添加消息的实体内容。同时，对于每一个消息，消息头部可以有多条也可以没有。

```
消息 = 起始行
      *(消息头 换行回车)
      换行回车
      [ 消息实体 ]
```

解析消息步骤为，将起始行读入其结构中，通过消息头部的域将消息头读入哈希表中，直到空行，最后通过对之前内容的解析决定是否消息实体需要读取。如果有，则对消息实体已字节流的方式解析读取直到读取完消息实体的长度或连接被中断。

### 3.1 起始行

一个学生信息查询协议消息要么是从客户端到服务端的请求，要么是服务端到客户端的响应。从句法上来说，这两种消息的格式的区别仅表现在起始行的不同，前者为请求行，后者为状态行，以及在解析消息的算法上会有区别。

理论上，一个消息是请求消息还是响应消息可以通过起始行为请求行还是状态行来进行判断。但是具体实现中，一般来说，客户端仅用来处理响应消息，而服务端仅用来处理请求消息。

起始行 = 请求行 / 状态行

#### 3.1.1 请求行

请求行以一个方法标记开头，之后是请求目标，以及学生信息查询协议版本，且这三项信息已一个空格符进行分割。

请求行 = 方法 空格 请求目标 空格 协议版本 换行回车

接收者通常对请求行利用分割空格符的方法来获得以上三个独立的信息，因为在每个单独的部分中空格符都是不被允许的。

接收者一旦受到了某个非法的请求行，必须已 400 或 301 错误，同时将请求目标消息正确编码后进行回应。接受者不能试图去自动纠正并处理相应的消息，因为这种错误可能是传输路径上的不确定或确定因素造成的。

学生信息查询协议没有为请求行指定最大字节长度。接收者若受到长于自己处理能力范围的方法应当以 501 错误码进行回应。接受者若受到长于自己处理能力范围的 URI 应当已 414 错误进行回应。

#### 3.1.2 状态行

状态行依次包括学生信息查询协议版本、状态代码、描述状态代码意义的文字段，他们之间以一个空格进行分割。

状态行 = 协议版本 空格 状态代码 空格 描述文字 换行回车

状态代码以三个十进制数字进行表示，用以说明服务端解析并实现原请求消息的结果。

描述文字的存在只是为了提供一种便于人类阅读的方式表现状态代码，客户端可以忽略这些消息。

## 3.2 消息头

每一个消息头包括三部分：域名，冒号，域值。  
消息头 = 域名：域值

### 3.3 消息主体

消息主体用于传送详细的相应信息。一般来说，一个消息主体就是一个文件。学生信息查询协议并不对文件的格式作任何限制与要求，比如对于查询请求，服务端返回的可以是 XML 文件，可以是 json 文件，也可以是一个被序列化的类，但是必须在消息头中被定义。

#### 3.3.1 传输编码

若无特殊指定，消息主体直接用明码进行传输，即不进行加密或压缩。对于特殊的文件，可以由服务端在消息头中指定其加密或压缩方式。

#### 3.3.2 消息主体长度

若有消息主体，那么必须在消息头中定义主题长度，单位为 byte。

## 4. 资源

资源即客户端每次查询期望服务端返回的内容。一个请求消息有且只有一个资源信息，但一个资源信息可能包括多个元信息。比如服务端提供的数据包括 ID 和 Name 这两个元信息，那么一个请求消息可以希望服务端同时返回这两个信息。

## 5. 表示

## 6. 请求方法

请求方法表明客户端请求的目的以及其希望从服务端得到的返回消息或希望服务端进行的任务。

方法	描述
ADD	添加信息
DELETE	删除符合条件的信息
EDIT	修改符合条件的信息
QUERY	查询符合条件的信息

### 6.1 请求方法细节

#### 6.1.1 QUERY

QUERY 方法请求用于查询学生信息，服务端通过对请求目标，也就是资源标识的解析判断自己的响应消息应该包含的原信息的种类与数量。QUERY 是最基本的服务，因此一般白说他与查询是等价的。QUERY 旨在提供一种忽略底层存储细节的抽象。因此，不论具体数据是存储在内存、磁盘、网络，以及其存储媒介为文件、数据库还是具体的数据结构，都应该提供用于按照一个或多个元信息查询的接口，以及一定的按照条件筛选功能。

客户端可以使用条件查询的功能要求服务端回应的数据是经过处理与筛选的，其具体实现需要消息头部的使用。

客户端的查询结果是可以缓存的，是否使用缓存决定于具体的消息头部定义。

#### 6.1.2 EDIT

EDIT 方法请求用于修改已经存在的信息。

#### 6.1.3 DELETE

DELETE 方法请求用于删除已经存在的信息，或者使相应的的记录对外不再可见，且无法修改。在功能上，他相当于文件系统中的 rm、数据库中的 delete 等。他将删除整组源信息，所以请求目标在此不再有意义，但仍需填写且不为空，仅为保持与其他函数的格式对齐。显然单纯的删除的功能有限且是极其危险的，所以一般来说删除操作都会在消息头中设置一条或多条的筛选条件。

处于安全性的考虑，并不是所有客户端都有权限执信删除操作，或者说只有极少数的客户端才被授权以删除的权利。

如果一个删除命令被成功执行，服务端应当返回 202 状态码作为操作成功的标识。

#### 6.1.4 ADD

ADD 方法用于添加一条新的记录。

### 7. 请求头域

客户端发送请求头向服务端提供更多且具体的信息，以实现内容筛选、编码选择、传输限制、权限认证、缓存机制、语言设置、服务端标识等功能。

头部域名	意义
Range	
Accept	
Accept-Charset	
Accept-Encoding	
Accept-Language	
Authorization	
Filter	
If-Modified-Since	

#### 7.1 Accept

Accept 被客户端用来表明自己允许接收的文件格式。这一项常被用来将请求目标限定在一个比较小的范围内，例如请求一张图片，客户端可以要求只接受 jpg 格式的文件，而忽略 png、bmp 等其他格式的文件。当然，客户端可能会允许多种格式的文件，那么这些文件格式需要以符号作为分隔符进行分割。由于文件格式中是不允许出现空格符的，所以允许在符号之后跟随一个空格符。文件格式的表示规定为 mime，而不是单纯的拓展名，同时应支持 mime 表中中对通配符的定义。如果客户端希望获得任意格式的音频文件以及符合上述规范的图像文件，那么 Accept:audio/\*; image/jpeg 将是一种合法的表达。

#### 7.2 Accept-Charset

Accept-Charset 用来表明客户端可以接收的纯文本文件的编码格式。文字编码格式在 RFC6365 中被定义。客户端应当允许服务端使用多种文字编码以支持更多的语言种类，不同的文字编码之间应当以分号相隔。分号之后允许跟随一个没有意义的空格符。如果客户端只允许 UTF-8 和 GBK2312 这两种编码的纯文本文件，那么一种合法的表达为：Accept-Charset:UTF-8; GBK2312

#### 7.3 Accept-Encoding

Accept-Encoding 用来表明客户端允许的文件内容编码格式。最常见的编码格式为 gzip，他有利于减小文件的大小从而使传输所占用的时间简短。若支持任意编码格式，那么符号“\*”便表示不作任何限制，这样的定义在很多地方都是一样的。若要支持多种编码格式，则需以符号进行分割，分号之后允许跟随一个没有意义的空格符。已 gzip 为例，一种合法的表达为 Accept-Encoding:gzip

#### 7.4 Accept-Language

Accept-Language 用来说明客户端可以接受的自然语言。关于自然语言简写的定义在 RFC5646 中，在此不多赘述。客户端可以允许多种自然语言，之间用分号相隔即可，分号之后可以更随一个没有意义的空格符。例如设定语言为美国英语和中文普通话，则表达为：Accept-Language:en-US; zh-cn

#### 7.5 Range

Range 仅允许在查询的被使用。当服务端的信息存储量较大时，该域允许客户端每次只请求检索结果的一部分，从而避免了由于数据量过大导致的大量数据传输的问题。该域的值必须包含两个大于零整数，分别表示信息开始的位置和总共信息组数，中间用分号隔开。例如客户端需要查询满足某一条件的前 10 条记录，那么相应的表达为：Range:1;10

#### 7.6 Filter

Filter 用于服务端对所有学生信息的筛选，他允许客户端对所有符合一定条件的学生信息进行查询、

修改或删除。Filter 域值为一句逻辑表达式，服务器对所有使该逻辑表达式值为真的信息进行操作。

7.7 If-Modified-Since

If-Modified-Since 用于客户端的缓存机制。客户端将缓冲区中信息的获取时间发送给服务端，若服务端发现该信息的最后修改时间早于该时间，则返回 304 状态码而不会将具体信息传送给客户端。学生信息查询协议采用的时间格式与 HTTP 相同，当客户端的时间信息发生错误时，服务端应当忽略该值。

8. 响应状态码

响应状态码用于标识服务端按照客户端请求的执行结果，他会告知客户端执行是成功还是失败，以及具体失败的原因、客户端接下来需要做的事等。

状态码	意义
200	OK
201	创建成功
202	修改成功
203	删除成功
300	无内容
301	未添加
302	未修改
303	未删除
400	错误请求
401	方法不被允许
402	不支持的文件格式
403	非法范围标识
404	需要升级
500	服务器内部错误
501	不支持的学生信息查询协议版本
600	缓存有效

8.1 200

该状态码表明客户端的请求已经被成功执行，用于查询语句。

8.2 201

该状态码表明客户端的请求已经被成功执行，用于添加语句。

8.3 202

该状态码表明客户端的请求已经被成功执行，用于修改语句。

8.4 203

该状态码表明客户端的请求已经被成功执行，用于删除语句。

8.5 300

该状态码表明客户端的查询请求被成功执行，但无返回数据。

8.6 301

该状态码表明客户端的添加请求被成功执行，但未修改数据。

8.7 302

该状态码表明客户端的添加请求被成功执行，但未修改数据。

#### 8.8 303

该状态码表明客户端的删除请求被成功执行，但未修改数据。

#### 8.9 400

该状态码表示客户端的请求消息在解析阶段出现错误。

#### 8.10 401

该状态码表示客户端请求的方法不被服务端所支持。

#### 8.11 402

该状态码表示客户端所请求的文件格式不被服务端所支持。

#### 8.12 403

该状态码表示客户端所请求的范围标识为非法表达。

#### 8.13 404

该状态码表示客户端的协议版本过低，需要升级。

#### 8.14 500

该状态码表示服务端在执行请求的时候意外终止，出现内部错误。

#### 8.15 501

该状态码表示服务端无法解析或识别客户端所提供的版本信息。

#### 8.16 600

当客户端和服务端均支持缓存机制的时候，该状态码表示客户端的缓存任然有效。

### 9. 响应头域

Last-Modified 响应头域使服务端能够将额外的信息传递给客户端，这些信息包括服务端的信息、相关资源的获取等。

头部域名	意义
Date	消息产生时间
Warning	警告信息
Error	错误信息
Last-Modified	最后修改时间
Content-Length	消息主体长度
Media-Type	文件格式

#### 9.1 Date

Date 记录相应消息生成的时间，时间格式与 HTTP 相同，用以告诉客户端该相应消息的生成时间。

#### 9.2 Warning

Warning 表示客户端的请求被正确执行但有可能并没有按照客户端的原始用意被执行。例如企图添加一条原来就存在的信息、企图删除一条不存在的信息、对一条信息进行修改却未改变原值等情况。具体哪些情况下会导致 Warning 的产生由服务端决定。Warning 表示没有固定的语法规定，可以直接使用自然语言或者使用额外的协议与客户端进行约定，但是不允许出现换行回车符，且所有的字符必须是可打印字符。

#### 9.3 Error

Error 表示服务端在解析或执行客户端的请求时发生错误。包括请求语法错误、执行非法操作等。同样，具体的错误产生是由服务端来决定的。具体的 Error 信息没有统一的格式规定，可以直接使用自然语言，

但不允许出现回车换行符。

#### 9.4 Last-Modified

Last-Modified 表面客户端查询信息的最后修改时间戳。若客户端一次查询了多条信息，那么该值为多条信息对应的最后修改时间中最晚的。Last-Modified 采用的时间格式与 HTTP 时间格式相同。

#### 9.5 Content-Length

用于指定消息主体的长度。

#### 9.6 Media-Type

用于指定消息主体所对应文件的长度。