

GRM 云服务器的 Web 数据接口

版本 2.09

广州市巨控电子科技有限公司

2019-06-28

目录

一. 系统简介.....	3
1.客户端的操作原理.....	3
2.客户端登录的推荐作法.....	4
3.历史数据的容量档次.....	4
二. 客户端操作命令说明.....	5
1.[登录].....	5
2.[读变量].....	5
3.[重复上次读].....	6
4.[写变量].....	6
5.[枚举变量信息].....	6
6.[读定位信息].....	7
7.[读电话号码].....	8
8.[写电话号码].....	9
9.[*读历史报警短信].....	10
10.[*删历史报警短信].....	12
11.[*枚举历史数据变量信息].....	12
12.[*读历史数据记录].....	13
13.[*删历史数据记录].....	14
14.[*查询历史报警短信的全局状态].....	15
15.[*查询历史数据记录的全局状态].....	15
16.[读取模块信息].....	16
17.[写入自定义信息].....	17
18.[读取自定义信息].....	17
三. 编程调试中需要注意的一些地方.....	18
1.网页界面上的系统变量名称转换.....	18
2.常见的编程错误.....	18
3.常见的一种错误作法流程.....	19
4.正确的简单作法流程.....	19
5.网页 JS 脚本使用 CORS 跨域功能的最简单做法.....	19
6.推荐的最佳作法流程（较复杂，但性能最好）.....	19
四. 程序返回的错误码和错误描述汇总.....	20

注：标有*的命令属于历史数据相关命令，需要额外开通权限才能使用。

一. 系统简介

GRM 云服务器提供多种不同的接口以访问 GRM 设备中的数据。各种接口的优缺点见下表：

接口	支持的客户端	优点	缺点	是否默认开通
内置网页系统	任何浏览器，包括 PC 浏览器，手机浏览器等等。	无需任何编程，所有平台包括手机均可用。可直接看到变量表格，也可以修改变量。	页面布局是固定的，不能随意排版，也不支持动画效果。	否
Web 数据接口	任何支持 HTTP 协议的设备，可以是 PC，服务器，手机或者平板电脑等。包括网页中的 JS 脚本。	可以跨广域网，跨操作系统来访问数据。客户端可以是任何操作系统，包括手机系统。	客户必须自行编写程序来执行 HTTP POST 动作。得到数据后，客户必须自行编写程序来显示数据。	否
OPC Server	所有的 PC 组态软件，包括西门子 WinCC，组态王，InTouch 等等。也支持客户自编程序（例如 C++,C#,VB 等等）	对组态软件无需任何编程。所有的组态软件均支持 OPC 标准协议。	只能用于 Windows 平台，并且 OPC Server 不能动态的添加设备（受组态软件和 OPC 协议限制）	是

本文档描述的是 Web 数据接口。另外两种接口（内置网页系统和 OPC Server）在本文档中不涉及。

Web 数据接口对于每个 GRM 设备默认是不开通的，需要开通该功能的请与我们联系。

1.客户端的操作原理

客户端需要 2 步骤动作：第一步是在主服务器上登录，得到一个代表该设备的 Session ID（以后简称 SID），以及该设备对应的数据服务器的地址。第二步是在数据服务器上用 SID 来做读写操作。

SID 是一个 64 位的随机数，它是客户端能够访问这个 GRM 设备的唯一凭据。客户端的 IP 地址可能随时变化，只要拥有正确的 SID 就可以跟设备通讯。在不同的设备之间传递 SID 也是可以的。例如，客户自己的服务器到 GRM 服务器来登录，得到 SID，然后把 SID 传给手机 APP 程序，这样手机 APP 程序就可以在不知道 GRM 设备密码的情况下读写数据了。

网络上所有的变量都来自于 GRM 模块，在网络上只能使用 GRM 模块中已配置的变量，网络读写权限也是在 GRM 模块里面配置。如果配置 GRM 模块和编写客户端程序的不是同一个人，请自行协调有关变量名的问题。变量名里面的英文是不区分大小写的。如果您从内置网页系统里面查看变量名，要注意系统变量名的替换（参见第三章第 1 节）。当然，客户端也可以用 E 命令来枚举 GRM 设备中的变量名和变量的各种属性。

因为所有变量都完全按名字访问，建议在编写客户端程序之前预先确定所有需要在网络上操作的变量的名字。如果客户端程序已经写好，而 GRM 模块又被重新配置，改变了变量名，客户端程序就可能无法正确读写了！除了变量名之外，变量的其它属性可以在 GRM 模块里面随便修改，对网络上的读写操作完全没有影响，包括改变这个变量在 PLC 中的地址，改变数据类型，改变 PLC 通讯参数等等。

每个 Session 只要每隔 10 分钟有至少 1 次数据读或写操作，这个 SID 就会一直有效。而如果在连续 10 分钟内没有发生任何数据读写操作，这个 SID 就自动被服务器删除。如果读写操作返回了错误 8，表示服务器认为这个 SID 是无效的，请重新登录一次，不需要管是什么原因。服务器为了防止客户端程序忘了处理错误 8，可能在 1-2 天之内随机的踢掉客户端（曾经有客户端程序忘了处理 SID 失效，结果上线后一个多月才碰上服务器维护重启，导致 SID 无效，结果他的程序就出错了。后来我们增加了这个功能）。

因为服务器上的每个 Session 需要占用一定的系统资源，系统给每个 GRM 设备分配的 Session 数目限制为 100 个。如果您在 10 分钟之内登录超过 100 次，将得到错误 15，您只能等到某个旧的 SID 在 10 分钟后被删掉，然后才能登录上。客户对于不想使用的 SID 没有办法明确的删除，只能等 10 分钟超时自动删掉。

如果您读取 GRM 设备数据的间隔比较长（1 分钟或者更长），建议可以在每次读取数据之前登录一次，这样您的程序内部不需要记录任何状态。但如果数据读取间隔小于 1 分钟，推荐您只登录一次然后保存着登录得到的 SID，下次就不用再登录了。

我们的主服务器是固定的地址 <http://www.yunplc.com:7080>（如果客户自行搭建主服务器，请向服务器提供商索取地址），数据服务器有多个，具体某个 GRM 设备属于哪个数据服务器由登录命令返回。如果有多个客户端连接到同一个 GRM 设备，这些客户端会得到同样的数据服务器。

如果没有任何客户端访问，GRM 设备端不会有数据流量，只有心跳数据包以便维持 GPRS 连接。此时一个月只消耗几兆的流量。只要有任何客户端连接到云服务器，不论是服务器支持的三种接口（内置网页系统，Web 数据接口，OPC Server）中的哪一种，数据服务器都会自动启动到 GRM 设备的读操作。所有客户端的读都被数据服务器缓冲。不论有多少个客户端，到 GRM 设备的读流量只有 1 份。但是写操作是不缓冲的，如果多个客户端从不同的入口都来写同一个 GRM 设备变量，后发出的写会覆盖之前的写的结果，导致只有最后一个写有效。客户需要自行避免其中的冲突问题。

2. 客户端登录的推荐作法

我们云服务器已支持标准的 CORS 跨域功能。客户端如果是网页，推荐使用网页上的 JS 代码通过 CORS 功能直接连到我们云服务器来获取数据，无需客户端自己的网页服务器从我们云服务器读数据再转发！（提示：**CORS 功能对 WinXP 自带的 IE6 不可用！需要升级到 IE8 以上！其它所有非 IE 的浏览器，包括安卓和 iPhone 的浏览器均可正常使用。如果一定要兼容 IE6，只能继续由客户的网站服务器来转发数据。**）

如果客户自己做网站来从我们服务器获取数据，请特别注意：**不要在每次 Web 客户请求页面的时候登录！**这样不仅影响系统性能，而且是错误的！因为每个 GRM 设备同时只能登录 100 个 SID。如果 10 分钟内有若干个 Web 客户累计刷新了 100 次页面，那么这个设备的 SID 就用光了，之后您的网站就无法再登录了！

如果负载量不大，简单的做法是：在 Web 客户登录您的网站的时候来登录，然后保存 SID 供以后的数据访问来使用。也就是您的每个 Web 客户占用一个我们服务器的 SID。因为每个设备只能同时登录 100 个 SID，这样就限制了您的网站对每个设备只能同时连接 100 个 Web 客户。

对于手机程序（安卓或 IOS），通常也使用这种方法。这样就不需要额外的服务器了，每个手机程序都是直接连接到我们数据服务器的。一般来说每个设备同时上线 100 个手机已经足够用了。如果您要让每个设备同时上线更多的手机，仍然得用下面的做法，通过自己的一个服务器来缓存数据。

对于负载量很大的网站，因为一个 GRM 设备上只有一份数据，在每个 Web 客户请求页面的时候都从我们数据服务器重复读取是没有意义的，我们推荐的**最佳做法**是：用客户网站的 Web 服务器内部做一个数据缓存。也就是说，在有 Web 用户登录到客户网站的时候，客户的 Web 服务器到我们服务器登录一次（对每个设备只需要登录这一次！），然后用单独的一个线程或者定时器，**定时读取**该设备的**全部变量**，缓存起来（读全部数据只有第一次需要带所有变量名，以后都可以用[重复上次读]命令）。Web 客户的所有网页请求都从 Web 服务器的缓存里面直接获取数据，无需任何等待。网页如果要写入变量，也是由网页首先提交到 Web 服务器的缓存里面（以保证其他的用户的页面能立刻看到），然后再排队，用 Web 服务器自己的 SID 写到我们数据服务器。（后面 3.4 和 3.6 节有相应的简单做法和最佳做法的流程）

注意，每个 SID 只有一个读命令缓存。如果客户 A 用一个 SID 发出某种方式的读，然后就开始使用 [重复上次读]命令以节约流量，而客户 B 也用这个 SID 发出了另一个不同的读命令，那么客户 A 的“重复上次读”就会读到错误的结果（变成了重复 B 的读命令结果）。因此不要在不同的地方复用同一个 SID。

3. 历史数据的容量档次

历史数据有多个档次，对应的存储容量和最短存储间隔不同。具体见下表：

	小容量	中容量	大容量
最短记录间隔(秒)	180	60	5
历史数据容量(MB)	32	128	512
历史报警容量(条)	2000	5000	10000

使用历史数据功能需要额外收费，详细信息请联系我们的销售人员。

二. 客户端操作命令说明

客户端所有的操作命令都是 HTTP 的 POST 动作。POST 请求里面的必备部分只需要 **Host** 和 **Content-Length**，如果要读取大量数据，强烈推荐在 HTTP 请求中带上 **Accept-Encoding:gzip** 选项，以便启用 HTTP 的 gzip 压缩功能（通常可以把服务器发回的数据量压缩到 50% 以下）。其他的 HTTP 头选项均不被服务器识别，也无意义。

POST 请求里面的 **Content-Type** 实际被服务器强制当作 "text/plain;charset=UTF-8"，也就是无格式的 UTF-8 纯文本。客户端指定其他任何 Content-Type 均无效。因为 HTTP 协议规定 POST 默认的 Content-Type 是 "application/x-www-form-urlencoded"，请一定要记得修改这个！否则服务器会返回 HTTP 错误码 406。

POST 的数据区永远是 UTF8 编码的(如果有中文，请特别注意编码!)。数据区如果有多行，行与行之间的分隔符是回车换行\r\n（也就是十六进制的 0x0D 和 0x0A，或者十进制的 13 和 10）。

1.[登录]

该命令用于客户端到主服务器登录，以得到 SID 和该设备所属的数据服务器地址。

发送目标	主服务器 (http://www.yunplc.com:7080) 只是登陆服务器地址，不用于数据操作
HTTP 动作	POST
URL	/exlog
请求内容	GRM=xxxxxx PASS=xxxxxx
成功返回	OK ADDR=xxxx:xx 这里是后面操作命令要用的数据服务器地址! SID=xxxxxx
出错返回	ERROR 错误码 错误描述

请求内容中的 GRM=xxxxxx 就是 GRM 设备的序号(11 位数字)，PASS=xxxxxx 就是 GRM 设备的密码。行与行之间的分隔符是回车换行\r\n

返回值有 3 行，第一行 OK 或者 ERROR 表示成功还是出错，如果出错会有错误码和描述。

ADDR=xxxx:xx，冒号前面的部分是该设备所属的数据服务器的地址，后面的部分是数据服务器的端口。

2.[读变量]

该命令用于客户端从数据服务器读指定的变量值，同时更新服务器内部的读命令缓存。每次读取的变量个数没有限制，用一个命令把设备里面所有变量全部读回也是可以的。

发送目标	数据服务器，地址由前面的登录命令返回值 ADDR 返回，不能用登陆服务器地址操作!
HTTP 动作	POST
URL	/exdata?SID=xxxxxx&OP=R
请求内容	第一行是变量个数，以后每一行是一个变量名(示例有 3 个变量)。行与行之间的分隔符是回车换行\r\n
请求内容示例	3 变量名 1 变量名 2 变量名 3
成功返回	第一行是状态码 OK，第二行是变量个数，以后是每行一个变量的数值或者#开头的特殊值(针对单个变量的错误码是#ERROR#错误码，没有值是#NAN#)

出错返回	ERROR 错误码 错误描述
成功应答内容示例	OK 3 变量值1 (变量名错误是#ERROR#16, PLC 通讯故障是#ERROR#7, 没有值是#NAN#) #NAN# 变量值3

3.[重复上次读]

该命令的 URL 和返回内容都与[读变量]命令完全相同，唯一的区别是请求内容为空白（也就是 POST 命令的 Content 没有任何内容）。效果相当于使用了服务器读命令缓存里面的读请求内容。这样可以明显节约客户端流量。

4.[写变量]

该命令用于客户端从数据服务器写指定的变量值。

发送目标	数据服务器，地址由前面的登录命令返回
HTTP 动作	POST
URL	/exdata?SID=xxxxxx&OP=W
请求内容	第一行是变量个数，以后每一行依次是第一个变量名，第一个变量值，第二个变量名，第二个变量值.....
请求内容示例	2 变量名1 变量值1 变量名2 变量值2
成功返回	第一行是状态码,OK 或者 ERROR,第二行是变量个数,以后是每行一个变量的成功标志,0 是成功,其他是错误码
出错返回	ERROR 错误码 错误描述
成功应答内容示例	OK 2 (变量个数) 0 (变量名1 的成功标志) 0 (变量名2 的成功标志)

注意事项:

单个写命令可以写入的变量数目没有限制，但是对连接在 GRM 模块串口上的 PLC 或者 IO 设备来说，通常每秒钟只能写入不超过 10 个变量。如果发出太多的写，可能很久才能执行完毕。

服务器对每个 GRM 模块会缓存 512 个未完成的写操作（包括所有来源： OPC 客户端， Web 客户端， 内置网页）。如果超出缓存数量，最早发出的未执行完的写命令会被丢弃。

所有的写命令不会合并，每个写命令都会真实的写到 PLC 或者 IO 设备里面。也就是说如果对同一个变量先写入 0，后写入 1， GRM 模块也会先发出对该变量写 0 的通讯命令，再发出对该变量写 1 的通讯命令。

5.[枚举变量信息]

该命令用于客户端从数据服务器枚举设备中的变量信息

发送目标	数据服务器，地址由前面的登录命令返回
HTTP 动作	POST
URL	/exdata?SID=xxxxxx&OP=E
请求内容	<p>只有一行，是一个字符串，指定返回的变量信息格式参数。可以是下面 6 个选项字母的任意组合：</p> <ul style="list-style-type: none"> ● N 是变量名，返回值为字符串 ● T 是变量类型，返回值为 B/I/F，分别代表 开关量/整数/浮点数 ● R 是变量读写属性，返回值为 R/W，分别代表 只读/可读写 ● P 是网络权限，返回值为 0/1/2，分别代表 低/中/高 ● G 是变量组名，返回值为字符串。如果有两级变量组，中间是.分隔。 ● C 是 Web 变量描述，返回值为字符串 <p>如果只枚举变量名，内容就是 N； 如果枚举变量名，类型，读写属性，网络权限（常用的选项），内容就是 NTRP 如果枚举变量的所有信息，内容就是 NTRPGC</p>
请求内容示例	NTRP
成功返回	<p>第一行是状态码,OK 第二行是变量的数目 以后各行每个变量的信息，每个信息项之间用逗号分隔</p>
出错返回	<p>ERROR 错误码 错误描述</p>
成功应答内容示例	<p>OK 6 \$COM1ERROR,B,R,0 \$SIGNAL,I,R,0 \$NETTRAFFIC,I,R,0 数字只读整数,I,R,0 数字可读写整数,I,W,0 高权限浮点变量,F,W,2</p>

注意事项:

变量组名 **G** 和 Web 变量描述 **C**，这两个选项的内容都可能是空白。如果某一项是空白，它跟后面一项之间的逗号分隔符仍然存在，因此只要查找逗号来拆分各个选项的内容就可以了。

因为 Web 变量描述的内容可能含有逗号，为了防止解析逗号的时候出现混乱，尽量不要指定 **C** 选项。

如果确实需要获取 Web 变量描述，请把 **C** 选项放在最后。这样在处理了前面几个逗号以后，最后剩下的部分都是 Web 变量描述了。或者先用 **NTRPG** 选项请求一次，每行的内容正常按照逗号来拆分；再用 **C** 选项请求一次，直接得到每行只有 Web 变量描述，这样就不需要处理逗号了。

6.[读定位信息]

该命令用于客户端从数据服务器读取设备的定位信息

发送目标	数据服务器，地址由前面的登录命令返回
HTTP 动作	POST
URL	/exdata?SID=xxxxxx&OP=L
请求内容	(无)
成功返回	<p>第一行是状态码,OK 或者 ERROR, 如果成功，第二行是定位数据项的个数（目前是 9） 以后各行就是位置信息</p>

	如果基站位置信息无法查到，定位数据项个数是 1，后面只有基站原始数据 如果当前没有任何基站信息，定位数据项个数是 0，后面没有任何内容
出错返回	ERROR 错误码 错误描述
成功应答内容示例	OK 9 (定位数据项的个数) 2;460;0; 2508;6F19;20;2508;0A83;10; (基站原始数据，没有基站定位则是 0;0;0) 113.4009 (坐标：经度) 23.1310 (坐标：纬度) 广东省 (位置：省/自治区/直辖市) 广州市 (位置：地/市) 天河区 (位置：区/县) 车陂路 300 号 (位置：乡镇/街道) 广东省广州市天河区车陂路 362 号 (完整的位置字符串) 时尚明苑 (临近标志点)

注意事项:

1. 这里得到的经纬度坐标都是百度地图上的坐标，该坐标值与 GPS 真实坐标有差别（国家测绘局规定中国境内的地图不得使用真实 GPS 坐标）。我们推荐客户使用百度地图来显示位置。如果您要使用其他地图软件，请自行转换坐标。
2. 某些偏远地区可能没有区/县，乡镇/街道和临近标志点，此时这些没有的内容都是空白，但尾部换行符保留。
3. 基站原始数据的格式为（每个数据项之间用分号来分隔）：
基站数目;移动国家码 MCC;移动运营商码 MNC;后面每个基站有 3 个数据，依次是 LAC;CID;信号强度 RSSI;
 - LAC 和 CID 是以十六进制表示的，都是 4 个字符，也就是 16 位数字。
 - 其他参数都是十进制的数字。
 - 基站数目最多可能有 7 个。
 - RSSI 的范围是 0 到 63。
 - 这些原始数据主要是供客户自己执行基站定位算法所用，如果客户直接使用我们的基站定位结果，就不需要解析这些原始数据。
4. 基站定位的精度大约在 100 米到 1 公里之间。在城市里，基站比较密集的时候可以到 100 米级别的精度，如果在偏远地方，基站非常稀少，精度可能降低到 1 公里级别。
5. 基站位置无法查到的原因，可能是运营商新建的基站，或者该位置太偏远，我们的基站信息提供商还没有收录。此时如果您自己有更好的基站信息提供商，也可以自行解析基站数据来定位。
6. 如果设备不支持基站定位功能（例如设备配置中没有打开地图定位功能，或者是旧版本的固件无此功能），会返回错误码 5（设备不支持此功能）。对于旧版本固件不支持基站定位的，可以联系我们客服，获取新版本固件来升级。
7. 如果启用了"自定义地图位置"功能，不论模块是否有基站定位都可以读到定位信息。此时没有基站原始数据，在基站原始数据的这一行是 0;0;0

7.[读电话号码]

该命令用于客户端从数据服务器读取设备内部的电话号码本

发送目标	数据服务器，地址由前面的登录命令返回
HTTP 动作	POST
URL	/exdata?SID=xxxxxx&OP=P
请求内容	电话号码本的最近更新时间，单位秒，可取 1-3600 秒。

	如果服务器还没读过设备的电话号码本，或者上次读取是在最近更新时间之前，服务器就重新开始读电话号码本。
请求内容示例	60
成功返回	<p>第一行是状态码,OK 或者 ERROR,或者 BUSY BUSY 表示系统正在读取电话号码本的过程中，请过一会重试。 OK 表示读取电话号码成功， 如果成功，第二行是电话号码本的最近刷新时间，14 位的 yyyyMMddhhmmss 第三行电话号码本的条数 以后每一行是电话号码本的一项，最多 120 项 每一项由逗号分开的如下内容组成： 序号,管理员标志,用户名,电话号码,密码</p> <p>序号: 从 1 开始，每一项递增（序号 0 有特殊用途） 管理员标志: 管理员是 1,非管理员是 0 用户名: 字符串,最多 15 字符 电话号码: 字符串，最多 15 字符，没有电话号码的是空白，禁用的用户是* 密码: 字符串，最多 15 字符,没有密码就是空白</p>
系统忙返回	BUSY
出错返回	ERROR 错误码 错误描述
成功应答内容示例	OK 20150920093010 3 0,0,SIM,10086, 1,1,管理员,, 2,0,用户 1,,

注意事项:

因为从模块读取电话号码本可能很慢，所以服务器平常并不会主动读取电话号码本，除非有客户需要。在客户发出该命令后，如果服务器上没有电话号码本，或者电话号码本很久没有更新过，服务器就返回 BUSY，并开始从设备读取电话号码本。见到 BUSY 返回值请隔几秒钟重试，一直重试到成功读到为止。

电话号码本的第 0 项永远是用户名 SIM，它的电话号码就是 GRM 工程选项中的“免费短信号码”，用于探测该模块能不能发短信。请不要轻易修改该用户！（对国内运营商我们会自动识别，并不依赖这个选项，只有国外用户需要这个）。工程中实际配置的用户从序号 1 开始。

8.[写电话号码]

该命令用于客户端通过数据服务器来写入设备内部的电话号码本

发送目标	数据服务器，地址由前面的登录命令返回
HTTP 动作	POST
URL	/exdata?SID=xxxxxx&OP=PW
请求内容	<p>第一行是需要修改的电话号码项的数量，以后每 3 行是一条电话号码项 单条命令最多可写入 13 条电话号码项，也就是整个请求最多有 40 行（1+13*3） 每条电话号码项的三行依次是： 序号 电话号码 密码</p>

请求内容 示例	1 3 13412345678 pass
成功返回	第一行是状态码,OK 或者 ERROR,或者 BUSY BUSY 表示系统忙, 请过一会重新执行该命令。 OK 表示修改电话号码项成功。
系统忙返回	BUSY
出错返回	ERROR 错误码 错误描述
成功应答 内容示例	OK

注意事项:

修改电话号码本的操作有可能比较慢。一次修改命令发出后, 电话号码本会变成 BUSY 状态, 以后对电话号码本的操作都会返回 BUSY, 直到修改完成。

因为每次必须同时写入电话号码和密码, 如果您只想修改其中一项, 另外一项需要写入之前的旧值(必须先读一次电话号码本, 然后才能写入)。如果不指定电话号码, 电话号码这一行就是空白; 如果没有密码, 密码这一行就是空白 (空白的行就是只有末尾的换行符); 如果要禁用该用户, 请把电话号码写成*

9.[*读历史报警短信]

该命令用于客户端从数据服务器读取设备的历史报警短信记录

发送目标	数据服务器, 地址由前面的登录命令返回
HTTP 动作	POST
URL	/exdata?SID=xxxxxx&OP=A
请求内容	3 行或 4 行, 第一行是起始时间(年月日必须有, 时分秒可选), 加上可选的时区标志。时区标志与时间用逗号分隔。无时区标志隐含为中国(东八区 GMT+8:00), 时区标志也可以用距离中国时区的偏移量(分钟)来表示。 第二行是结束时间, 格式与起始时间相同, 但不可带时区。时区始终是起始时间中的时区 第三行是结果排序方式, 可选+或-。时间递增是+, 时间递减是- 第四行是最多返回的记录数, 可不写, 默认为 500, 最大值 3000。如果实际数据超过这个数目会被裁剪, 只返回指定的数目, 并且返回的 ClipFlag 标志是 Y。
请求内容 示例	20150618 (只有年月日没有时分秒就是查整天) 20150618 - (结果按递减排列, 最新的在最前面) 2 (最多返回 2 条结果)
含有 UTC 时 区的请求内 容示例	20150618,UTC+1:00 (示例在法国, 东 1 区) 20150618 - 2
含有时区偏 移量的请求 内容示例	20150618,-420 (示例在法国, 东 1 区, 比中国时间早 7 小时, 就是-420 分钟) 20150618 - 2
成功返回	第一行是状态码,OK 或者 ERROR,

如果成功，

第二行是返回结果的属性信息，逗号分隔，格式为：

`OnlineFlag,TimeZone,RecCount,ClipFlag,DbOldTime,DbNewTime,MaxQuery`

OnlineFlag 是模块是否在线的标志，在线是 Y，不在线是 N

TimeZone 是模块的时区距离中国时区的偏移量(分钟)

RecCount 是返回的短信的数量

ClipFlag 表示返回结果是否超出最多返回数目而被裁剪，被裁剪了是 Y，否则是 N

DbOldTime 表示数据库中最老的短信的时间

DbNewTime 表示数据库中最新的短信的时间

MaxQuery 表示每次最多可以读取的记录数量(目前固定为 3000)

后面开始是各个短信，每条短信占 2 行。

前一行是短信的属性，后一行是短信内容。

- 短信属性的各个项目是逗号分隔的，格式为：

`RecId,Tm,Type,AlarmID,Flag,RefTm`

RecId 是结果序号，从 1 开始，到上面的 **RecCount** 为止。例如查询得到 100 条结果，序号就是 1 到 100

Tm 是本条短信产生的时间，格式是 17 位的 `yyyyMMddhhmmsszzz`。对报警短信这就是报警的时间。对恢复/应答，这是恢复或者应答的时间。

Type 是短信类型，单个字母：A 是报警，R 是报警恢复，K 是报警应答

AlarmID 是报警的序号，就是 GRM 开发软件里面配置的报警的序号，从 1 开始。(目前报警序号仅供显示)

Flag 是短信标志，只对报警短信有，可能是 0-2 字母，R 表示该报警已恢复，K 表示该报警已应答，没有标志位的这里就是空白，但逗号分隔符仍然有。(目前版本未启用此功能，这里全部是空白)

RefTm 是对应报警短信的报警时间，只有报警恢复和报警应答短信有，报警短信这里是空白。(目前版本未启用此功能，这里全部是空白)

- 如果短信内容中有回车换行，会被替换为 `
`，免得干扰内容的解析。接收者按照回车换行拆分出各个短信后，可以把短信内容中的 `
` 替换回 `\r\n`。如果只作为网页显示，并不需要替换，因为 `
` 在网页上就是换行符。

出错返回	<code>ERROR</code> 错误码 错误描述
成功应答 内容示例	<code>OK</code> <code>Y,0,2,Y,20160202154300,20160202162301,3000</code> (各种属性信息) <code>1,20150618141300,A,1,,</code> (短信 1 属性) <code>测试报警短信 1
这是第二行</code> (短信 1 内容) <code>2,20150618141200,A,1,,</code> (短信 2 属性) <code>测试报警短信 2</code> (短信 2 内容)

注意事项:

1. 请求中的时间格式可以是 8 位的 `yyyyMMdd`，或者 14 位的 `yyyyMMddhhmmss`。应答内容中的时间是 17 位格式 `yyyyMMddhhmmsszzz`。最后 3 位代码 `zzz` 是时间的毫秒部分(实际上 GRM 模块内部的报警时间只精确到秒，这里显示的毫秒主要是为了区别一秒钟之内产生的多个报警。也就是说，一秒钟之内，第一个报警的毫秒是 000，第二个报警的毫秒是 001，以此类推)。请求中可以带时区，应答的时间就是请求的时区的时间。无时区标志隐含为中国(东八区 GMT+8:00)。同样的数据如果用不同的请求时区来读取，得到的应答里面的时间是不同的。如果模块不在中国而又想得到与模块本地时间相同的记录，请按照返回结果属性中的 **TimeZone** 来发送请求，或者用默认的中国时区读取然后再用时区偏移量来做转换。

yyyy 是 4 位的年，MM 是 2 位的月(01-12)，dd 是 2 位的日(01-31)，

hh 是 2 位的小时(00-23)，mm 是 2 位的分(00-59)，ss 是 2 位的秒(00-59)

- 对于 8 位格式的时间，结束时间实际为该天的末尾(后面隐含 235959)。也就是说，起始时间和结束时间可以是同一天，结果是查询该天的数据。
- 如果查询的时间段内的数据数超过最多返回的记录数，只会返回指定数量的数据：如果时间递增，数据从起始时间开始，结束时间忽略；如果时间递减，数据从结束时间开始，起始时间忽略。
- 推荐不要一次返回太多的数据，否则可能很慢。每一条短信的数据通常在 200 字节左右(极限可能有 700 多字节，因为短信内容极限是 240 个中文，而每个中文的 UTF8 是 3 个字节)。500 条短信的数据就是 100KB 左右。
- 每个用户的报警短信存储容量取决于开通的历史数据档次，参见[\[1.3 历史数据的容量档次\]](#)。超出后会自动绕回覆盖最老的数据。最老/最新短信时间可用于判断当前查询的数据有没有到达数据库的头或尾。
- 这个命令即使模块不在线也可使用。如果想知道模块是否在线，可以通过返回的 **OnlineFlag** 得到。
- 如果您读取报警数据是为了存入自己的数据库，可用返回的时间来做每条记录的唯一 **Key**。这个时间对每一条记录都是唯一的。

10.[*删历史报警短信]

该命令用于客户端从数据服务器删除设备的历史报警短信记录

发送目标	数据服务器，地址由前面的登录命令返回
HTTP 动作	POST
URL	/exdata?SID=xxxxxx&OP=AD
请求内容	1 行，是删除后保留的最早时间(年月日必须有，时分秒可选)，早于此时间的报警数据会被删除。如果使用 8 位时间（只有年月日），则时分秒隐含为全零。 *表示删所有。 如果模块不在中国，时间后面也可以加上可选的时区标志。参见 [读历史报警短信] 命令的请求内容
请求内容示例	20150619
成功返回	第一行是状态码,OK 或者 ERROR, 如果成功, 第二行是删掉的数据条数 第三行是剩余的数据条数。如果剩余 0 条，则表示所有的数据都删空了。
出错返回	ERROR 错误码 错误描述
成功应答内容示例	OK 2 (删掉的数据条数) 0 (剩余数据条数，剩 0 条表示全部删空了)

注意事项:

- 为了保证历史报警的连续性，删除命令只能把某个时间之前的数据全删掉，而不能在数据中间删除某一段。
- 删报警数据的功能主要是为了在系统调试完成后删掉调试数据。平常您完全不需要自己删报警数据来释放空间，我们内部会自动滚动存储。写满以后会自动从最早的时间开始覆盖。您只要保证在数据被覆盖之前读走或者导出需要保存的旧数据即可。

11.[*枚举历史数据变量信息]

该命令用于客户端从数据服务器枚举设备的历史数据变量信息

发送目标	数据服务器，地址由前面的登录命令返回
HTTP 动作	POST

URL	/exdata?SID=xxxxxx&OP=HE
请求内容	只有一行，是一个字符串，指定返回的变量信息格式参数。可以是下面几个选项字母的任意组合： <ul style="list-style-type: none"> ● N 是变量名，返回值为字符串 ● T 是变量类型，返回值为 B/I/F，分别代表 开关量/整数/浮点数 一般情况下只需要枚举变量名，请求内容就是 N ，其它选项字母通常无需使用。
请求内容示例	NTLF
成功返回	第一行是状态码,OK 第二行是变量的数目 以后各行每个变量的信息，每个信息项之间用逗号分隔
出错返回	ERROR 错误码 错误描述
成功应答内容示例	OK 3 开关变量,B, 1,3 整数变量,I,16,16 浮点变量,F,32,32

12.[*读历史数据记录]

该命令用于客户端从数据服务器读取设备的历史数据记录

发送目标	数据服务器，地址由前面的登录命令返回
HTTP 动作	POST
URL	/exdata?SID=xxxxxx&OP=H
请求内容	4 行或 5 行， 第一行是起始时间(年月日必须有，时分秒可选)。如果模块不在中国，时间后面也可以加上可选的时区标志。参见[读历史报警短信]命令的请求内容。 第二行是结束时间，格式与起始时间相同，但不可带时区。时区始终是起始时间中的时区 第三行是结果排序方式，可选+或-。时间递增是+，时间递减是- 第四行是读取的变量名列表，逗号分隔，读所有变量可以写* 第五行是最多返回的记录数，可不写，默认为 500，最大值至少为 3000(可能更大，具体值由返回的 MaxQuery 可以得到)。如果实际数据超过这个数目会被裁剪，只返回指定的数目，并且返回的 ClipFlag 标志是 Y。
请求内容示例	20160202160000 (查询时间从 2016 年 2 月 2 日的 16:00:00 查到 16:26:23) 20160202162623 - (结果按时间递减排列，也就是从结束时间开始) 变量 1,变量 2,变量 3 (查询了 3 个变量) 2 (最多返回 2 条结果)
成功返回	第一行是状态码,OK 或者 ERROR, 如果成功， 第二行是返回结果的属性信息，逗号分隔，格式为： OnlineFlag,TimeZone,RecCount, ClipFlag,DbOldTime,DbNewTime,MaxQuery OnlineFlag 是模块是否在线的标志，在线是 Y，不在线是 N TimeZone 是模块的时区距离中国时区的偏移量(分钟) RecCount 是返回的数据的数量 ClipFlag 表示返回结果是否超出最多返回数目而被裁剪，被裁剪了是 Y，否则是 N

	<p>DbOldTime 表示数据库中最老的数据时间</p> <p>DbNewTime 表示数据库中最新的数据时间</p> <p>MaxQuery 表示每次最多可以读取的记录数量</p> <p>后面开始是各条数据记录，每条占一行。每行用逗号分隔为多个字段。最前面是结果序号，然后是时间，后面就是按请求的变量顺序依次排列各个变量的值。</p> <ul style="list-style-type: none"> ● 结果序号，从 1 开始，到上面的 RecCount 为止。例如查询得到 100 条结果，序号就是 1 到 100 ● 时间，格式是 14 位的 yyyyMMddhhmmss。 ● 数据项如果这个变量的采样点没有数据就是#NAN#（例如 PLC 通讯故障），有数据就是数据。
出错返回	<p>ERROR</p> <p>错误码</p> <p>错误描述</p>
成功应答内容示例	<p>OK</p> <p>Y,0,2,Y,20160202123036,20160202162815,10500</p> <p>1,20160202162623,0,23,10000095</p> <p>2,20160202162621,0,21,10000093</p>

注意事项:

1. 请求中的时间格式可以是 8 位的 yyyyMMdd ，或者 14 位的 yyyyMMddhhmmss。应答内容中的时间都是 14 位格式。请求中可以带时区，应答的时间就是请求的时区的时间。无时区标志则隐含为中国(东八区 GMT+8:00)。同样的数据如果用不同的请求时区来读取，得到的应答里面的时间是不同的。如果模块不在中国而又想得到与模块本地时间相同的记录，请按照返回结果属性中的 **TimeZone** 来发送请求。

yyyy 是 4 位的年，MM 是 2 位的月(01-12)，dd 是 2 位的日(01-31)，

hh 是 2 位的小时(00-23)，mm 是 2 位的分(00-59)，ss 是 2 位的秒(00-59)

2. 对于 8 位格式的时间，结束时间实际为该天的末尾(后面隐含 235959)。也就是说，起始时间和结束时间可以是同一天，结果是查询该天的数据。

3. 如果查询的时间段内的数据数超过最多返回的记录数，只会返回指定数量的数据：如果时间递增，数据从起始时间开始，结束时间忽略；如果时间递减，数据从结束时间开始，起始时间忽略。

5. 每个用户的历史数据存储容量取决于开通的历史数据档次，参见[\[1.3 历史数据的容量档次\]](#)。超出后会自动绕回覆盖最老的数据。最老/最新数据时间可用于判断当前查询的数据有没有到达数据库的头或尾。

6. 这个命令即使模块不在线也可使用。如果想知道模块是否在线，可以通过返回的 **OnlineFlag** 得到。

7. 如果您读取历史数据是为了存入自己的数据库，可用返回的时间来做每条记录的唯一 **Key**。这个时间对每一条记录都是唯一的。

8. 如果只想查询某个特定时间点的数据，要注意模块的记录时间可能不是精确的在某个时间点。应当在指定时间点的前后各增加记录间隔的一半。比如 1 分钟记录一次，而想查询 10:00:00 的数据，那么实际应当查询 09:59:30 到 10:00:30 之间的 1 条数据。这样查到的就是离 10:00:00 最近的那个时间点的数据了。

13.[*删历史数据记录]

该命令用于客户端从数据服务器删除设备的历史数据记录

发送目标	数据服务器，地址由前面的登录命令返回
HTTP 动作	POST
URL	/exdata?SID=xxxxxx&OP=HD
请求内容	<p>1 行，是删除后保留的最早时间(年月日必须有，时分秒可选)，早于此时间的历史数据会被删除。如果使用 8 位时间（只有年月日），则时分秒隐含为零。</p> <p>*表示删所有。</p> <p>如果模块不在中国，时间后面也可以加上可选的时区标志。参见[读历史报警短信]命令的</p>

	请求内容
请求内容示例	20150619
成功返回	第一行是状态码,OK 或者 ERROR, 如果成功, 第二行是删掉的数据条数 第三行是剩余的数据条数。如果剩余 0 条,则表示所有的数据都删空了。
出错返回	ERROR 错误码 错误描述
成功应答内容示例	OK 2 (删掉的数据条数) 0 (剩余数据条数,剩 0 条表示全部删空了)

注意事项:

1. 为了保证历史报警的连续性,删除命令只能把某个时间之前的数据全删掉,而不能在数据中间删除某一段。
2. 删报警数据的功能主要是为了在系统调试完成后删掉调试数据。平常您完全不需要自己删报警数据来释放空间,我们内部会自动滚动存储。您只要保证在数据被覆盖之前读走或者导出需要保存的旧数据即可。

14.[*查询历史报警短信的全局状态]

该命令用于客户端从数据服务器查询历史报警短信数据库的全局状态

发送目标	数据服务器,地址由前面的登录命令返回
HTTP 动作	POST
URL	/exdata?SID=xxxxxx&OP=AG
请求内容	(无)
成功返回	第一行是状态码,OK 或者 ERROR, 如果成功, 第二行是数据库的各种全局状态,逗号分隔,格式为 AllRecCount,RecCountLimit,TimeZone,DbOldTime,DbNewTime,MaxQuery AllRecCount 是数据库中当前的记录条数 RecCountLimit 是数据库的容量限制。如果记录条数超过容量限制,系统会自动删除时间最早的若干条记录 TimeZone 是模块的时区距离中国时区的偏移量(分钟) DbOldTime 表示数据库中最老的短信的时间 DbNewTime 表示数据库中最新的短信的时间 MaxQuery 表示每次最多可以读取的记录数量
出错返回	ERROR 错误码 错误描述
成功应答内容示例	OK 80,10000,60,20160202154300,20160202162301,3000

注意事项:

1. 如果数据库很大(尤其对历史数据记录的数据库),该命令可能比较慢,不要频繁的执行该命令!

15.[*查询历史数据记录的全局状态]

该命令用于客户端从数据服务器查询历史数据记录的全局状态。

发送目标	数据服务器，地址由前面的登录命令返回
HTTP 动作	POST
URL	/exdata?SID=xxxxxx&OP=HG
请求内容	(无)
成功返回	<p>第一行是状态码,OK 或者 ERROR, 如果成功, 第二行是数据库的各种全局状态,逗号分隔,格式为</p> <p>AllRecCount,RecCountLimit,TimeZone,DbOldTime,DbNewTime,MaxQuery</p> <p>AllRecCount 是数据库中当前的记录条数 RecCountLimit 是数据库的容量限制。如果记录条数超过容量限制,系统会自动删除时间最早的若干条记录 TimeZone 是模块的时区距离中国时区的偏移量(分钟) DbOldTime 表示数据库中最老的数据时间 DbNewTime 表示数据库中最新的数据时间 MaxQuery 表示每次最多可以读取的记录数量</p>
出错返回	<p>ERROR 错误码 错误描述</p>
成功应答内容示例	<p>OK 5316,492544,60,20160202123036,20160202163123,10500</p>

注意事项:

1. 如果数据库很大 (尤其对历史数据记录的数据库), 该命令可能比较慢, 不要频繁的执行该命令!

16.[读取模块信息]

该命令用于客户端从数据服务器查询模块的各种信息。不论模块是否在线都可以用。

发送目标	数据服务器，地址由前面的登录命令返回
HTTP 动作	POST
URL	/exdata?SID=xxxxxx&OP=I
请求内容	(无)
成功返回	<p>第一行是状态码,OK 或者 ERROR, 如果成功, 第二行是模块的名称 (这是 GRMDEV 软件中设置的名称), 如果模块未登录, 就是空白 第三行是模块的 Web 设备描述信息 (GRM 内置网页系统里面配置) 第四行是模块的 Logo 图片的文件 URL (GRM 内置网页系统里面配置) 第五行是模块当前登录的 Web 客户端的数目 第六行是模块的状态标志, 有如下几个值: 未登录:0, 正常工作:2, 数据词典初始化:3, 下载状态:5, 错误:6, 透传:12 第七行是模块登录时间 第八行是模块最近活动时间 这两个时间都是 17 位数字 yyyyMMddhhmmsszzz。最后 3 位代码 zzz 是时间的毫秒部分 第九行是模块登录来源的 IP 地址, 注意这是服务器看到的模块公网 IP 地址, 一般来说是模块所在局域网的出口路由器的地址, 并不是模块自身设置的那个 IP 地址! (服务器的未来版本可能增加其它更多的返回信息, 因此如果后面还有内容请忽略)</p>
出错返回	<p>ERROR 错误码 错误描述</p>

成功应答 内容示例	OK	
	冷库远程监控系统 1	
	天津商业大学 XX 楼 XX 冷库	
	/devimg/200010000055999.gif	
	4	(有 4 个 Web 客户端)
	2	(模块状态: 正常工作)
	20160101080000000	(模块登录时间 2016 年 1 月 1 日 8:00:00)
20160101091234000	(最近活动时间 2016 年 1 月 1 日 9:12:34)	
119.131.142.120:9416	(模块登录来源 IP 地址)	

17.[写入自定义信息]

该命令用于客户端保存一个自定义的字符串到服务器。

发送目标	数据服务器, 地址由前面的登录命令返回
HTTP 动作	POST
URL	/exdata?SID=xxxxxx&OP=JXW
请求内容	需要保存的自定义信息, 是字符串。可以包含回车换行。服务器不解析这些内容, 只是原样保存。通常建议该信息是个 JSON 字符串。
成功返回	第一行是状态码,OK
出错返回	ERROR 错误码 错误描述

18.[读取自定义信息]

写入自定义信息对应, 读回之前写入的字符串。

发送目标	数据服务器, 地址由前面的登录命令返回
HTTP 动作	POST
URL	/exdata?SID=xxxxxx&OP=JXR
请求内容	无
成功返回	之前 JXW 命令写入的字符串, 可以是多行。服务器不解析这些内容。

注意事项:

读/写自定义信息这两个命令, 主要用于客户端是纯 JS 没有服务器, 无法保存配置信息的场合。

这个时候可以用该命令在我们的云服务器上保存一些与模块相关的配置信息。这两个命令不论模块是否在线都可以使用。

三. 编程调试中需要注意的一些地方

1. 网页界面上的系统变量名称转换

为了方便调试,很多网页开发者使用我们内置的网页系统来查看变量名和对应的值,但是我们网页上显示的系统变量的变量名是经过转换的。下面是全部系统变量的转换对应表(因为变量名不区分大小写,所以任何大小写形式的系统变量名都可以识别):

系统变量名	网页显示名
\$Com1Error	串口 1 错误标志
\$Com2Error	串口 2 错误标志
\$Com3Error	串口 3 错误标志
\$NetComError	网口设备错误标志
\$SIGNAL	信号强度
\$NetTraffic	网络流量
\$NETOFF	禁用网络标志
\$ALARMFLAG	全局报警标志
\$ALARMOFF	全局报警禁用
\$ErrorCode	系统错误代码
\$NetState	网络连接状态
\$Year	日期: 年
\$Month	日期: 月
\$Day	日期: 日
\$Hour	时间: 时
\$Minute	时间: 分
\$Second	时间: 秒
\$WeekDay	日期: 星期
\$SIMMATCH	SIM 卡相同标志
\$SIMERROR	SIM 卡错误代码
\$POWERIN	系统电源电压
\$POWERIN2	辅助电源电压

2. 常见的编程错误

HTTP 错误 403	表示禁止访问,常见的原因是对我们的网站错误的发出了 GET 请求。注意我们的所有请求都是 POST,需要编程来发起请求,而不能在浏览器的地址栏里面直接输入地址。
HTTP 错误 406	无法接受的类型。 Post 请求的 Content-Type 不能是 application/x-www-form-urlencoded , 请改为 text/plain; charset=UTF-8
HTTP 错误 417	通常只有.net 程序会遇到,请在程序中增加下面一句 <code>System.Net.ServicePointManager.Expect100Continue = false;</code> 或者对单个 <code>HttpWebRequest</code> 对象设置: <code>WebRequest.ServicePoint.Expect100Continue = false;</code>

中文遇到乱码	检查 UTF8 编码。注意跟我们服务器通讯的任何中文字符都是 UTF8 编码的，不论是发到服务器的数据还是从服务器收回的数据。
--------	---

3.常见的一种错误作法流程

- 1) 客户服务器收到用户的网页请求
- 2) 客户服务器到 GRM 服务器登录，获取一个 SID
- 3) 客户服务器用这个 SID 到 GRM 数据服务器读取本次网页需要的数据
- 4) 客户服务器生成网页，送给网页用户
- 5) 这个 SID 扔掉，再也不使用。遇到下一个网页请求，从步骤 1) 重新开始

4.正确的简单作法流程

- 1) Web 用户在网站登录的时候，网站服务器到 GRM 服务器登录，得到 SID 与该用户关联。
- 2) 该用户的每次网页请求，都用这个 SID 到 GRM 服务器做一次数据读，得到数据来生成网页。
- 3) 如果网页有写操作，也用这个 SID 发出写命令。
- 4) 如果某一次读或写返回错误码，发现这个 SID 已经过期，就重新登录一次。

5.网页 JS 脚本使用 CORS 跨域功能的最简单做法

- 1) 客户的网站服务器只需要提供静态网页功能，完全不需要任何服务器端程序来处理数据。
- 2) Web 用户打开网页后，网页上的 JS 脚本通过 CORS 跨域功能到我们云服务器登录，然后自己负责读写数据填入网页即可。所有的数据读写全部由网页上的 JS 脚本完成。（我们有示例提供）

6.推荐的最佳作法流程（较复杂，但性能最好）

除了网页生成逻辑以外，对每个 GRM 设备有一个独立的“变量读取器”模块，该模块里面有：

- 登录得到的当前有效 SID
- 变量缓存，包括设备中的所有变量名和对应的变量值
- 写变量队列，保存所有从网页发来的写操作
- 网页最近使用时间（方便知道什么时候没人使用网页了，不需要再读取），但如果需要记录历史数据就不需要最后使用时间，因为必须不停的读，不然就没有历史数据了。

变量读取器的内部流程：

- 1) 到 GRM 服务器登录，得到 SID
- 2) 用这个 SID 定时读取 GRM 设备的全部变量（第一次是正常读；以后都用重复读命令，无需变量名）
- 3) 读到的变量放在自己的缓存里面，供生成页面的时候使用。
- 4) 如果写变量队列里面有内容，就发出 GRM 的写命令，把数据写到 GRM 数据服务器
- 5) 不论读或写，如果见到错误码(常见的是错误 8，表示上次登录的 SID 已经过期)，就重新跳到步骤 1) 来登录，否则继续定时读/写
- 6) 如果网页最近使用时间距离现在已经超过一定时间（例如半个小时，或者你们自己愿意的某个时间），表示很久都没有人使用网页了，不需要再读了。停止变量读取器，以便停止 GPRS 流量。

网页生成逻辑的内部流程：

- 1) 客户服务器收到用户的网页请求
- 2) 检查服务器内部的**变量读取器**是否启动，如果没有就启动**变量读取器**
- 3) 从变量读取器中得到数据的最新值，生成网页，送给网页用户。同时更新网页最近使用时间。
- 4) 如果网页有写操作，把变量名和对应的值加入到**变量读取器**的写变量队列，并且写入变量缓存。

四. 程序返回的错误码和错误描述汇总

错误码	错误描述
0	无错误
1	设备序号输入错误
2	设备密码不可为空
3	设备密码错误
5	设备不支持此功能。基站定位/历史报警/历史数据相关命令可能出现此错误。这些命令要求模块固件支持相应命令，并且在工程配置中启用对应功能才能使用。
6	当前设备序号未开通此功能。
7	设备与 PLC 通讯故障(对应于系统变量\$Com1Error 到\$Com3Error 中的一个或几个，以及\$NetComError)
8	网络异常或者您太久没有操作，请重新登录（实际就是 SID 非法，见到此错误码请重新登录，不要管是什么原因。实际原因可能有很多种，包括服务器维护重启，或者服务器故意踢掉客户端，以防止客户端忘了处理该错误） 如果登录的时候没有正确识别返回的 ADDR=xxxx，把命令发到了错误的数据服务器，也会收到该错误码。 注意: 除了登录命令需要发到固定的登录服务器，其他命令都要发到登录后返回的 ADDR 内容中的服务器。
9	该设备没有登录信息，请确认您的设备是否已经开机并连接到网络
10	该设备是透传状态，无变量信息
12	变量无法写入（也就是说，这个变量是只读的，不可写入）
13	写入的变量值非法
14	设备已经失去连接（设备已经登录过，然后临时通讯故障）
15	客户端数目过多（每个 GRM 设备在同一时间最多只能登录 100 个 SID）
16	命令参数格式错误 (如果读变量或写变量请求对于某个变量返回此错误，表示这一行的变量名找不到。如果英文变量名正常而中文变量名返回此错误，常见的原因是变量名没有使用 UTF8 编码)
17	服务器内部错误

注意：

在数据服务器上操作时（除了登录之外，其它的所有操作都是在数据服务器上），如果遇到错误码，除非是错误 8，否则其它错误情况请不要马上重新登录！其它的错误情况不会因为重新登录而消除！！

因为每个 GRM 设备只能同时登录 100 个 SessionID，如果遇到某个设备端的临时错误就重新登录，重新登录后当然还会遇到这个错误，然后继续重新登录。。。。。。结果很快就会耗光 100 个 SessionID 配额，以后任何命令都会收到错误 15。即使设备的临时故障早已恢复正常了，也因为 Session 耗光而继续收到错误 15，无穷循环，永远不能恢复正常！

对各种类型错误的推荐处理：

- 登录错误：（1， 2， 3， 6）请检查您的模块序号，密码和权限
- 程序的逻辑错误：（5， 6， 12， 13， 16）请修改您的接口程序，如果没有权限就去开通权限。
- 设备端的临时错误：（7， 9， 10， 14）请定时轮询，等着设备恢复正常状态即可（可以给用户显示设备不在线，错误 7 可以对发生这个错误的变量显示通讯故障）。
- 错误 8：请重新登录。
- 错误 15：如果某个设备 SessionID 数目超限，该设备上所有的 SessionID 的所有操作都会返回错误 15。请不要重新登录！为了保险起见，建议等待 1 分钟后重试之前的操作（希望某些接近 10 分钟的 SessionID 已经过期被删除了，这样数目就不超限了）