Comway Web 云通信开发参考手册

本章节介绍 Comway Web 云通信开发提供的所有应用程序接口、方法以及对应的参数。

Comway Web 云通信开发 API 返回内容分为二种类型：第一种是直接返回一个字符串，用来表示调用的结果；第二种是返回 XML 格式的查询结果。

具体而言，Comway Web 云通信开发 API 支持您的网站/业务系统用以下方式调用 Comway Web 云通信开发 API：

1. 发送数据到 Comway Web 云通信平台，并得到 dtu 下位机的响应结果。
2. 发送控制指令到 Comway Web 云通信平台，并得到响应结果。
3. 下位机数据主动上报。DTU 下位机上报数据以及 DTU 状态变化信息，通过用户提交的 callback 参数指定的链接上报，便于用户取得 DTU 上报数据，与自有系统集成。

API 基本工作流程

若需要使用该 API，用户必须先在 Comway 服务器进行注册后，将注册信息提交云通信开发后台管理员进行 API 开户，对 Comway Web 云通信开发 API 的每个调用流程都分为以下 2 个步骤：

调用请求的鉴权

将 Comway 服务器用户 ID 赋值给 userid，密码通过 md5 加密后的密文赋值给 pass 这一必填变量，并用 HTTP GET/POST 方法发送给 Comway Web 云通信开发 API。Comway Web 云通信开发 API 根据收到的 *pass* 以及用户 ID 验证请求的合法性和有效性。

请求的处理

对于通过成功鉴权的请求，Comway Web 云通信开发 API 将根据调用的方法进行对应的处理。
对于未通过成功鉴权的请求，Comway Web 云通信开发 API 将拒绝并返回异常信息。

参数说明

所有 Comway Web 云通信开发 API 请求都必须包含以下参数：
pass：用户密码，字符型 您在 Comway 客户端注册的帐号对应的密码通过 MD5 加密后的 32 位字符串。
userid：用户 ID，字符型 您在 Comway 客户端注册的帐号对应的用户 ID。
其他参数说明：
dtuid：Comway dtu 设备编码，数据将发送到对应此设备编码的 dtu。
timeout： 请求的超时时间，默认为 30 秒。
发送数据的 timeout 若为 0，则只将数据下传到下位机，不等待结果，直接返回。
控制指令的 timeout 不能为 0。

返回的状态码（返回信息的前 2 个字节表示状态码）

00：正常，‘00’ + ’,’号分隔 + 相应的 dtu 下位机响应结果。
01：关键参数缺失或者参数赋值错误。
02：用户名或者密码不正确，无法登录 Comway Web 云平台。
03：下位机请求超时。
04：服务器内部错误，请联系 Comway Web 云平台技术支持。
05：指令调用失败，请确认该 DTU 是否在线。
06：用户未在 Comway Web 云通信注册，请先联系管理员注册使用该服务。
07：该 dtu 未在指定账户下，请通过公司无线串口客户端软件在账户下添加该 dtu。
08：指定 dtu 目前是离线状态，请确认该 DTU 是否已经上线，也可以通过无线串口软件查看 dtu 状态。

API 数据主动上报流程

若用户注册是提交了 callback 参数，且为一个正常可以访问的链接，那么该用户下 dtu 的状态信息以及下位机上报的数据会通过这个链接，以 POST 的方式上传数据。Callback 上报数据服务，需通知 Comway Web 云通信的后台管理员开通。

普通用户在发生过 API 的正常请求后，数据的主动上报只能维持一天，若需要持久稳定的数据上报服务，请联系 Comway Web 云通信的后台管理员。

参数说明

所有 Comway Web 云通信开发 API 上报包含以下参数，所有数据均为小写：
userid：用户 ID，字符型 您在 Comway 客户端注册的帐号对应的用户 ID。
dtuid：Comway dtu 设备编码，数据将发送到对应此设备编码的 dtu。
type： 上报数据的类型
0： 下位机透传数据，data 参数的值为 byte 转换为字符串的数据。例如：data=1B76，表示上传数据为：0x1B 0x76
1： DTU 上线，无 data 参数，data=
2： DTU 掉线，无 data 参数，data=
3： DTU 心跳链接，无 data 参数，data=
4： DTU 信号强度，data 参数的值为信号强度。例如:data=26,信号强度为 26
5： GPS 信息，data 参数值为：dtuID,gps 上报时间,纬度(度),经度(度),海拔(米),速度(米/秒),航向(度)。例如：
data=306517170384,2012-07-27 16:10:30,26.06032,101.67215,1583.6,2.4178868,178.9
data： 上报的数据

1 发送 DTU 数据，DTU 透传到下位机并提取响应结果

Comway Web 云通信开发 API 最重要的功能，发送数据到 Comway Web 云通信开发 API 并通过 DTU 透传到下位机。因为在透传下位机的数据以及返回的结果中存在大量 HEX 码，为便于适应不同的服务器，且提高兼容性，统一转换为字符串传输。

请求的链接为：/webapi/senddata，

参数：msgdetail：发送数据的字符串，无论任何数据类型，请按照 byte2string 的方式发送，便于网络传输 HEX 码。例如发送 0x1B 0x76，则转化为字符串“1B76”；若发送字符串“01”即 0x30 0x31 则转化为字符串“3031”发送。若发送字符串为空，则不向串口发任何数据，仅提取超时时间内的下位机上报数据。

返回结果：00+’,’号分隔+相应的 dtu 下位机响应结果，前面 2 个字节 00 表示状态码，后面收到的字符串请按照 string2byte 的方式转化为 HEX 码。

请求示例

1.向串口发送指令 0x1B 0x76

http://dc.comway.com.cn/webapi/senddata?userid=zhangxunemail@gmail.com&pass=c4ca4238a0b923820dcc509a6f75849b&dtuid=110110000570&msgdetail=1B76

返回：00,00

若返回：00,030a3b,前面 2 个 00 表示响应正常，后面的响应结果为：030a3b→0x03 0x0a 0x3b.

2 发送控制指令

1./webapi/dtusignal: 查询指定 dtu 信号强度，返回各个数据',' 号分隔。
必填参数：dtuid，可选参数：timeout
返回：00,22
说明：前面 2 个 00 表示响应正常，后面表示信号强度 22.

2./webapi/gpsinfo: 查询指定 dtu（包含 gps 模块）的 gps 信息，返回各个数据',' 号分隔。
必填参数：dtuid，可选参数：timeout
返回：响应结果,dtuID,gps 上报时间,纬度(度),经度(度),海拔(米),速度(米/秒),航向(度)
例如：00,306517170384,2012-07-27 16:10:30,26.06032,101.67215,1583.6,2.4178868,178.9

3./webapi/kickdtu: 强制指定的 dtu 离线
必填参数：dtuid
返回：00
说明：dtu 会自动离线，若 dtu 设置了断线重拨，则会自动重拨上线。

4./webapi/listdevice: 按照指定条件查询账户下的 dtu 列表
可选参数：type，默认为返回账户下所有 dtu 信息。可选的值为：all,online,offline.分别为查询所有 dtu，查询所有在线的 dtu,以及查询所有离线 dtu。
返回：查询返回信息采用 XML 格式，编码是 utf-8，例如:

```
<?xml version="1.0"?>
<dtulist>
    <dtu>
    <dtuid>4600000006463940</dtuid>
    <dtuname>测试 1</dtuname>
    <login>2010-09-20 10:09:20</login>
    <heartbeat>2010-09-26 18:09:29</heartbeat>
    <status >Online</status>
    </dtu>
    … …
</dtulist>
```

login: dtu 上线登录时间
heartbeat: dtu 最后一次心跳链接时间
status: 是指 dtu 的连接状态，包括：online(在线)，offline(离线)

请求示例

1.查询指定 dtu 的信号强度：
http://dc.comway.com.cn/webapi/dtusignal?userid=zhangxunemail@gmail.com&pass=c4ca4238a0b923820dcc509a6f75849b&dtuid=307561152395
返回结果：00,26

2. 查询指定 dtu（包含 gps 模块）的 gps 信息：

http://dc.comway.com.cn/webapi/gpsinfo?userid=zhangxunemail@gmail.com&pass=c4ca4238a0b923820dcc509a6f75849b&dtuid=307561152395

返回结果：00,307561152395,2013-08-26 13:47:05,39.97193,116.29842,97.9,0.0,0.0

3. 强制指定的 dtu 离线：

http://dc.comway.com.cn/webapi/kickdtu?userid=zhangxunemail@gmail.com&pass=c4ca4238a0b923820dcc509a6f75849b&dtuid=307561152395

返回结果：00

4.按照指定条件查询 dtu 列表：

http://dc.comway.com.cn/webapi/listdevice?userid=zhangxunemail@gmail.com&pass=c4ca4238a0b923820dcc509a6f75849b&type=all

返回结果（编码 UTF-8）：

```
<?xml version="1.0"?>
<dtulist>
    <dtu>
        <dtuid>307547180461</dtuid>
        <dtuname>dtu 307547180461</dtuname>
        <login></login>
        <heartbeat></heartbeat>
        <status>offline</status>
    </dtu>
    <dtu>
        <dtuid>307547180197</dtuid>
        <dtuname>dtu 307547180197</dtuname>
        <login></login>
        <heartbeat></heartbeat>
        <status>offline</status>
    </dtu>
    <dtu>
        <dtuid>307561152395</dtuid>
        <dtuname>gpsdtu</dtuname>
        <login>2013-08-26 12:42:18</login>
        <heartbeat>2013-08-26 13:35:00</heartbeat>
        <status>online</status>
    </dtu>
</dtulist>
```

3 附录

附录中列举了可能用到的接口函数 ByteToHex，HexToByte 以及 MD5 加密算法接口函数。该函数都是较为通用的接口函数，若有不明白的可以上网搜索一下相关资料。

另外，若想计算用户自己密码的 md5 值，可以访问：
http://md5calculator.chromefans.org/?langid=zh-cn

1. Java 示例，仅供参考

```
public static void vStrToByte( String sIn , int iLen , byte[] bOut )
{
    byte[] b = sIn.getBytes();
    byte b1 , b2;
    for ( int i = 0; i < iLen / 2; i++ )
    {
        b1 = b[2 * i];
        if ( b1 > 0x39 )
            b1 = ( byte ) ( b1 + 9 );
        b1 = ( byte ) ( b1 & 0x0F );
        if ( ( 2 * i + 1 ) == iLen )
            b2 = 0;
        else
            b2 = b[2 * i + 1];
        if ( b2 > 0x39 )
            b2 = ( byte ) ( b2 + 9 );
        b2 = ( byte ) ( b2 & 0x0F );
        bOut[i] = ( byte ) ( ( ( b1 << 4 ) | b2 ) & 0xFF );
    }
}

public static String sByteToStr( byte[] bIn , int iLen )
{
    String sRet = "";
    String sTemp = "0123456789ABCDEF";
    int i = 0 , iTemp = 0;
    byte[] bTemp = sTemp.getBytes();
    byte[] bOut = new byte[2 * iLen];
    for ( i = 0; i < iLen; i++ )
    {
        iTemp = ( bIn[i] >> 4 ) & 0x0F;
        bOut[i * 2] = bTemp[iTemp];
```

```
            iTemp = ( bIn[i] & 0x0F );
            bOut[i * 2 + 1] = bTemp[iTemp];
        }
        sRet = new String( bOut );
        return sRet;
    }

    //MD5 校验算法
    import java.security.MessageDigest;

// 将字符串加密成MD5，32位16进制字串，如"3031209"转成"e043a49740adde7aae4f34818c52528e"
    public static String EncodeMD5Hex(String text) throws Exception
    {
        MessageDigest md = MessageDigest.getInstance("MD5");
        md.update(text.getBytes("utf-8"));
        byte[] digest = md.digest();
        StringBuffer md5 = new StringBuffer();
        for (int i = 0; i < digest.length; i++) {
            md5.append(Character.forDigit((digest[i] & 0xF0) >> 4, 16));
            md5.append(Character.forDigit((digest[i] & 0xF), 16));
        }
        return md5.toString();
    }
```

2. Python 示例，仅供参考

```
    def ByteToHex(byteStr):
        """
        Byte2String: '\x11\x22\x33'->'112233'
        """
        return ''.join([ "%02X" % ord(x) for x in byteStr ]).strip()

    def HexToByte(hexStr):
        """
        String2Byte: '112233'->'\x11\x22\x33'
        """
        Bytes = []
        for i in range(0, len(hexStr), 2):
            Bytes.append(chr(int (hexStr[i:i + 2], 16)))
        return ''.join(Bytes)

    #得到密码的 MD5 加密数
    from hashlib import md5
    md5('password…').hexdigest()
```

3. C#示例，仅供参考

```
//0xAE00CF => "AE00CF"
public static string ToHexString(byte[] bytes)
{
    string hexString = string.Empty;
    if (bytes != null)
    {
        StringBuilder strB = new StringBuilder();
        for (int i = 0; i < bytes.Length; i++)
        {
            strB.Append(bytes[i].ToString("X2"));
        }
        hexString = strB.ToString();
    }
    return hexString;
}

//"AE00CF"=>0xAE 0x00 0xCF
public static byte[] HexToByte(string hexString)
{
    byte[] returnBytes = new byte[hexString.Length / 2];
    for (int i = 0; i < returnBytes.Length; i++)
        returnBytes[i] = Convert.ToByte(hexString.Substring(i * 2, 2), 16);
    return returnBytes;
}

/// 取得一串字符串的md5值
public static string GetMD5Hash(string input)
{
    System.Security.Cryptography.MD5CryptoServiceProvider x = new
System.Security.Cryptography.MD5CryptoServiceProvider();
    byte[] bs = Encoding.GetEncoding("iso8859-1").GetBytes(input);
    bs = x.ComputeHash(bs);
    System.Text.StringBuilder s = new System.Text.StringBuilder();
    foreach (byte b in bs)
        s.Append(b.ToString("x2").ToLower());
    string md5string = s.ToString();
    return md5string;
}
```

4. php 示例，仅供参考

```
//单个整形转换成16进制字符串，例如15->"f"
function SingleDecToHex($dec)
{
    $tmp="";
    $dec=$dec%16;
    if($dec<10)
        return $tmp.$dec;
    $arr=array("a","b","c","d","e","f");
    return $tmp.$arr[$dec-10];
}

//单个字符串转换为整形，例如'0'-> 0
function SingleHexToDec($hex)
{
    $v=ord($hex);
    if(47<$v&&$v<58)
        return $v-48;
    if(96<$v&&$v<103)
        return $v-87;
}

//将16进制字符串的0x03 0xC5 0x01,转换成字符串“03C501”
function Byte2String($str)
{
    if(strlen($str)==0)
      return "";
    $tmp="";
    for($i=0;$i<strlen($str);$i++)
    {
        $ord=ord($str[$i]);
        $tmp.=SingleDecToHex(($ord-$ord%16)/16);
        $tmp.=SingleDecToHex($ord%16);
    }
    $tmp=strtoupper($tmp);
    return $tmp;
}

//将字符串“03C501”转换成16进制字符串的0x03 0xC5 0x01
function String2Byte($str)
{
  if(strlen($str)==0||strlen($str)%2!=0)
      return "";
    $tmp="";
```

```
    for($i=0;$i<strlen($str);$i+=2)
    {
$tmp.=chr(SingleHexToDec(substr($str,$i,1))*16+SingleHexToDec(substr($str,$i+1,
1)));
    }
    return $tmp;
}
```

Php 直接调用 md5 库函数即可。