

1. 简介

1.1 前言

非常感谢您使用我们公司的产品，我们将竭诚为您提供最好的服务。

本手册可能包含技术上不准确的地方或文字错误。

本手册的内容将做定期的更新，恕不另行通知，更新的内容将会在本手册的新版本中加入。

我们随时会改进或更新本手册中描述的产品或程序。

当您阅读该开发手册时，同时应该拿到以下内容：

- 1.HttpUtilib（以下简称 OpenAPI 安全认证库（C#））开发包，包含导出库 HttpUtilib.dll 以及使用说明文档。
2. 基于 Visual Studio 2015 的 HttpUtilib 源代码及其测试源代码。

1.2 OpenAPI 安全认证库（C#）简介

OpenAPI 安全认证库封装了 HTTP/HTTPS 的 POST 与 GET 方法，提供 C# 调用 OpenAPI HTTP 接口统一的入口。OpenAPI 安全认证库屏蔽了 OpenAPI HTTP 接口签名细节，降低 C# 对接 OpenAPI 的复杂度，使用方只需引入 OpenAPI 安全认证库（C#）即可方便快速的实现 HTTP 通信。

提供源代码，已使用 Visual Studio 2015 以及 .Net4.5.2 基于 Any CPU 平台编译出 HttpUtilib.dll，位于 bin 目录中，兼容 32/64 位程序对接。

1.3 运行环境

- 支持 Win7、Win8、Win10 32/64 位操作系统

1.4 更新说明

V1.0.1

- 1、修复无法适配.net core 问题。
 - 2、修复获取二进制数据且数据量较大时数据丢失问题。

V1.0.0

- 1、根据 OpenAPI 安全认证协议，封装 C# 版本的 HTTP/HTTPS 请求安全认证过程。

2. 接口定义

2.1 设置平台信息接口

接口名称：

```
void SetPlatformInfo(string appkey, string secret, string ip, int port = 443, bool isHttps = true)
```

接口描述：

统一设置 HTTP/HTTPS 请求的平台参数信息。

参数说明：

[in] appkey: 合作方 APPKey
[in] secret: 合作方 APPSecret
[in] ip: 平台 IP 地址
[in] port: 平台端口，视现场环境、平台实际端口指定。如果是 HTTP 协议，一般端口为 80；如果是 HTTPS 协议，一般端口为 443，是使用 HTTP 还是使用 HTTPS 协议由 isHttps 参数控制
[in] isHttps: 是否启用 HTTPS 协议, true-使用 HTTPS 协议 false-使用 HTTP 协议, 使用 HTTP 还是 HTTPS 协议需视现场环境以及平台配置决定

返回值：

无。

备注：

使用其它接口之前需要通过此接口设置平台信息。

示例（使用 HTTPS 协议）：

详见[基于 OpenAPI 安全认证库（C#）接口的使用示例](#)。

2.2 POST 请求接口

接口名称：

byte[] HttpPost(string uri, string body, int timeout)

接口描述：

封装 HTTP/HTTPS POST 请求，提供 POST 请求统一入口。

参数说明：

[in] uri: POST 请求的 URI, 如/artemis/api/vms/v1/videoParam
[in] body: POST 请求体
[in] timeout: 请求超时时间, 单位: 秒

返回值：

返回字节数组的响应。

备注：

返回值是 byte[] 数组，OpenAPI 接口的实际返回值有两种情况：

- 1、请求返回 json 字符串，此时将字节数组转换成 json 字符串即可，详细方法可参考示例代码。
- 2、请求返回字节流，根据 OpenAPI 接口的实际业务逻辑进行处理，如果是图片流，可直接将 byte[] 数组存成图片文件。

如何判断返回值是以上两种情况中的哪种：

建议先将 byte[] 直接转换成 json 字符串，如果抛出异常，则认为返回的是字节流，详细方法可参考示例代码。
uri 标识的接口是否可能返回字节流，请前往<https://open.hikvision.com/docs> 软件产品选择具体使用的产品，在 API 列表找到具体的接口，查看接口说明。

使用此接口前需要设置平台信息，详见[设置平台信息](#)接口。

示例：

详见[基于 OpenAPI 安全认证库（C#）接口的使用示例](#)

2.3 GET 请求接口

接口名称：

`byte[] HttpGet(string uri, int timeout)`

接口描述：

封装 HTTP/HTTPS GET 请求，提供 GET 请求统一入口。

参数说明：

[in] uri : GET 请求的 URI，需要自行拼接参数，如
`/artemis/api/resource/v1/cameras/indexCode?cameraIndexCode=a10cafaa777c49a5af92c165c95970e0`

[in] timeout: 请求超时时间，单位：秒

返回值：

返回字节数组的响应。

备注：

返回值是 byte[] 数组，OpenAPI 接口的实际返回值有两种情况：

- 1、请求返回 json 字符串，此时将字节数组转换成 json 字符串即可，详细方法可参考示例代码。
- 2、请求返回字节流，根据 OpenAPI 接口的实际业务逻辑进行处理，如果是图片流，可直接将 byte[] 数组存成图片文件。

如何判断返回值是以上两种情况中的哪种：

建议先将 byte[] 直接转换成 json 字符串，如果抛出异常，则认为返回的是字节流，详细方法可参考示例代码。
uri 标识的接口是否可能返回字节流，请前往<https://open.hikvision.com/docs> 软件产品选择具体使用的产品，在 API 列表找到具体的接口，查看接口说明。

使用此接口前需要设置平台信息，详见[设置平台信息](#)接口。

示例：

详见[基于 OpenAPI 安全认证库（C#）接口的使用示例](#)

3. OpenAPI 安全认证库（C#）使用说明

3.1 使用前提

- 基于 Visual Studio2015 与 .Net4.5.2 开发，需要确保本地已安装 .Net，如果版本不是 .Net4.5.2，请自行修改依赖 .Net 版本重新编译；
- 已获取到综合安防管理平台 IP 地址、端口号以及 APPKey、APPSecret。如未获取到相关信息，可联系综合安防管理平台系统管理员获取，或通过[如何获取 APPKey 和 APPSecret](#) 获取 APPKey 和 APPSecret。

3.2 使用 OpenAPI 安全认证库（C#）

3.2.1 基于源码生成 DLL 文件

OpenAPI 安全认证库（C#）提供基于 Visual Studio 2015 的源码，基于源码生成 DLL 文件步骤如下：

- 1、解压 OpenAPI 安全认证库（C#），打开解决方案文件 HttpUtilib.sln（如打开解决方案文件时因使用方 VS 版本较高需要升级，可能存在警告或者错误的情况，请自行根据升级结果文档解决），如下图：

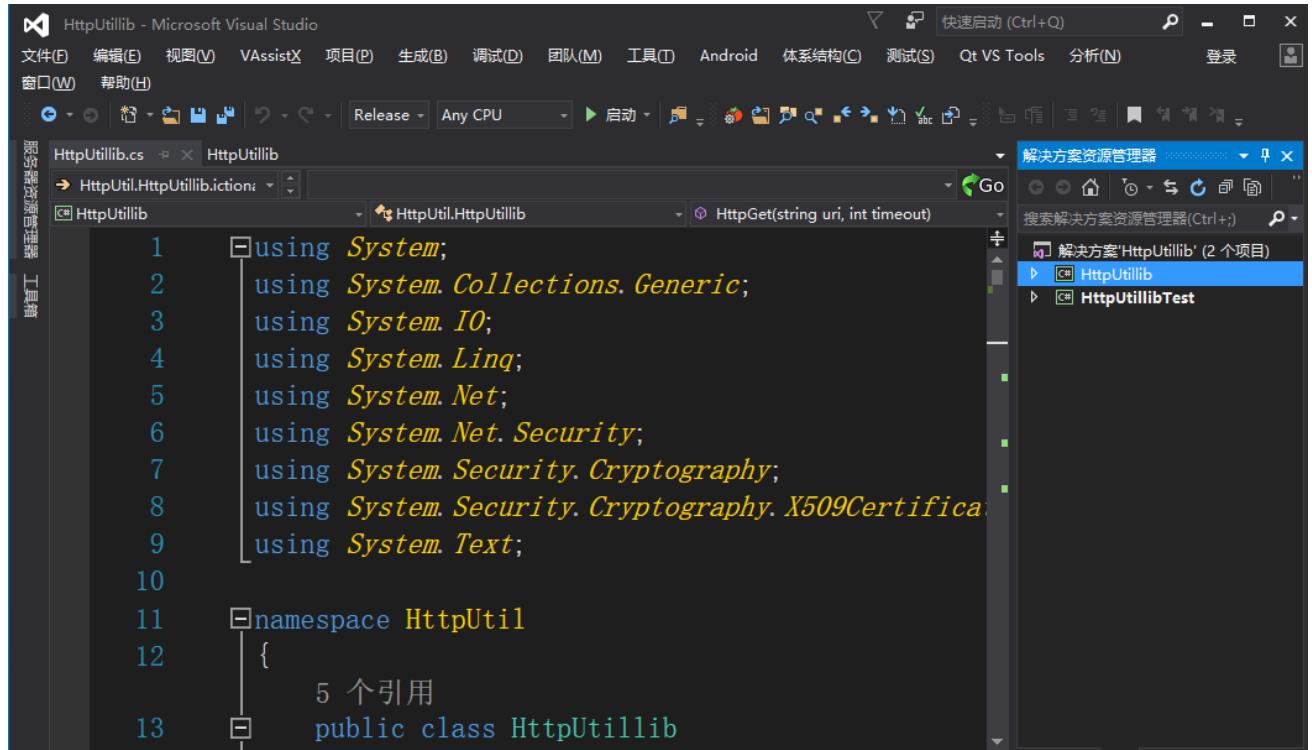


图 3.2.1-1 HttpUtilib 工程

- 2、选择 Any CPU 模式下的平台。选择“生成”->“生成 HttpUtilib”（或英文模式下的“Build”->“Build HttpUtilib”），生成的 DLL 位于 bin 目录下。生成 HttpUtilib 以及生成结果如下图：



图 3.2.1-2 生成 HttpUtillib

3.2.2 使用 Demo

OpenAPI 安全认证库 (C#) 提供基于 Visual Studio 2015 的 demo 源码，并已配置输出 bin 目录，以及依赖的 HttpUtilib。基于 Demo 源码编译步骤如下：

- 1、解压 OpenAPI 安全认证库（C#），打开解决方案文件 HttpUtilib.sln（如打开解决方案文件时因使用方
VS 版本较高需要升级，可能存在警告或者错误的情况，请自行根据升级结果文档解决），如下图：

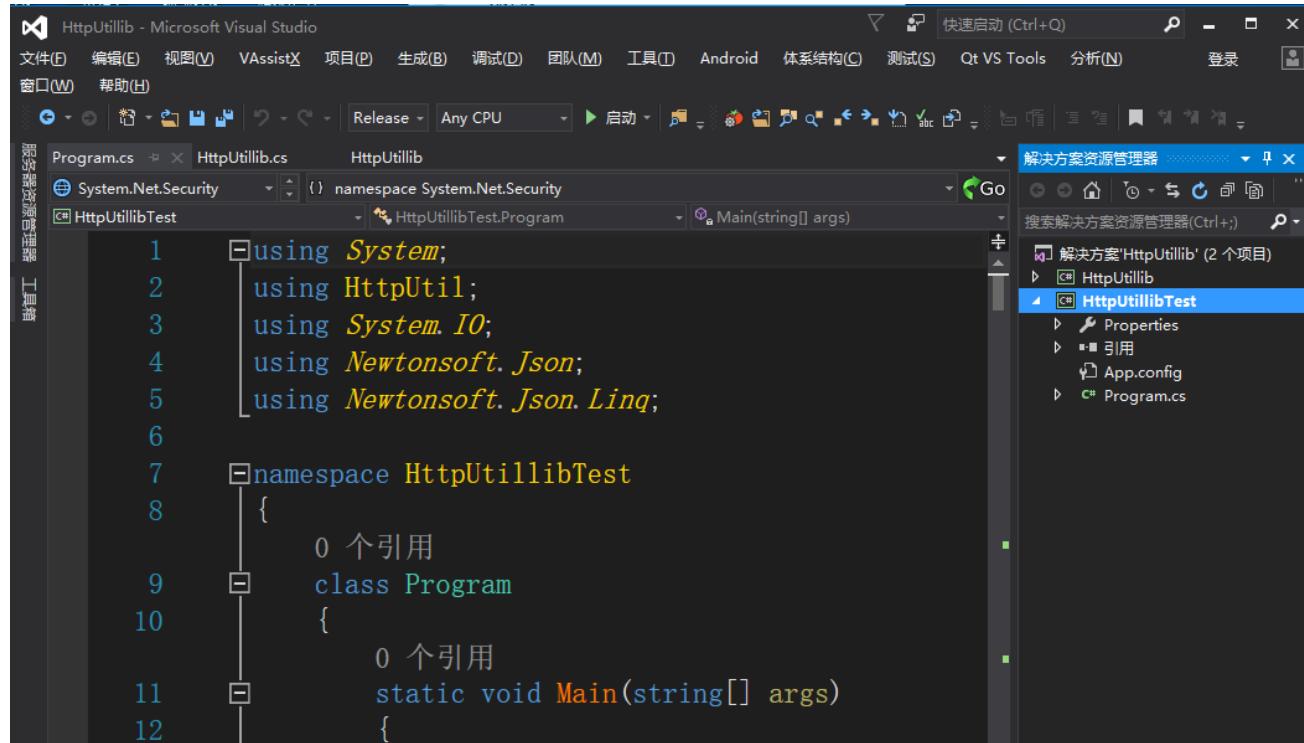


图 3.2.2-1 HttpUtilib Demo

- 2、选择 Any CPU 模式下的平台。选择“生成”->“生成 HttpUtilib”（或英文模式下的“Build”->“Build HttpUtilibTest”），生成的 EXE 位于 bin 目录下。生成 HttpUtilib 以及生成结果如下图：

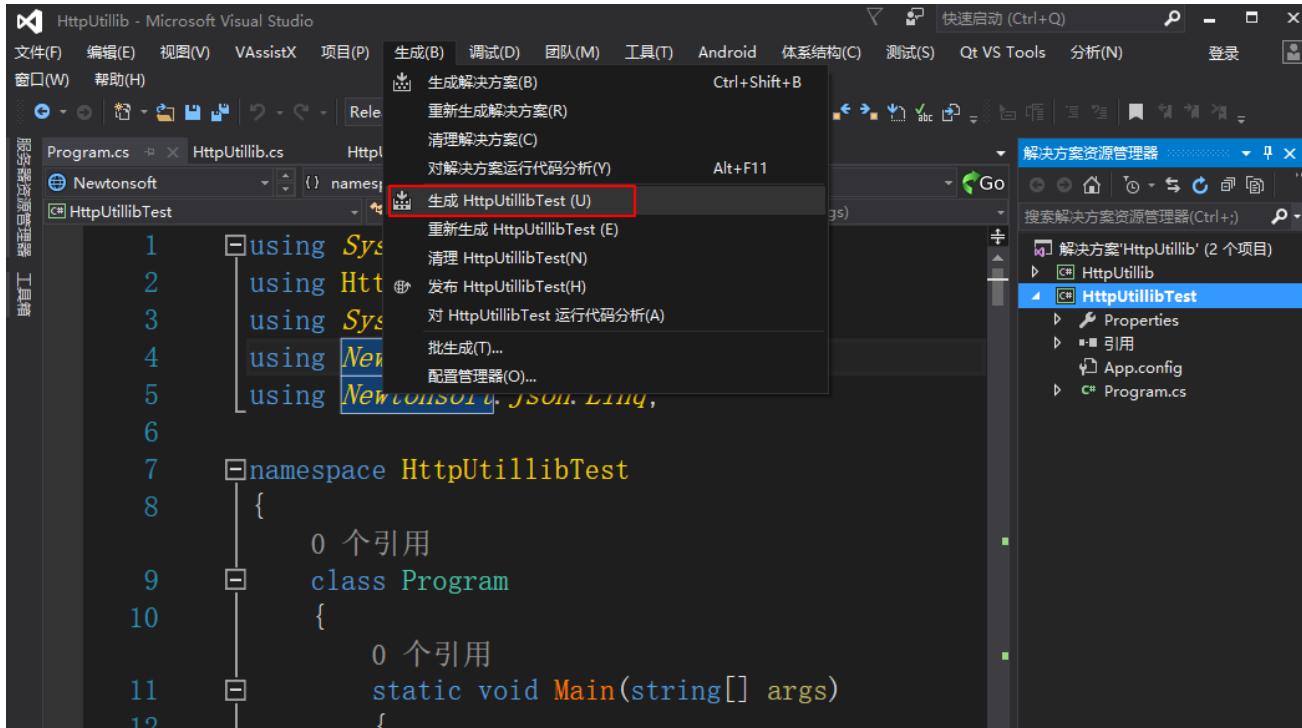


图 3.2.2-2 HttpUtilib Demo 工程属性

3.3 基于 OpenAPI 安全认证库（C#）接口的使用示例

以下为根据监控点编号查询监控点信息、根据图片 URL 下载图片数据使用示例。完整示例代码详见 HttpUtilibTest。

1、引入 HttpUtilib

```
using HttpUtilib
```

2、设置平台信息参数

```
// 设置平台参数：合作方 APPKey、合作方 APPSecret、平台 IP、平台端口、以及是否适用 HTTPS 协议
```

```
// 只要平台信息参数一致，多个请求只需设置一次参数
```

```
HttpUtilib.SetPlatformInfo("28730566", "HSZkCJpSJ7gSUYr06wVi", "10.119.132.75", 443, true);
```

3.1 示例：POST 请求根据监控点编号查询监控点信息

```
// step1：组装POST请求body
```

```
string body = "{\"cameraIndexCode\": \"a10caf77c49a5af92c165c95970e0\"}";
```

```
// step2：填充Uri
```

```
string uri = "/artemis/api/resource/v1/cameras/indexCode";
```

```
// step3：发起POST请求，超时时间15秒，返回响应字节数组
```

```
byte[] result = HttpUtilib.HttpPost(uri, body, 15);
```

```

// step4: 处理响应

if (null == result)
{
    // 请求失败, 这里打印失败日志
    Console.WriteLine("/artemis/api/resource/v1/cameras/indexCode: POST fail");
}
else
{
    //// 注意: 使用方应当知道哪个Uri返回的是字节流（如根据图片Url获取图片数据流）,
    //// 哪个是返回字符串（请前往https://open.hikvision.com/docs软件产品的接口描述中查看）,
    //// 如果肯定是返回字符串的, 直接转成字符串即可; 如果是返回字节流的, 有可能因为请求失败
    //// 了返回json报文, 因此对于这种情况需要将字节流转换成字符串判断是否存在失败的情况,
    //// 如果存在失败则按字符串处理, 否则认为返回的是字节流
    //// 本例中根据监控点编号查询监控点信息, 不存在返回字节流的情况, 直接转换成字符串来处理即可

    Console.WriteLine(System.Text.Encoding.UTF8.GetString(result));
}

```

3.2 示例：POST 请求根据图片 URL 下载图片数据

```

// step1: 组装POST请求body
string body = "{\"svrIndexCode\": \"c2cbb247-452e-4278-a341-729dbcd85e50\", \"picUri\":
\"/pic?2dd461i36-e*1a943077c165m6ep=t=i1p*i=d1s*i=d2b*i2d26*8b74cebd6-9d80ee--1329088z719s=7if6=\\"}";

// step2: 填充Uri
string uri = "/artemis/api/visitor/v1/record/pictures";

// step3: 发起POST请求, 超时时间15秒, 返回响应字节数组
byte[] result = HttpUtillib.HttpPost(uri, body, 15);

// step4: 处理响应

if (null == result)
{
    // 请求失败, 这里打印失败日志
    Console.WriteLine("/artemis/api/visitor/v1/record/pictures: POST fail");
}
else
{
    //// 注意: 使用方应当知道哪个Uri返回的是字节流（如根据图片Url获取图片数据流）,
    //// 哪个是返回字符串（请前往https://open.hikvision.com/docs软件产品的接口描述中查看）,
    //// 如果肯定是返回字符串的, 直接转成字符串即可; 如果是返回字节流的, 有可能因为请求失败
}
```

```
//// 了返回json报文，因此对于这种情况需要将字节流转换成字符串判断是否存在失败的情况，  
//// 如果存在失败则按字符串处理，否则认为返回的是字节流  
//// 本例中下载图片数据，返回的是图片字节流，需要转换成字符串来判断是否存在错误
```

```
// 这里需要先将byte[]转换成字符串，对字符串判断是否可以转换为json对象，如果可以则说明请求失败，否则成功
string tmp = System.Text.Encoding.UTF8.GetString(result);
JObject obj = null;
try
{
    obj = (JObject) JsonConvert.DeserializeObject(tmp);

    // 说明是字符串，并且请求失败了
    Console.WriteLine(tmp);
}
catch (Exception e)
{
    // 转换成json对象异常说明响应是字节流
    File.WriteAllBytes("D://test.jpeg", result);
    Console.WriteLine("写入图片成功: D://test.jpeg\n");
}
```

3.3 示例：GET 请求根据监控点编号查询监控点信息

```
// step1: 填充Uri
string uri =
"/artemis/api/resource/v1/cameras/indexCode?cameraIndexCode=a10cafaa777c49a5af92c165c95970e0";

// step2: 发起POST请求，超时时间15秒，返回响应字节数组
byte[] result = HttpUtilib.HttpPost(uri, body, 15);

// step3: 处理响应
if (null == result)
{
    // 请求失败，这里打印失败日志
    Console.WriteLine("/artemis/api/resource/v1/cameras/indexCode: POST fail");
}
else
{
    //// 注意：使用方应当知道哪个Uri返回的是字节流（如根据图片Url获取图片数据流），  

    //// 哪个是返回字符串（请前往https://open.hikvision.com/docs软件产品的接口描述中查看），  

    //// 如果肯定是返回字符串的，直接转成字符串即可；如果是返回字节流的，有可能因为请求失败
```

```
//// 了返回json报文，因此对于这种情况需要将字节流转换成字符串判断是否存在失败的情况，  
//// 如果存在失败则按字符串处理，否则认为返回的是字节流  
//// 本例中根据监控点编号查询监控点信息，不存在返回字节流的情况，直接转换成字符串来处理即可
```

```
Console.WriteLine(System.Text.Encoding.UTF8.GetString(result));  
}
```

4. 附录

4.1 如何获取 APPKey 和 APPSecret

- 1、请前往 <https://open.hikvision.com/docs>，选择软件产品；
- 2、根据具体对接的产品（如综合安防管理平台），进入对接指南；
- 3、对接指南中查阅“快速上手”章节，在“开发前准备”中根据获取 AK/SK 步骤获取合作方 APPKey 和合作方 APPSecret；