**RESTful Web API调用设计文档**

艾普阳科技(深圳)有限公司

2017

# 概述

## 目的

## 范围

## 定义

## 参考

|  |  |
| --- | --- |
| **编号** | **描述** |
|  |  |
|  |  |

# 用户行为设计

## 构造请求消息

* **构造GET请求消息**

HttpClient lnv\_HttpClient

// 构造GET请求消息(支持所有头信息设置)

lnv\_HttpClient.SetRequestHeader(“Accept”, “\*/\*”)

lnv\_HttpClient.SetRequestHeader(“Accept-Encoding”, “gzip”)

lnv\_HttpClient.SetRequestHeader(“Accept-Language”, “zh-CN,zh”)

lnv\_HttpClient.SetRequestHeader(“Connection”, “keep-alive”)

lnv\_HttpClient.SetRequestHeader(“User-Agent”, “Chrome/60.0.3112.113”)

lnv\_HttpClient.SetRequestHeader(“Cache-Control”, “no-cache”)

**……**

// 发送GET请求消息

li\_rc = lnv\_HttpClient.SendRequest(“GET”, “https://www.appeon.com/users?id=100&name=evan”)

// 获取响应数据

if li\_rc = 1 and lnv\_HttpClient.GetResponseStatusCode() = 200 then

lnv\_HttpClient.GetResponseBody(ls\_string)

lnv\_HttpClient.GetResponseBody(lblb\_blob)

end if

* **构造POST请求消息**

HttpClient lnv\_HttpClient

String ls\_json = {“id”:”1001”, “name”:”evan”, “age”:30}

// 构造POST请求消息(支持所有头信息设置)

lnv\_HttpClient.SetRequestHeader(“Content-Type”, “application/json;charset=UTF-8”)

// Content-Length头信息一般由SendRequest自动设置

**……**

// 发送POST请求消息(数据加到Body并自动设置Content-Length头信息)

li\_rc = lnv\_HttpClient.SendRequest(“POST”, “https://api.appeon.com/employee/update”, ls\_json)

// 获取响应数据

if li\_rc = 1 and lnv\_HttpClient.GetResponseStatusCode() = 200 then

lnv\_HttpClient.GetResponseBody(ls\_ReturnJson)

end if

## 发送请求消息

* **发送GET请求消息**

HttpClient lnv\_HttpClient

// 发送GET请求消息

li\_rc = lnv\_HttpClient.SendRequest(“GET”, “https://www.appeon.com/users?id=100&name=evan”)

// 获取响应数据

if li\_rc = 1 and lnv\_HttpClient.GetResponseStatusCode() = 200 then

lnv\_HttpClient.GetResponseBody(ls\_string)

lnv\_HttpClient.GetResponseBody(lblb\_blob)

end if

* **发送POST请求消息**

HttpClient lnv\_HttpClient

String ls\_json = {“id”:”1001”, “name”:”evan”, “age”:30}

// 发送POST请求消息(数据加到Body并自动设置Content-Length头信息)

li\_rc = lnv\_HttpClient.SendRequest(“POST”, “https://api.appeon.com/employee/update”, ls\_json)

// 获取响应数据

if li\_rc = 1 and lnv\_HttpClient.GetResponseStatusCode() = 200 then

lnv\_HttpClient.GetResponseBody(ls\_ReturnJson)

end if

* **发送PUT请求消息**

HttpClient lnv\_HttpClient

// 发送PUT请求消息(数据加到Body并自动设置Content-Length头信息)

li\_rc = lnv\_HttpClient.SendRequest(“PUT”, “https://api.appeon.com/file/put”, lblb\_data)

// 获取响应数据

if li\_rc = 1 and lnv\_HttpClient.GetResponseStatusCode() = 200 then

lnv\_HttpClient.GetResponseBody(ls\_ReturnJson)

end if

* **发送DELETE请求消息**

HttpClient lnv\_HttpClient

// 发送DELETE请求消息(数据加到Body并自动设置Content-Length头信息)

li\_rc = lnv\_HttpClient.SendRequest(“DELETE”, “https://api.appeon.com/file/delete”, lblb\_data)

// 获取响应数据

if li\_rc = 1 and lnv\_HttpClient.GetResponseStatusCode() = 200 then

lnv\_HttpClient.GetResponseBody(ls\_ReturnJson)

end if

* **发送其它请求消息**

HttpClient lnv\_HttpClient

lnv\_HttpClient.SendRequest(“{Method Name}”, “https://api.appeon.com/test1”)

lnv\_HttpClient.SendRequest(“{Method Name}”, “https://api.appeon.com/test2” , lblb\_data)

## 获取响应消息

HttpClient lnv\_HttpClient

// 发送GET请求消息

li\_rc = lnv\_HttpClient.SendRequest(“GET”, “https://www.appeon.com/users?id=100&name=evan”)

// 获取响应消息

if li\_rc = 1 then

// 获取响应状态信息

li\_StatusCode = lnv\_HttpClient.GetResponseStatusCode()

ls\_StatusText = lnv\_HttpClient.GetResponseStatusText()

// 获取头信息

ls\_AllHeaders = lnv\_HttpClient.GetResponseHeaders() // 获取所有头信息

ls\_ContentType = lnv\_HttpClient.GetResponseHeader(“Content-Type”) // 获取指定头信息

// 获取响应数据

lnv\_HttpClient.GetResponseBody(ls\_string)

lnv\_HttpClient.GetResponseBody(lblb\_blob)

end if

## 大数据发送/接收

* **发送大数据**

HttpClient lnv\_HttpClient

// 构造POST请求消息

lnv\_HttpClient.SetRequestHeader(“Content-Length”, “9999999”)

// 发送POST大数据请求消息

if lnv\_HttpClient.PostDataStart(“https://api.appeon.com/file/upload”) = 1 then

for i = 1 to li\_PackCount

li\_rc = lnv\_HttpClient.PostData(lblb\_NextData, 1024)

if li\_rc != 1 then break

next

end if

if li\_rc = 1 then

li\_rc = lnv\_HttpClient.PostDataEnd()

end if

// 获取响应数据

if li\_rc = 1 and lnv\_HttpClient.GetResponseStatusCode() = 200 then

lnv\_HttpClient.GetResponseBody(ls\_ReturnJson)

end if

* **接收大数据**

HttpClient lnv\_HttpClient

// 设置发送请求后不自动读取响应数据(默认为true会自动读到Body)

lnv\_HttpClient.AutoReadData = false

// 发送GET请求消息

li\_rc = lnv\_HttpClient.SendRequest(“GET”, “https://api.appeon.com/file/download?id=1”)

// 接收大数据

if li\_rc = 1 and lnv\_HttpClient.GetResponseStatusCode() = 200 then

do while true

li\_rc = lnv\_HttpClient.ReadData(lblb\_NextData, 1024)

if li\_rc = 0 then break // 接收数据完成

if li\_rc = -1 then break // 发生错误

loop

end if

## 文件上传/下载

* **上传一个JPG文件**

HttpClient lnv\_HttpClient

// 构造POST请求消息

lnv\_HttpClient.SetRequestHeader(“Content-Type”, “multipart/form-data;boundary=-----------------------------7db372eb000e2”)

// 发送POST请求消息(数据加到Body并自动设置Content-Length头信息)

lnv\_HttpClient.SendRequest(“POST”, “https://api.appeon.com/upload”, lblb\_data)

/\* lblb\_data数据格式如下：

-------------------------------7db372eb000e2

Content-Disposition: form-data; name="file"; filename="me.jpg"

Content-Type: image/jpeg

(此处省略me.jpg文件二进制数据...）

-------------------------------7db372eb000e2--

\*/

* **下载一个JPG文件**

HttpClient lnv\_HttpClient

// 发送GET请求消息

li\_rc = lnv\_HttpClient.SendRequest(“GET”, “https://api.appeon.com/download?file=me.jpg”)

// 获取响应数据后创建JPG文件

if li\_rc = 1 and lnv\_HttpClient.GetResponseStatusCode() = 200 then

if lnv\_HttpClient.GetResponseBody(lblb\_data) = 1 then

of\_WriteFile(“c:\me.jpg”, lblb\_data)

end if

end if

## 指定HTTPS加密协议

HttpClient lnv\_HttpClient

Constant Integer SECURE\_PROTOCOL\_TLS1 = 3

// 指定加密协议TLS1.0

lnv\_HttpClient.SecureProtocol = SECURE\_PROTOCOL\_TLS1 // 默认值为ALL

// 发送GET请求消息

lnv\_HttpClient.SendRequest(“GET”, “https://www.appeon.com/users?id=100&name=evan”)

## 指定客户端证书(HTTPS双向认证)

待设计。。。

# 对象接口设计

## Request Functions

### ClearRequestHeaders

**Description**

Clears headers of the request.

**Syntax**

Integer ClearRequestHeaders()

**Argument**

None.

**Return Value**

Integer. Returns 1 if it succeeds and -1 if an error occurs.

### SetRequestHeader

**Description**

Sets the request header.

**Syntax**

Integer SetRequestHeader(string headerName, string headerValue)

**Argument**

* headerName. String specifying the header name.
* headerValue. String specifying the header value.

**Return Value**

Integer. Returns 1 if it succeeds and -1 if an error occurs. If any argument’s value is null, the method returns null.

### GetRequestHeader

**Description**

Gets the request header by name.

**Syntax**

String GetRequestHeader(string headerName)

**Argument**

* headerName. String specifying the header name.

**Return Value**

String. Returns value related to header name. If any argument’s value is null, the method returns null.

### SetRequestHeaders

**Description**

Sets the request all header information.

**Syntax**

Integer SetRequestHeader(string headers)

**Argument**

* headers. String specifying the all header information.

**Return Value**

Integer. Returns 1 if it succeeds and -1 if an error occurs. If any argument’s value is null, the method returns null.

### GetRequestHeaders

**Description**

Gets the request all header information.

**Syntax**

String GetRequestHeaders()

**Argument**

None.

**Return Value**

String. Returns the request all header information.

### SendRequest

**Description**

Sends the request to server.

**Syntax 1**

Integer SendRequest(string methodName, string urlName)

**Argument**

* methodName. String specifying the request method name.
* urlName. String specifying the URL.

**Return Value**

Integer. Returns values as follows. If any argument’s value is null, the method returns null.

* 1 Success
* -1 General error
* -2 Invalid URL
* -3 Cannot connect to the Internet
* -4 Timeout

--------------------------------------------------------------------

**Description**

Sends the request to server.

**Syntax 2**

Integer SendRequest(string methodName, string urlName, string data)

**Argument**

* methodName. String specifying the request method name.
* urlName. String specifying the URL.
* data. String specifying the data. The data Encoding will convert to EncodingUTF8!.

**Return Value**

Integer. Returns values as follows. If any argument’s value is null, the method returns null.

* 1 Success
* -1 General error
* -2 Invalid URL
* -3 Cannot connect to the Internet
* -4 Timeout

--------------------------------------------------------------------

**Description**

Sends the request to server.

**Syntax 3**

Integer SendRequest(string methodName, string urlName, string data, encoding encodingType)

**Argument**

* methodName. String specifying the request method name.
* urlName. String specifying the URL.
* data. String specifying the data.
* encodingType. Encoding specifying the data to send.

**Return Value**

Integer. Returns values as follows. If any argument’s value is null, the method returns null.

* 1 Success
* -1 General error
* -2 Invalid URL
* -3 Cannot connect to the Internet
* -4 Timeout

--------------------------------------------------------------------

**Description**

Sends the request to server.

**Syntax 4**

Integer SendRequest(string methodName, string urlName, blob data)

**Argument**

* methodName. String specifying the request method name.
* urlName. String specifying the URL.
* data. Blob specifying the data.

**Return Value**

Integer. Returns values as follows. If any argument’s value is null, the method returns null.

* 1 Success
* -1 General error
* -2 Invalid URL
* -3 Cannot connect to the Internet
* -4 Timeout

### PostDataStart

**Description**

Starts sending the request with POST method.

**Syntax**

Integer PostDataStart(string urlName)

**Argument**

* urlName. String specifying the URL.

**Return Value**

Integer. Returns values as follows. If any argument’s value is null, the method returns null.

* 1 Success
* -1 General error
* -2 Invalid URL
* -3 Cannot connect to the Internet
* -4 Timeout

### PostData

**Description**

Sends the next request with POST method that passing string data.

**Syntax 1**

Integer PostData(string data, long bufferSize)

**Argument**

* data. String specifying the post data.
* bufferSize. Long specifying the buffer size.

**Return Value**

Integer. Returns values as follows. If any argument’s value is null, the method returns null.

* 1 Success
* -1 General error
* -2 Timeout

--------------------------------------------------------------------

**Description**

Sends the next request with POST method that passing blob data.

**Syntax 2**

Integer PostData (blob data, long bufferSize)

**Argument**

* data. Blob specifying the post data.
* bufferSize. Long specifying the buffer size.

**Return Value**

Integer. Returns values as follows. If any argument’s value is null, the method returns null.

* 1 Success
* -1 General error
* -2 Timeout

### PostDataEnd

**Description**

Ends sending the request with POST method.

**Syntax**

Integer PostDataEnd()

**Argument**

None.

**Return Value**

Integer. Returns values as follows. If any argument’s value is null, the method returns null.

* 1 Success
* -1 General error
* -2 Timeout

## Response Functions

### GetResponseHeader

**Description**

Gets the response header by name.

**Syntax**

String GetResponseHeader(string headerName)

**Argument**

* headerName. String specifying the header name.

**Return Value**

String. Returns value related to header name. If any argument’s value is null, the method returns null.

### GetResponseHeaders

**Description**

Gets the response all header information.

**Syntax**

String GetResponseHeaders()

**Argument**

None.

**Return Value**

String. Returns the response all header information.

### GetResponseBody

**Description**

Gets the response body into string.

**Syntax 1**

Integer GetResponseBody(ref string data)

**Argument**

* data. String variable into which the function returns data. The response data Encoding EncodingANSI!, EncodingUTF8!, EncodingUTF16LE! and EncodingUTF16BE! are supported.

**Return Value**

Integer. Returns values as follows. If any argument’s value is null, the method returns null.

* 1 Success
* -1 General error
* -2 Timeout

--------------------------------------------------------------------

**Description**

Gets the response body into string.

**Syntax 2**

Integer GetResponseBody(ref string data, encoding encodingType)

**Argument**

* data. String variable into which the function returns data.
* encodingType. Encoding specifying the data which is convert to EncodingUTF16LE!.

**Return Value**

Integer. Returns values as follows. If any argument’s value is null, the method returns null.

* 1 Success
* -1 General error
* -2 Timeout

--------------------------------------------------------------------

**Description**

Gets the response body into blob.

**Syntax 3**

Integer GetResponseBody(ref blob data)

**Argument**

* data. Blob variable into which the function returns data.

**Return Value**

Integer. Returns values as follows. If any argument’s value is null, the method returns null.

* 1 Success
* -1 General error
* -2 Timeout

### ReadData

**Description**

Reads the response body to blob.

**Syntax**

Integer ReadData(ref blob data, long bufferSize)

**Argument**

* data. Blob variable into which the function returns data.
* bufferSize. Long specifying the buffer size.

**Return Value**

Integer. Returns values as follows. If any argument’s value is null, the method returns null.

* 1 Success
* 0 Reading data is finished
* -1 General error
* -2 Timeout

### GetResponseStatusCode

**Description**

Gets the response status code.

**Syntax**

Long GetResponseStatusCode()

**Argument**

None.

**Return Value**

Long. Returns the response status code.

### GetResponseStatusText

**Description**

Gets the response status description.

**Syntax**

String GetResponseStatusText()

**Argument**

None.

**Return Value**

String. Returns the response status description.

## Properties

### AutoReadData

**Description**

Specifies whether the program is reads response body automatically. Values are:

TRUE – The program is reads response body automatically.

FALSE – The program is not reads response body automatically.

**Data type**

Boolean

### SecureProtocol

**Description**

Specifies the secure protocol with the integer value. Values are:

0 – All secure protocols. This is a default value.

1 – SSL2

2 – SSL3

3 – TLS1.0

4 – TLS1.1

5 – TLS1.2

**Data type**

Integer

### Timeout

**Description**

Specifies the timeout seconds. The default value is 60 and the 0 is not timeout.

**Data type**

Long.

# 代码设计

此处设计包括两部分：

PBHttpClient LIB库 ：用于实现新增加的PB对象HttpClient的各个方法；

HttpClient DLL库：HttpClient通用功能库的实现部分；

## PBHttpClient LIB库

### 代码结构

如图4-1所示，其对外导出用于实现PB对象HttpClient各个方法的C函数（图中略去导出的C函数）；



图4-1 PBHttpClient LIB组件

### 结构说明

其内部类的调用关系如图4-2所示：



图4-2 PBHttpClient LIB库内部结构

其中IHCRequest、IHCConnect、IHCResponse三个抽象类是HttpClient DLL导出的三个接口类，CPBHttpClient类内部调用三个类的实例来实现HttpClient对象的各个功能。接口调用过程如图4-3所示：



图4-3 HttpClient通用库接口调用流程

PBHttpClient类调用HttpClient通用库（DLL）提供的函数CreateHCConnector()来创建一个IHCConnect接口实例，然后再利用IHCConnecto类的OpenConnect()接口实现对IHCRequest和IHCResponse接口的实例化，接下来就可以利用三个接口类的方法进行操作了。

### 接口定义

|  |  |  |
| --- | --- | --- |
| 类 | 接口 | 说明 |
| 标准 C 函数接口 | Fn\_ResetRequest | 重置请求 |
| Fn\_SetSecureProtocol | 设置安全连接版本 |
| Fn\_SetRequestHeader | 设置请求头 |
| Fn\_GetRequestHeader | 获取请求头 |
| Fn\_SetHeaders | 设置请求头 |
| Fn\_GetRequestHeaders | 获取请求头 |
| Fn\_SendRequest | 发送请求方法 |
| Fn\_StartPostData | 开始发送POST请求 |
| Fn\_PostData | 发送POST请求的数据 |
| Fn\_EndPostData | 结束发送POST请求 |
|  |  |  |
|  |  |  |

## HttpClient DLL库

### 代码结构

HttpClient通用库DLL部分，如图4-3所示，其对外导出三个接口类，周导出一个用于创建IHCConnect对象和一个销毁此对象的接口；



图4-3 HttpClient通用库对外接口

### 结构说明

HttpClientDLL库内部类之间的关系如图 4-4所示：



图4-4 HttpClient通用库内部结构

HttpClient通用库实现部分（DLL库）分三个层次实现，分别是：

**接口层**：对外导出三个接口，IHCConnect、IHCRequest、IHCResponse。其中

* IHCConnect的主要职责：设置Http连接的属性，如SSL版本、HTTP版本、代理配置、连接超时值等，打开连接并创建在此连接上工作的IHCRequest和IHCResponse对象实例。
* IHCRequest的主要职责：设置请求方法、设置请求头信息、设置连接的URL、设置请发送的内容、发送请求等；
* IHCResponse的主要职责：接收数据、获取响应码及描述、获取响应头等；

**业务层**：由CHttpClientWrapper类实现。主要实现各种请求方法的调用逻辑。此导不直接调用底层HTTP相关的API接口。

**功能实现层**：此层实现具体的HTTP各种功能。可以由libcur实现，也可由WinHTTP实现。

### 接口定义

|  |  |  |
| --- | --- | --- |
| 类 | 接口 | 说明 |
| 标准 C 函数接口 | CreateHCConnect | 创建IHCConnect的实例 |
| DeleteHCConnect | 销毁IHCConnect的实例 |
| IHCConnect | SetSSLVer | 设置SSL连接的版本 |
| SetHttpVer | 设置HTTP协议版本 |
| SetTimeout | 设置连接超时值 |
| SetUserAgent | 设置UserAgent |
| SetRedirect | 设置连接是否自动重定向 |
| OpenConnect | 打开连接 |
| CloseConnect | 关闭连接 |
| SetProxyInfo | 设置代理信息 |
| IHCRequest | SetMethod | 设置采用的HTTP请求方法 |
| SetHostName | 设置服务器HostName |
| SetContentLength | 设置要发送的数据长度 |
| SetSendContent | 设置要发送的数据体 |
| SetURI | 设置请求的URI |
| SetPort | 设置请求的端口号 |
| SetAutoReceive | 设置请求发送后是否自动接收数据 |
| Send | 发送请求 |
| SetHeader | 设置单个请求头域 |
| SetHeaders | 一次设置多个请求头域 |
| SetExtraInfo | 设置发送的额外信息 |
| SetContentType | 设置发送的数据的类型 |
| StartSendData | 开始发送数据（用于发送大数据） |
| SendData | 发送数据（用于发送大数据） |
| EndSendData | 结束数据发送（用于发送大数据） |
| IHCResponse | ReceiveData | 接收数据 |
| GetHTTPVer | 获取响应头中的HTTP协议版本 |
| GetStatusCode | 获聚响应头状态行中的状态码 |
| GetStatusText | 获聚响应头状态行中的状态描述 |
| GetResponseHeads | 获取响应头 |
| GetResponseBody | 获取响应数据体 |
|  |  |
|  |  |