

# 使用微软的 C++ REST SDK - 博客 - 伯乐在线



MSDN官网简介：借助 C++ REST SDK（代号“Casablanca”），您可以更轻松地编写与基于云端的服务相连的新式异步 C++ 代码。

## 新的SDK让你可以在C++中使用REST服务

Visual Studio 2013中包括了C++ REST SDK version 1.0（也称为Casablanca）。这个微软的开源项目包括在[CodePlex](#)中，这个项目同时也利用了c++11中的新特点，设计了现代、异步、多平台API，从而简化了基于云端的编程。这是两篇关于C++ REST SDK系列文章的第一篇，我会介绍如何通过SDK来使用REST服务。在下一篇文章中，我将介绍怎样利用SDK来检索和发送JSON文件。

## 理解C++ REST SDK结构

当你需要最佳性能时，你往往会在本地进行评估，那么C++就是最佳选择之一。微软认为C++在云端也是很有价值的，微软最新的C++ REST SDK使得开发者可以通过C++来调用REST服务，从而满足高性能和可扩展的要求。它让你在C++中就可以调用REST服务或者是编写其他与云计算相关的代码。

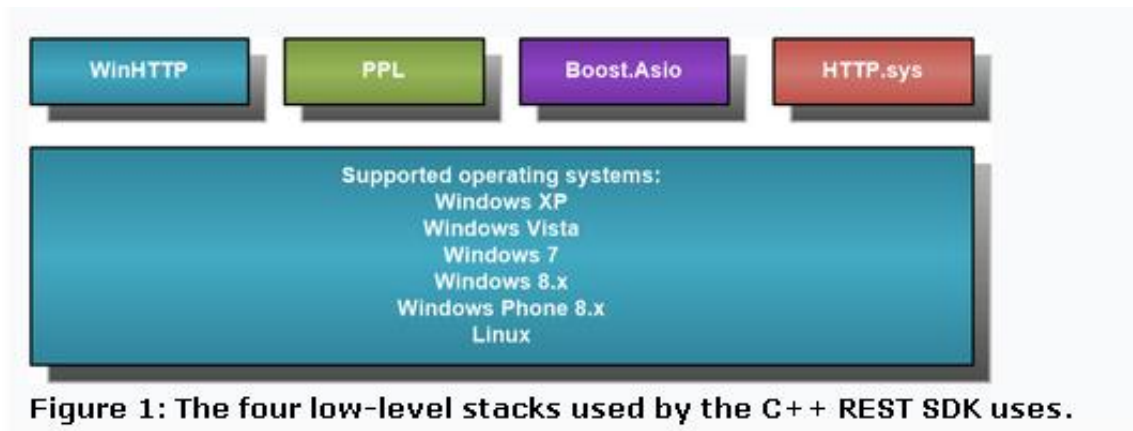
如果你正在使用C++来调用云端服务，但是你在回调函数中使用基于C语言的同步API，那么你没充分利用最新C++版本的高明之处。同时，你的代码可能会很难阅读和调试，同步的API使得你很难设计一个反应敏捷的用户界面（UI）。大部分现代的WEB API都尽量减少不必要的样板，并且提供异步方法。这些方法避免了C风格的回调函数的复杂性。

比如，你正在使用C++ 11，但是你利用它，通过调用一个同步的基于C的API，来进行HTTP的GET调用，那么这样做的效率是不能和其他编程语言比如C#、Python相提并论的。微软基于并行模式库PPL研发了C++ REST SDK，并且利用了PPL的基于任务编程模型。无论何时，你利用C++ REST SDK执行异步操作，你都创建了一个新的PPL任务。为了让C++ REST SDK可以移植到Linux平台上，微软确保PPL的关键部分可以在Linux上运行（并且使用GCC编译）。因此，C++ REST SDK为依赖C++ 11的C++提供了一个并发运行环境。不同于使用回调函数，你可以优雅地编写C++11标准的代码来创建任务并且在其他任务执行完毕后，调度任务。如果你之前有过PPL的编程经验，你会发现使用C++ REST SDK非常简单。

C++ REST SDK基于如下的四个底层栈或是API，这些栈或是API是基于不同操作系统提供的服务。（见Figure1）

- **WinHTTP**:也称为Microsoft Windows HTTP Services. 这是一个基于C语言的HTTP客户端API。
- **PPL**( Parallel Patterns Library的简称): 构成异步操作的编程模型。C++ REST SDK 在不同的Windows版本中使用WinHTTP 。
- **Boost.Asio**: 跨平台的C++库，可以用于网络编程或是底层的I/O编程。它提供了一致的异步模型。该库采用现代C++方法。 C++ REST SDK 使用Boost.Asio 在Linux平台上管理通信。

- **HTTP.sys**: Windows 服务器端的HTTP。C++ REST SDK 在不同的Windows版本中使用 HTTP.sys。

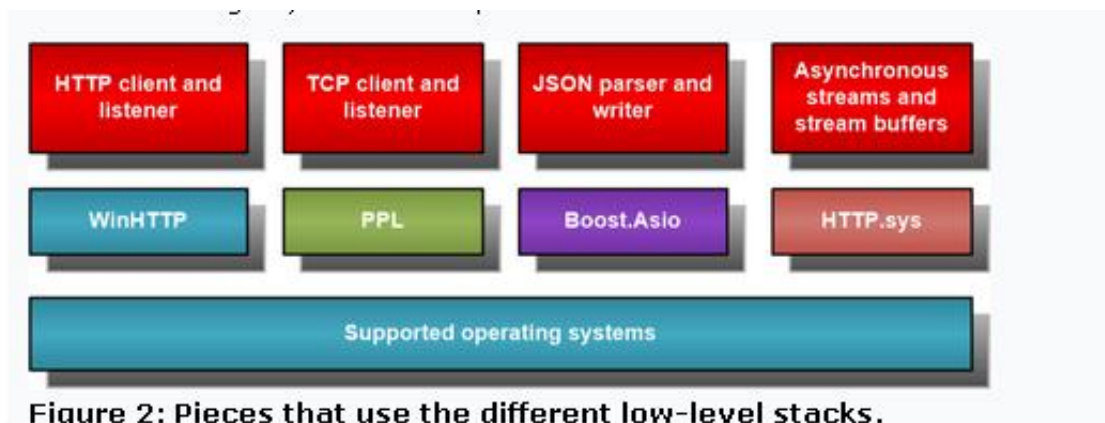


目前（写成本文时），C++ REST SDK支持以下操作系统。特别注意，对Windows XP 系统和Windows Phone 8.x系统的支持，还在试验阶段。

- Windows XP
- Windows Vista
- Windows 7
- Windows 8.x
- Windows Phone 8.x
- Linux

C++ REST SDK还包括如下部分，这些功能使用了不同的底层栈（见Figure 2）：

- **Asynchronous streams and stream buffers(异步流和流缓冲区)**:这部分提供了一种基本的异步流，一种基本的异步流缓冲区，以及许多实现，比如异步文件流。有一些特定的互操作流，可以支持STL iostreams和C++ REST SDK异步流之间的互相操作。
- **HTTP client and listener (HTTP客户端与监听)**:HTTP客户端允许你获得与HTTP服务的连接，并且向服务器发送请求。HTTP监听让你可以从指定的URI接收信息。
- **JSON parser and writer (JSON解析和写入)**:JSON对于像C++之类的静态语言是一个挑战。C++ REST SDK 中使用一个单独的类(`web::json::value`)来表示JSON值，并且提供了必要的操作来帮助数据序列化。你可以在代码执行阶段，声明与JSON数值相关的类型。C++ REST SDK 允许你通过JSON解析从流中读取JSON数据，以及向流中写入JSON数据。
- **TCP client and listener (TCP客户端与监听)**: TCP客户端为TCP网络服务提供了客户端连接。TCP监听会持续监听TCP客户端的网络连接。TCP客户端与监听的最大优势在于，可以完成非阻塞异步操作。但是，这些组件仍然在实验阶段。如果你要使用`System.Net.Sockets::TcpListener`或者`System.Net.Sockets::TcpClient`类，你要适应使用实验版的TCP客户端和监听，以及它们提供的简单的非阻塞异步操作。



微软的C++ REST SDK团队计划在不久的将来，在系统中增加如下部分：

- Web Sockets client and listener. WEB Socket客户端与箭筒
- UDP client and listener. UDP 客户端与监听
- Additional product-specific APIs built on top of the C++ REST SDK, such as Azure Storage, Mobile Services, and Bing Maps. 基于C++ REST SDK的其他产品专用API,比如Azure存储，移动服务和Bing地图。

在Visual Studio 工程中使用C++ REST SDK

在Visual Studio中创建使用SDK的项目并且成功生成该项目，你需要从

CodePlex (<http://casablanca.codeplex.com/>) 下载并安装最新的SDK。在本文中，使用的是1.3版本。

如果你下载的不是这一版本，你要在相应的配置值上做出调整。

版本1.3中默认的安装文件夹是C:\Program Files (x86)\Microsoft Cpp REST SDK for VS 2013 v1.3\SDK。VSProject子文件夹下有两个属性文件与你的Visual Studio工程相对应：CppRest.props和version.props。

当你在Visual Studio中创建了你准备使用C++ REST SDK的工程之后，你必须按照下列步骤操作：

- 将C++ REST SDK安装文件夹下的两个属性文件，即VSProject子文件夹中的CppRest.props以及version.props文件，复制到你创建Visual Studio项目的主文件夹中。
- 打开cpprest.props属性文件，其中的两个属性问价可能有错误值：CppRestSDKPath和CppRestRuntimeBaseFileName。如下图所示，如果你找到了下面的两行，你要将其中110替换为120。在64-bit的Windows版本中，规定了C++ REST SDK基本安装文件夹的属性值是InstallDir，这个属性值在HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Microsoft\CppRestSDK\OpenSourceRelease\120\v1.3\SDK中。不幸的是，一些SDK版本不会检查属性值，如果你不用上述的120替换掉110这个值，就不能成功生成对应工程。因为，不替换对应值的话，在生成项目的过程中，找不到正确的注册表入口。C++ REST SDK中提供的样例，也存在这样的问题，因此你必须在属性文件中替换相应值，才能正确生成项目文件。

未修改属性文件： 修改后的属性文件：

1

2

```
<CppRestSDKPath>$( [MSBuild]::GetRegistryValue(`HKEY_LOCAL_MACHINE\Software\M
```

```
icrosoft\CppRestSDK\OpenSourceRelease\120\v$(CppRestSDKVersionString)\SDK`,  
`InstallDir`))</CppRestSDKPath>  
<CppRestRuntimeBaseFileName>$(CppRestBaseFileName)120$(DebugFileSuffix)_$(CppRestSDKVersionFileSuffix)</CppRestRuntimeBaseFileName>
```

- 现在，打开在你喜欢的文本编辑器中，打开你的C++项目对应Visual Studio 中的vcxproj文件，在Project中加入下面一行。这样，你的项目就可以包含CppRest.props中的定义。值得注意的是，CppRest.props中有一行是引入已经拷贝的CppRest.props配置文件：<Import Project="CppRest.props" />
- 在Visual Studio中打开对应项目，打开项目属性。在Configuration Properties | VC++ Directories中的include Directories属性中加入C++ REST SDK安装路径下include子文件夹的完整路径。如果你使用的是默认安装文件夹，你需要添加的属性值为C:\Program Files (x86)\Microsoft Cpp REST SDK for VS 2013 v1.3\SDK\include。
- 在Configuration Properties | VC++ Directories中Library Directories属性中加入C++ REST SDK安装路径下lib子文件夹的完整路径。如果你使用的是默认安装文件夹，那么你需要添加的属性值为C:\Program Files (x86)\Microsoft Cpp REST SDK for VS 2013 v1.3\SDK\lib。
- 添加C++ REST SDK库。具体来说，在Configuration Properties | Linker | Input中的Additional Dependencies加入cpprest120d\_1\_3.lib文件。

现在，你的C++项目就可以开始和C++ REST SDK一起工作了。（你也可以在 Package Manager Console控制台，通过运行命令Install-Package cpprestsdk来安装最新版本的C++ REST SDK，同时，使用相应包来向已经建立的项目添加C++ REST SDK并完成相应设置。但是，在新项目中使用最新版本SDK的最佳方法，是按照上文所述的步骤进行配置和添加。）

### 通过参数执行一个异步的HTTP GET调用

下面的代码说明了一个Windows控制台应用程序，通过C++ REST SDK调用Flickr。这个例子使用了一个HTTP GET调用：<http://api.flickr.com/services/rest/method=flickr.test.echo&name=value>，并且检索由服务器返回的状态代码，返回报头类型和长度，以及返回主体报文。（在使用这些代码之前，你需要按照前面描述的步骤配置你的Visual Studio项目。）

代码：

```
HTTPGetAsync().wait();  
return 0;  
}
```

代码中的wmain方法在Windows 中称为HTTPGetAsync方法，这个方法返回一个pplx::task<void>实例来表示一个异步操作。在这个例子中，代码调用wait方法来处理返回的实例，因为在这个示例控制台应用中，只要打印出结果即可。但是，一般来说，你可能不会调用wait来处理返回的实例，这样不会阻塞UI操作。

HTTPGetAsync方法创建一个新的web::http::client:http\_client实例，从而保持与Flickr REST API的连接，其中Flickr REST API的URI是<http://api.flickr.com/services/rest/>。The C++ REST SDK使用独立于平台的字符串，所以为了按照不同平台的方式处理字符，代码中使用了U(str)宏来取字符串常量并且转化为与平台相应的类型。typedef utility::string\_t对应的就是平台的特点。

```
http_client client(U("http://api.flickr.com/services/rest/"));
```

在这一句中，代码指定了客户端并且决定了请求对应的URI。其中的路径为/services/rest，它已经包含在用于创建HTTP客户端的URI中了。因此，不需要特别指定这个路径。代码中创建了web::http::uri::builder类的实例，从而允许你用增量的方式建立URIs。这是一种方便而优雅的方法来追加查询片段（通过使用U(str)宏）。两个调用append\_query方法增加了必要的对URI生成器的查询，并首先将它们编码。然后，为生成器调用to\_string方法，生成必要的查询片段来完成请求。

```
1
2
3
4
uri_builder builder;
builder.append_query(U("method"), U("flickr.test.echo"));
builder.append_query(U("name"), U("value"));
auto path_query_fragment = builder.to_string();
```

下面的代码是初始化模块用于创建请求并且返回结果task实例来代表异步操作。request方法表明，它需要建立一个HTTP GET请求（methods::GET），同时需要附加查询片段（path\_query\_fragment）。值得注意的是，then这个方法为执行的请求添加了一个延续的task。这样，延续的任务可以在任务执行之后，将http\_response实例作为一个参数接收。因此，then之后的代码就会自行一个新的task，并且检查请求结果。如果你度过同步代码的话，这里的代码也很容易阅读。很容易看见，在请求task完成之后，代码会被阻塞，直到then执行完毕。你不需要研究复杂的流程，你可以使用then来连接许多任务并且加上需要执行的代码。

```
1
2
3
4
return client.request(methods::GET, path_query_fragment).then([ ]
(http_response response)
{ ... });
```

在本例中，then这个部分中定义的代码是为了展现从服务器端返回的状态码，头部内容的类型和长度。将http\_response作为后一个任务的参数接收，使得一切都易于访问。

```
1
2
3
4
5
6
7
8
9
```

```
std::wostringstream stream;</p>std::wostringstream stream;
stream << L"Server returned returned status code " << response.status_code()
<< L'.' << std::endl;
std::wcout << stream.str();stream.str(std::wstring());
stream << L"Content type: " << response.headers().content_type() <<
std::endl;
stream << L"Content length: " << response.headers().content_length() <<
L"bytes" << std::endl;
std::wcout << stream.str();
```

这里，简单的Flickr API调用，返回的XML如下所示。本例只是想介绍如何通过参数建立一个简单的HTTP GET请求，并且用C++ REST SDK处理返回结果。

```
1
2
3
<rspstat="fail">
<errcode="100"msg="Invalid API Key (Key has invalid format)"/>
</rsp>
```

处理返回结果的代码使用了一个流，来从请求中读取数据。本例中，为了简单，代码调用了 `read_to_end` 方法并且连续调用了 `get` 方法来等待任务中代码执行的结果。`read_to_end` 会先定位到流的尾部，再开始读取，然后返回一个 `task<size_t>` 来说明读入的字符数。因此，你可以异步调用 `read_to_end` 方法，并且调用另一个 `task` 来处理 `then` 中返回的结果。在本例中，`get` 方法等待任务完成，并且返回任务产生的结果。值得注意的是，当 `task` 是一个取消命令时，调用该方法会抛出 `task_canceled` 异常。一般来说，如果你另一个任务中的话，一般不会使用 `get` (也是为了避免阻塞UI)。

```
1
2
3
4
5
6
7
8
9
auto bodyStream = response.body();streams::stringstreambuf sbuffer;auto&
target =
sbuffer.collection();bodyStream.read_to_end(sbuffer).get();stream.str(std::
wstring());
stream << L"Response body: " << target.c_str();
std::wcout << stream.str();
```

正如你从本例中看到的那样，使用C++11和C++ REST SDK执行一个简单的HTTP GET请求相当简单。`then`使得理解异步调用顺序变得简单，C++ REST SDK中不同的帮助使得必要的样板代码很少。



在下一篇文章中，我会提供一个复杂的例子，来说明如何高效使用C++ REST SDK及其帮助和典型的异步操作，来检索和发送JSON数据。另外，我会介绍怎样使用异步流，以及其他REST服务中常用的HTTP方法。

## 合作联系

Email: [bd@Jobbole.com](mailto:bd@Jobbole.com)

QQ: 2302462408 （加好友请注明来意）

## 更多频道

[小组](#) – 好的话题、有启发的回复、值得信赖的圈子

[头条](#) – 分享和发现有价值的内容与观点

[相亲](#) – 为IT单身男女服务的征婚传播平台

[资源](#) – 优秀的工具资源导航

[翻译](#) – 翻译传播优秀的外文文章

[文章](#) – 国内外的精选文章

[设计](#) – UI,网页，交互和用户体验

[iOS](#) – 专注iOS技术分享

[安卓](#) – 专注Android技术分享

[前端](#) – JavaScript, HTML5, CSS

[Java](#) – 专注Java技术分享

[Python](#) – 专注Python技术分享