
文章

[Hao Ma](#) · 五月 31, 2021 阅读大约需 6 分钟

精华文章---HealthConnect中创建HTTP服务

HealthConnect中创建HTTP服务端

这里我说说怎么在HealthConnect上开发HTTP服务。

作为消息引擎，HealthConnect会需要从一个接口接收HTTP请求发送到另一个接口，中间做消息转换，路由等等，目的接口可能是HTTP，或者SOAP，REST等等。这里只介绍HTTP服务的内容，也就是最简单的两种实现：

第一种：实现客户定制的HTTP服务业务服务组件(Business Service)

创建Business Service类，继承EnsLib.HTTP.Service，如下面的示例：

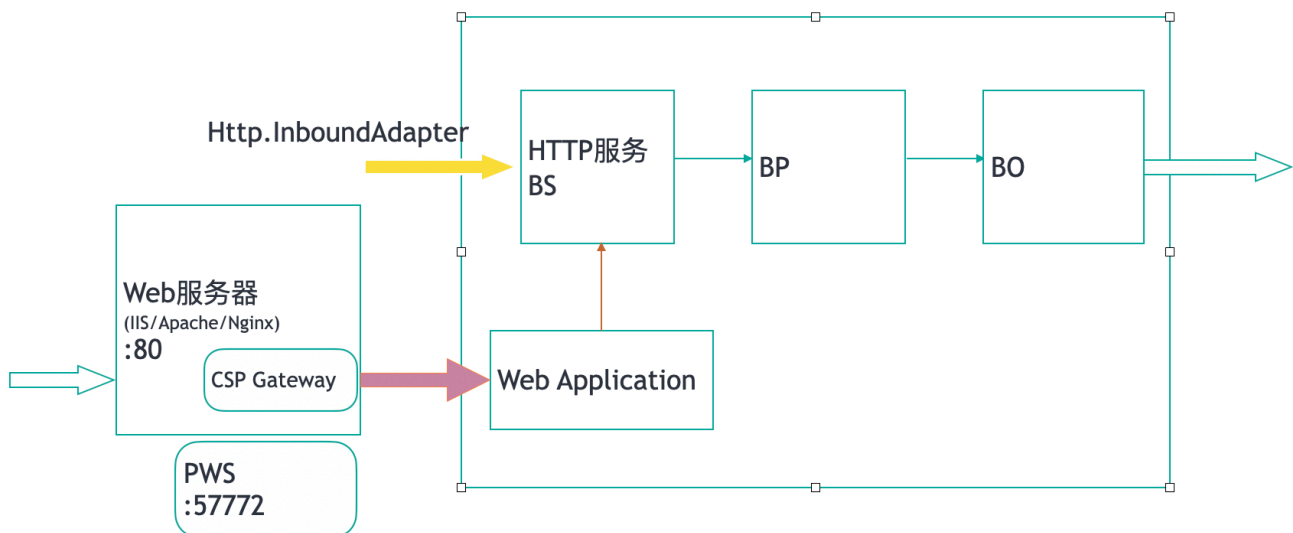
```
Class SEDemo.IO.HTTP.ServiceExample1 Extends EnsLib.HTTP.Service
{
    Parameter ADAPTER;
    Method OnProcessInput(pInput As %Stream.Object, Output pOutput As %Stream.Object)
    As %Status
    {
        //???Ensemble??????????
        set pRequest=##class(Ens.StreamContainer).%New()
        Set tSC=pRequest.StreamSet(pInput)
        set tSC= ..SendRequestAsync("Dummy1",pRequest,.pResponse)
        //????Stream,??????
        set pOutput=##class(%Stream.GlobalCharacter).%New()
        do pOutput.Write("yes, I recieved request")
        Quit tSC
    }
}
```

详细说明：

使用CSP机制接收请求，不要用HTTP的Inbound Adapter,这样能得到

如果您学习过在Ensemble上开发SOAP接口，一定对代码里的"Parameter ADAPTER;"不陌生，它的作用是确定不要使用父类里的Adapter。

EnsLib.HTTP.Service有两个工种模式，一个是使用内置适配器是Ens.HTTP.InboundAdapter，还有一个是使用CSP机制，从CSP Gateway接收请求。下面的图中黄色的箭头是用适配器接收消息，这时业务服务可以定义工种的URL，端口，SSL等等；下面红色的箭头是所谓的" CSP请求"，也就是HTTP请求经过Web服务器，CSP Gateway，到Web Application, 再被EnsLib.HTTP.Service收到。



使用CSP机制有更好的安全性和性能，所以在HealthConnect中任何HTTP的服务端接口我们都推荐CSP机制，包括HTTP接口，SOAP接口，REST接口。这些接口的开发都不要使用对应的InboundAdapter。

有关CSP Gateway的原理，还可以参见在线文档或者我的另一技术文章:[Web Gateway介绍](#)

注意的是：当不使用Adapter时，Production页面的组件配置中很多项目会消失，这些是Adapter的属性，比如允许的IP, 端口，包括编码等等。因为不用Adapter，您也不用定义IP,端口号；只有编码，可以在BS的代码里实现。实现的操作可以参考Adapter的设置：<https://docs.intersystems.com/healthconnectlatest/csp/docbook/Doc.View.c...>

OnProcessInput () 的入参pInput

业务组件收到的HTTP请求由pInput传入，真正的类型是%CSP.GlobalBinaryStream，它的父类是已经不推荐使用的流类型%GlobalBinaryStream。用%Stream.Object作为pInput的对象类型是合适的，这是一个新版本的Stream对象的超类，可以是任意类型的流。上面代码里业务服务组件发出的Ensemble消息的类型是StreamContainer，如果你看看消息跟踪的类型，你会发现里面流的类型是 " GB",也就是一个%GlobalBinaryStream类型的流。

pInput对象的属性Stream里存放的是HTTP Body，而HTTP头放在Attributes属性里，如下图所示：

如何获得请求里的消息头

以下是用pInput.GetAttributeList()得到的Attributes的内容：

```

<![CDATA[*CSPApplication/csp/healthshare/demo/CharEncoding1EnsConfigName SEDemo.IO.H
TTP.ServiceExample1
    HTTPVersion1.1
    HttpRequestGET
        IParamsParamsRawParamsTranslationTableRAWAURL:/csp/healthshare/demo/SEDem
o.IO.HTTP.ServiceExample1.clsaccept*/"accept-encodinggzip, deflatecache-control
    no-cacheconnectionkeep-alivecontent-length0content-typefcookieCSPSESSIONID-SP
-80-UP-csp-healthshare-=0000010100002R0lNxxgwxwUQuPHKQogHGNWfADLfF2xiPwce2s; CacheBrow
serId=ui$4lIZ_rJsPD_xUTPG$Rw; CSPWSERVERID=B33PnJAehost172.16.58.200mykeyimess7postma
n-token&83e942ce-ae73-4d98-b7ab-1e2803c95794%user-agentPostmanRuntime/7.18.0]]>
    
```

而获取消息头的某些值，可以用 pInput.Attributes(httpheadername)得到，比如下面这些：

```
set MethodTypeName=pInput.Attributes("HttpRequest")
set HTTPVersion = pInput.Attributes("HTTPVersion")
set ContentType = pInput.Attributes("content-type")
set ContentLength = pInput.Attributes("content-length")
set URL = pInput.Attributes("URL")
/???GET????URL??form????POST?????????????)
set pList = Attributes("Params",form_variable_name,n),
/????????????????????????????????????????"mykey"
set Mykey = pInput.Attributes("mykey")
```

更多的细节，请参考在线文档: [Using the HTTP Inbound Adap: About the Attributes Array](#)

中文编码转码

如果请求的流里面有中文字符，需要在代码里执行转码，比如下面这个例子，把pInput的流，转码成UTF8,放到Ens.StringRequest里传输。

```
set pRequest=##class(Ens.StringRequest).%New()
set pRequest.StringValue=$ZCVT(pInput.Read(),"I","UTF8")
set sc= ..SendRequestAsync("Dummy1",pRequest,.pResponse)
```

有关组件名称和访问的URL

请求的URL有两个通常的选择：

1. Production里面的业务组件名字和类名称一样

比如上面代码的类是"SEDemo.IO.HTTP.ServiceExample1",如果production的业务服务也是同样的名字，那么调用它的URL就是

```
http://hostip:port/csp/healthshare/demo/SEDemo.IO.HTTP.ServiceExample1.cls
```

其中"/csp/healthshare/demo/"是Web Application的名字。

- 如果上一条不成立。比如写了一个类，用于多个业务服务组件，那么需要组件的名字可以用自己的名字，调用的URL要包含?CfgItem=xxx表示寻找不同的业务组件服务。举例说，还是上面的类，用于添加了两个业务服务组件"http-service2"和"http-service3",访问它们的URL就是:

```
http://hostip:port/csp/healthshare/demo/SEDemo.IO.HTTP.ServiceExample1.cls?CfgItem=http-service2
```

```
http://hostip:port/csp/healthshare/demo/SEDemo.IO.HTTP.ServiceExample1.cls?CfgItem=http-service3
```

还有这么个操作，就是单独创建一个Web Application，配置DispatchClass，来接入一个Web服务。我觉得完全没有必要，而且新版本中Web Application的菜单里只剩下了为REST分配分派类的选择，因此这里就不说这个了。

第二种：使用EnsLib.HTTP.GenericService预置业务服务组件

使用预置的，开发好的组件意味着不用写代码，配置一下就可以使用。相应的，灵活性上就差了，大概只适合简单的透传，转发，路由。如果要修改数据包内容，http头内容，编码转换等等，需要要在其他的组件上，比如production中消息经过的BP, BO中实现。

EnsLib.HTTP.GenericService的父类是EnsLib.HTTP.Service，因此它也是可以使用Adapter机制或者CSP标准机制。如前面所述，我们需要用CSP机制，调用的URL必须包含?CfgItem=**组件配置名**，比如我在Production里面配置了组件 " GenericService1", 那么我访问的请求就应该是：

```
http://localhost/csp/healthshare/demo/EnsLib.HTTP.GenericService.cls?CfgItem=GenericHTTPService1
```

还有，因为我们不修改代码，所以

组件配置项中会有Adapter的配置，比如端口号，SSL配置，直接忽略它们，不用理睬。。如果您看到配置项上有 " 字符集",显示的是UTF-8，可是没起作用，请不要奇怪，这个是Adapter的配置，您用CSP机制时它是不起作用的。

EnsLib.HTTP.GenericService向其他业

务组件发出的Ensemble消息的类型是EnsLib.HTTP.GenericMessage

。以下是一个POST请求被业务服务组件发送给BO的消息样例。

```
<!-- type: EnsLib.HTTP.GenericMessage id: 531 -->
<HTTPMessage xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:s="http://www.w3.org/2001/XMLSchema">
  <Stream>{
    âserial_idâ: âC20114062017025â,
    âtake_timeâ: â2019-01-02 15:04:05â,
    âpos_timeâ: â2019-01-02 15:04:05â,
    âtypeâ: â1â,
    âwarnâ: â0â,
    âfile_idâ: â9cd586eef356c71f64b82a190b469e69â,
    âfile_nameâ: âA1012014400596520160714190338.hlyâ,
    âfile_pathâ: â/service/data/TE9100Yâ,
    âbegin_timeâ: â2016-07-04 20:17:21â,
    âend_timeâ: â2016-07-04 20:18:21â,
    âlengthâ: â60â,
    âsizeâ: â9650â,
    âresultâ: âçª|æ§â¿, æ¬æ¬|â¿µçæµªªè§â¼â,,â
  }</Stream>
  <Type>GB</Type>
  <HTTPHeaders>
    <HTTPHeadersItem HTTPHeadersKey="CSPApplication">/csp/healthshare/demo/</HTTP
HeadersItem>
    <HTTPHeadersItem HTTPHeadersKey="CharEncoding" xsi:nil="true"></HTTPHeadersIt
em>
    <HTTPHeadersItem HTTPHeadersKey="EnsConfigName">GenericHTTPService1</HTTPHead
ersItem>
    <HTTPHeadersItem HTTPHeadersKey="HTTPVersion">1.1</HTTPHeadersItem>
    <HTTPHeadersItem HTTPHeadersKey="HttpRequest">POST</HTTPHeadersItem>
    <HTTPHeadersItem HTTPHeadersKey="IPParams">1</HTTPHeadersItem>
    <HTTPHeadersItem HTTPHeadersKey="IPParams_1">CfgItem=GenericHTTPService1</HTTP
HeadersItem>
    <HTTPHeadersItem HTTPHeadersKey="RawParams">CfgItem=GenericHTTPService1</HTTP
HeadersItem>
    <HTTPHeadersItem HTTPHeadersKey="TranslationTable">RAW</HTTPHeadersItem>
    <HTTPHeadersItem HTTPHeadersKey="URL">/csp/healthshare/demo/EnsLib.HTTP.Gener
icService.cls</HTTPHeadersItem>
    <HTTPHeadersItem HTTPHeadersKey="accept">*/*</HTTPHeadersItem>
    <HTTPHeadersItem HTTPHeadersKey="accept-
encoding">gzip, deflate, br</HTTPHeadersItem>
    <HTTPHeadersItem HTTPHeadersKey="authorization">Basic X3N5c3RlbTptTWVM=</HTTPH
```

```
headersItem>
    <HTTPHeaderItem HTTPHeadersKey="cache-control">no-cache</HTTPHeaderItem>
    <HTTPHeaderItem HTTPHeadersKey="connection">keep-alive</HTTPHeaderItem>
    <HTTPHeaderItem HTTPHeadersKey="content-length">532</HTTPHeaderItem>
    <HTTPHeaderItem HTTPHeadersKey="content-type">text/plain</HTTPHeaderItem>
    <HTTPHeaderItem HTTPHeadersKey="cookie">CSPSESSIONID=SP-52773-UP-csp-healths
hare--=0000000100006fSAbXykPFTd0NFNEoaUH94y07Phmq0V92aeDg; CSPBrowserId=F4nsRY6yDGVipv
Q84sDN3w; CSPWSERVERID=I33xYkI3</HTTPHeaderItem>
    <HTTPHeaderItem HTTPHeadersKey="host">172.16.58.200:52773</HTTPHeaderItem>
    <HTTPHeaderItem HTTPHeadersKey="postman-token">0e0bf4a7-868b-4e69-bbac-
bfd89909fe99</HTTPHeaderItem>
    <HTTPHeaderItem HTTPHeadersKey="user-
agent">PostmanRuntime/7.28.0</HTTPHeaderItem>
</HTTPHeader>
</HTTPMessage>
```

使用这个组件要注意的几点：

- 上面的数据包里有中文乱码，我是故意这么做的，就是提醒您：强调一下选用这个组件时，如果要修改数据，可以在其他的组件处理。如果我做一个简单的HTTP转发，我可以选用EnsLib.HTTP.GenericService和EnsLib.HTTP.GenericOperation这一对预置组件，这样BS,BO就免开发了。而如果中间要做中文编码的转换，我会插入一个BP，专门处理EnsLib.HTTP.GenericMessage里的中文转码。
- 组件的"启动标准请求"配置项：要使用CSP Gateway请求机制时应该勾选。
- "持久消息已发送INPROC"配置项（PersistInProcData）
这个选项是专用于以InProc模式同步调用时是否要持久化Ensemble消息的选项，默认是保存。如果设置成off，那么HealthConnect即不保留消息头，也不保留消息体，在消息查看器里无法查看，也不能重传。
- "保持标准请求分区"配置项
也是用于InProc模式调用，是否保留到BO定义的外部系统的连接。和上面的"持久消息已发送INPROC"配置项一样，很少被用到，保留默认的选中就可以。
- "没有字符集转换"配置项
控制CSP Gateway是否对%reponse消息里的文本按content-type的类型转码，默认是不勾选，也就是保留CSP Gateway的转码功能。
- "One-Way"配置项
如果客户端不期待响应消息，那么选中这个选项后，EnsLib.HTTP.GenericService收到请求转发的同时会送一个Http状态码202，意思是"服务器已接受请求，但尚未处理"。默认是不选中这个配置。

[#互操作性](#) [#新手](#) [#Ensemble](#)

源

URL:

<https://cn.community.intersystems.com/post/%E7%B2%BE%E5%8D%8E%E6%96%87%E7%AB%A0-healthconnect%E4%B8%AD%E5%88%9B%E5%BB%BAhttp%E6%9C%8D%E5%8A%A1>
