記事
Tomoko Furuzono · 2020年11月24日 4m read

例：PHPから Caché Web サービスへの接続

最近、InterSystems 内で PHP から Caché ベースの Web サービスに接続が必要になる事例がいくつかありました。
これらの最初の事例とは、実はこの開発者コミュニティそのものであり、他の InterSystems サイト/アプリケーションとのシングルサインオンに Web サービスを使用しています。
次の例は、パスワード認証を使用して PHP から Caché ベースの Web サービス（具体的には SAMPLES ネームスペースの Web サービス）に接続する方法を示しています。

（注意: この例は、/csp/samples に対してパスワード認証が有効になっていることを前提としています。）

```
<?php
// ユーザー名/パスワード用の標準 SOAP ヘッダー
// 出典元: http://stackoverflow.com/questions/13465168/php-namespaces-in-soapheader...
class WSSESecurityHeader extends SoapHeader {
   public function construct($username, $password)
   {
      $wsseNamespace
= 'http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secex...';
      $security = new SoapVar(
         array(new SoapVar(
            array(
               new SoapVar($username, XSD_STRING,  null, null, 'Username',
$wsseNamespace),
               new SoapVar($password, XSD_STRING,  null, null, 'Password', $wsseNamespace)
            ),
            SOAP_ENC_OBJECT,
            null,
            null,
            'UsernameToken',
            $wsseNamespace
         )),
         SOAP_ENC_OBJECT
      );
      parent::SoapHeader($wsseNamespace, 'Security', $security, false);
   }
}

// Web サービスに接続するための主要パラメーター:
$location = "http://localhost:57772/csp/samples/SOAP.Demo.cls";
$uri = "http://tempuri.org";
$username = "SYSTEM" ;
$password = "SYS";
```

```
//
SOAP クライアントを WSDL
から構築する場合、または（その目的のために存在するさまざまなツールを使用して）
// WSDL を使用して PHP クラスを生成する場合、?WSDL=1 とユーザー名/パスワードは
// URL の最後に追加する必要があります:
$wsdl =  $location . "?WSDL=1&CacheUserName=" . $username . "&CachePassword=" .
$password;

// 理論的に言えば、次のようなことができるようになります。
// $client = new SoapClient($wsdl, array("trace"=>1));
// 多くの場合はこれで十分です。
// ただし、SAMPLES ネームスペース内の SOAP.Demo の WSDL は次のように他のクラスで
// 定義されている追加のスキーマを参照します:
// <s:import namespace="http://tempuri.org/QueryByNameDataSet "
schemaLocation="http://localhost:57772/csp/samples/SOAP.Demo.QueryByName.DS.cls?XSD"/>
// PHP の WSDL 処理で CSP セッション Cookie を使用させたり、これらを取得する際に URL 内で
// 資格情報を渡す方法はないようですので、これは失敗します。 回避策は代わりに location/uri を使用し、
// WSDL なしで SoapClient を作成することです。

// SoapClient オブジェクトを作成
$client = new SoapClient(null, array(
  "trace" => 1,
  "location" => $location,
  "uri" => $uri,
  "style" => SOAP_DOCUMENT,
  "use"  => SOAP_LITERAL));

// セキュリティヘッダーを追加します。
$client->setSoapHeaders(   new WSSESecurityHeader($username, $password));

// ID でPersonを取得します。ここでは URL から取得しますが、デフォルトでは ID
1であるため、常に何かが表示されます。
$id = 1;
if (isset($_GET[  "id"])) {
  $id = $_GET[  "id"];
}

// SOAP 呼び出しのパラメーター名と値を持つオブジェクト
$request = new stdClass();
$request->id = $id;

// これは FindPersonSoapIn のキーを持つ連想配列になります（WSDL を参照）
$params = array();
$params['FindPersonSoapIn']=$request;

// ソープコールのオプションの配列では、参照するWSDLがないので、 soapaction
を指定しなければなりません。
// PHP のデフォルト値 は <uri>#<method> ですが、これは正しくありません。
$options = array('soapaction'=>$uri.'/SOAP.Demo.FindPerson');

// 実際に Web サービスを呼び出します。
$result = $client->soapCall(   "FindPerson",$params,$options);

// 応答をページにダンプします。
```

```
var_dump($result);
?>
```

#Code Snippet #SOAP #フロントエンド #Caché

ソースURL:
https://jp.community.intersystems.com/post/%E4%BE%8B%EF%BC%9Aphp%E3%81%8B%E3%82%89-cach%C3%A9-web-%E3%82%B5%E3%83%BC%E3%83%93%E3%82%B9%E3%81%B8%E3%81%AE%E6%8E%A5%E7%B6%9A