sage

Enterprise Management

Create a Web Portal using Web services

02 2020

The purpose of this document is to explain how to invoke SOAP Web services from a web portal in PHP.

The portal gives you access to your data, such as orders or customer information in real-time. You do not need to share large files across networks or via email. Because the data stays in the application, not saved out to an external server, it is more secure. Remote employees like sales professions can not only view data, but they can also create new data such as sales orders from any computer with internet access via a browser.

You can create the portal using SOAP web services functionality and WampServer® is a Windows web development environment that allows you to create web applications with Apache2, PHP, and a MySQL database. In addition, PhpMyAdmin allows you to easily manage your databases. [Source: wampserver.com].

Audience

This document is intended for experienced Enterprise Management users with administrator level permissions who may or may not have prior experience with publishing web services. There is also a section specifically for developers who have advanced coding and web services knowledge.

Contents

Requirements

To build the PHP web portal, you need the following:

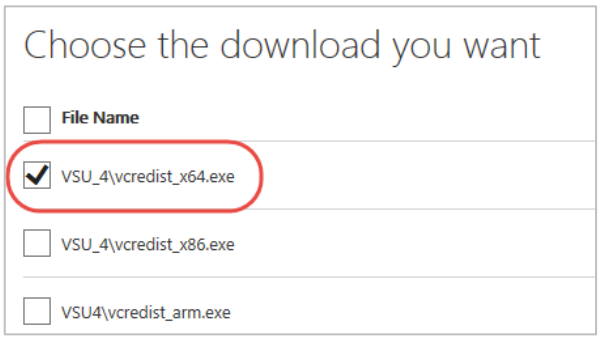
- Windows 64 bit operating system
- Update 9.0.2 or above
- Microsoft Visual C++ 2012 Redistributable (x64)

Install Microsoft Visual C++ 2012 Redistributable (x64)

From https://www.microsoft.com/en-US/download/details.aspx?id=30679

Select your language from the pull-down menu and click **Download**.

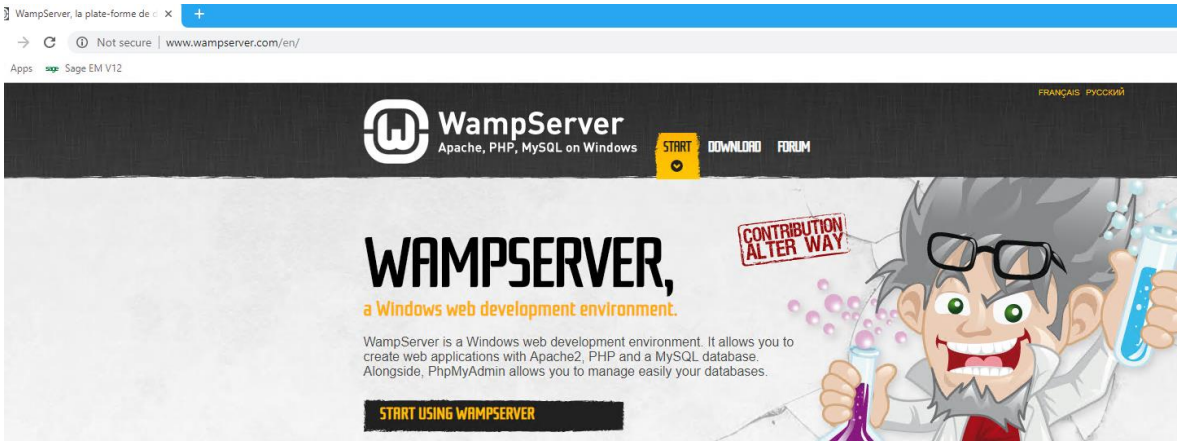
There are multiple files available for this download. When prompted, select **VSU_4/vcredist_x64.exe**.



Build the portal

Install and configure WampServer

You can download WampServer from [www.wampserver.com.](http://www.wampserver.com)



On the homepage, scroll down and download this one:

WAMP SERVER 64 BITS (X64) 3.0.6

The version that is actually downloaded is at least 3.1.4

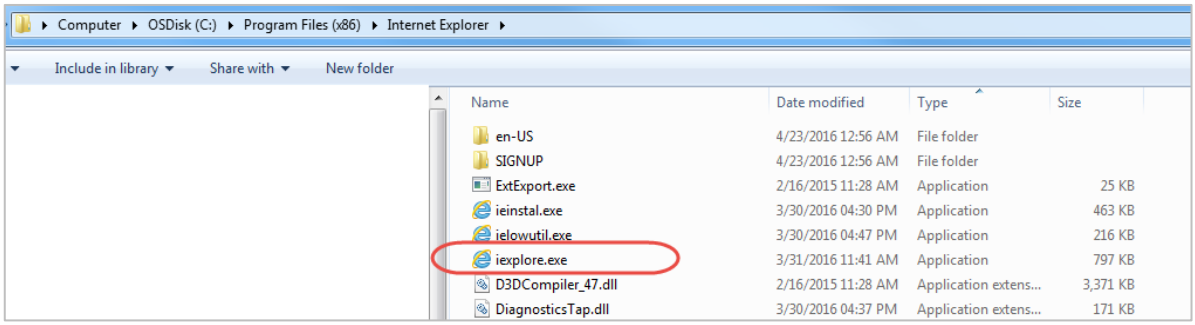
By default, WampServer installs in **C:\wamp64** but it is best to choose **c:\sage\wamp**, or a different folder.

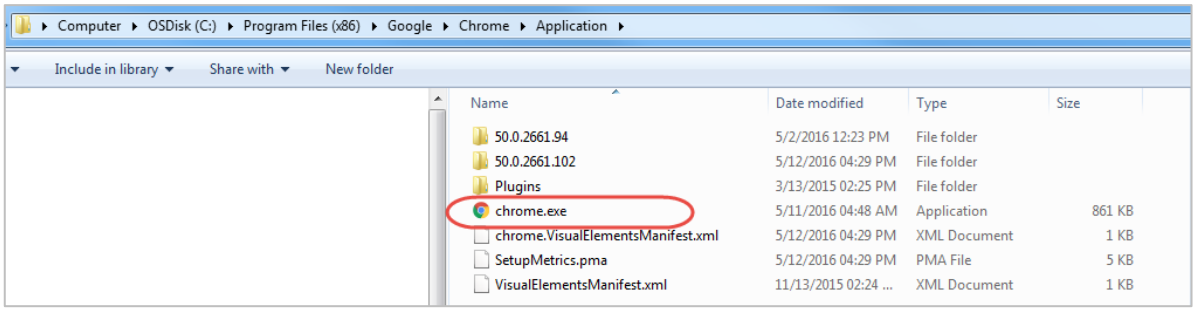
Next, you need to choose a default browser such as Chrome or IE.

Navigate to the EXE file on your computer.

Select the file and click Open.

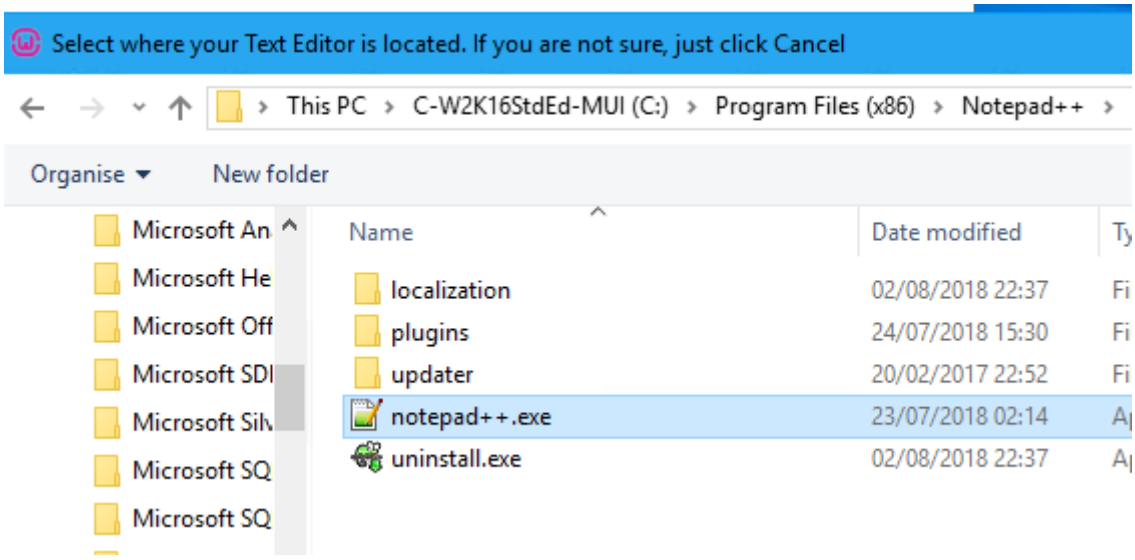
Examples for IE and Chrome are shown below.



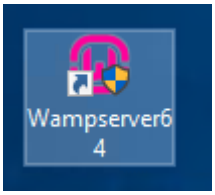


Then you can choose a default text Editor.

Example Notepad + + (32bits)



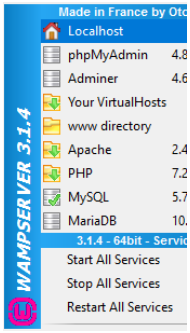
Then on your desktop, this program allows to start, stop or configure the server Wamp



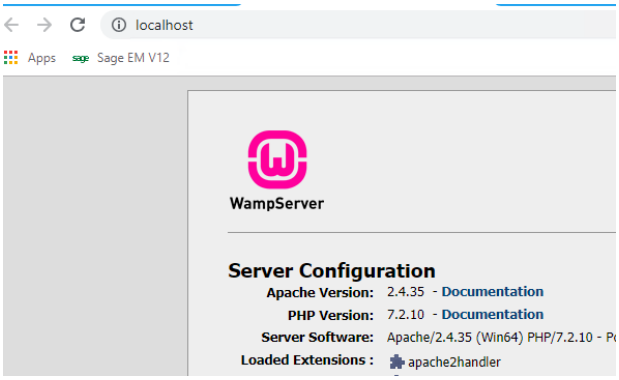
Launch it.
The notification icon changes color and must become Green.

Launch the Wamp server page by clicking on this Icon.

Then select Localhost.

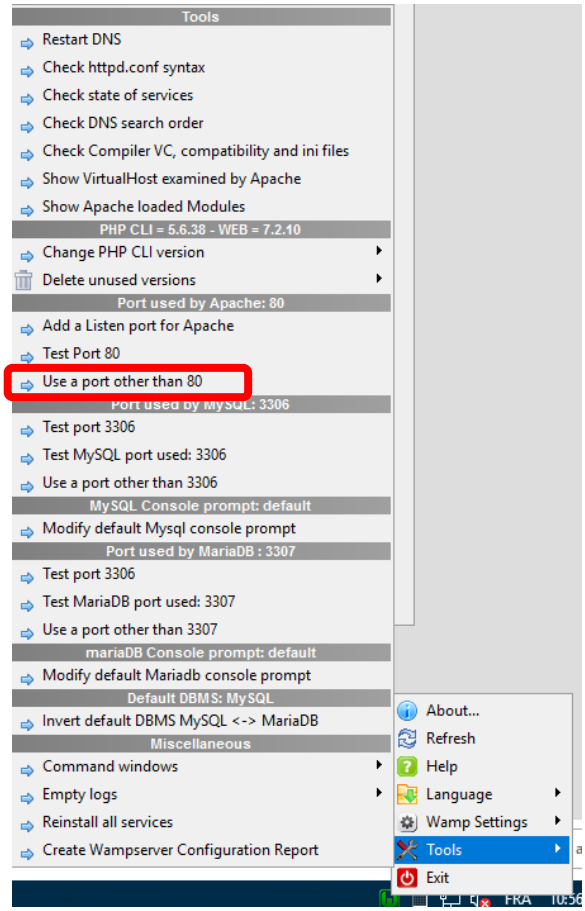


You then have the next page if all goes Well.

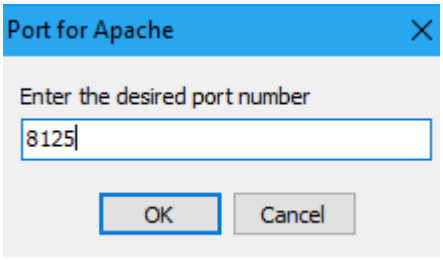


It is necessary to change the default HTTP port, which is **80**. For example, you can change it to **8125** or another port.

Right click on the Wamp icon then Tools then use a port other than 80



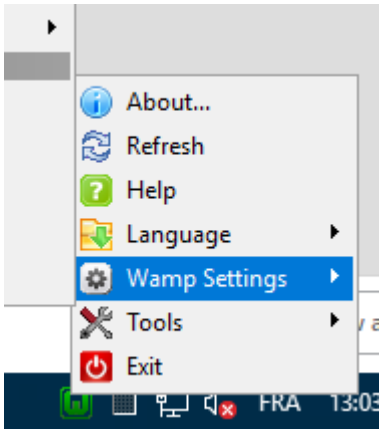
Enter the new port: **8125**



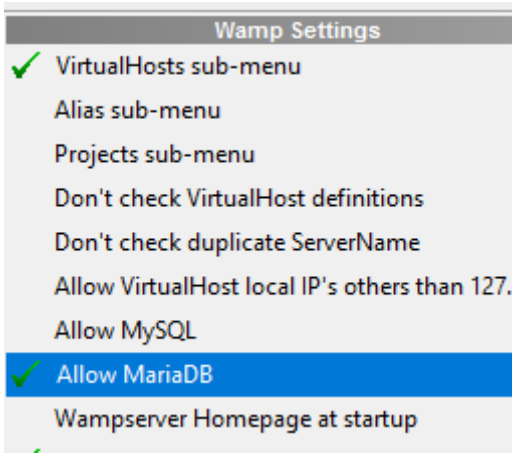
Verify that the page is running on this new port.

The Wamp server includes both MySQL and MariaDB Databases.

To have more resources on our machine, you can stop these two bases.



Deselect Allow MySql and Allow MariaDB



This PHP portal does not use any of these databases

Configure the server and the pool of Web services

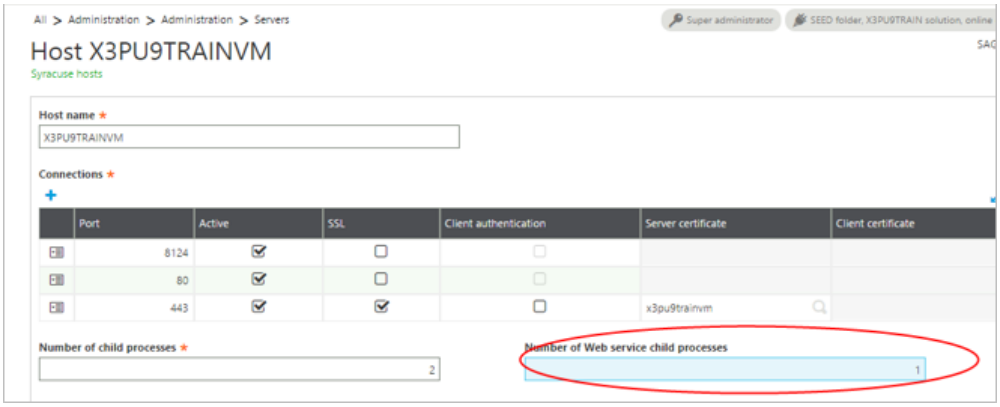
Set up the Syracuse Web server

In Enterprise Management, complete follow these steps:

> **Note**: The Host name etc. are examples. You might have other names.

Open Administration > Administration > Servers > **Hosts**.
Click the edit icon next to your host name.

On the next screen, in the **Number of Web service child processes** field, enter **1**.



Configure the WEB services pool

Open **Classic SOAP pools configuration** from Administration > Administration > Web Services > **Classis SOAP pools configuration**

Click **Create soapClassicPool**.

Complete the following fields:

Alias: Enter the name of the pool to be used in the web service call.

Initialization size: Enter **1**.

> Represents the number of clients (per node.js process) that are initialized during the pool startup.

Maximum size: Enter 1.

Represents the maximum number of clients (per node.js process) that can be started on this pool.

Auto start: check box

If checked, the pool starts when the Syracuse server starts.

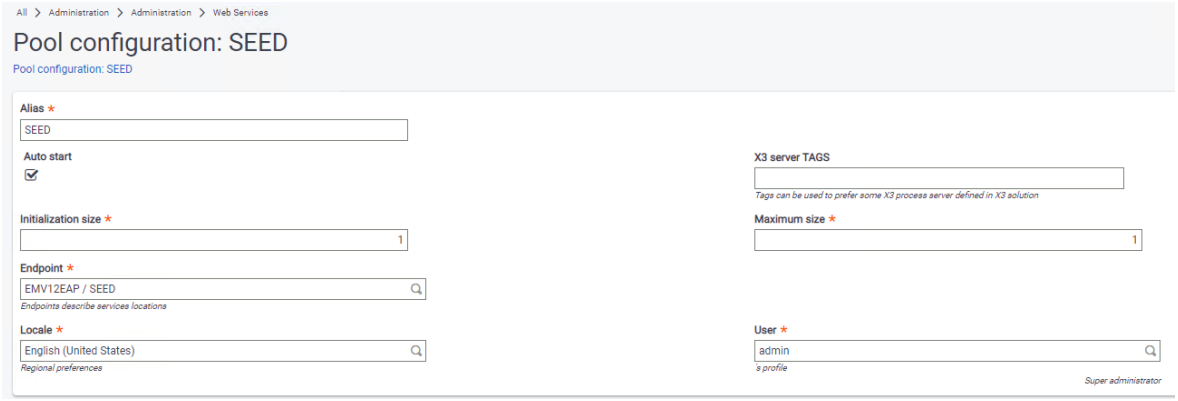
Server TAGS: Leave blank.

This field is best used by Developers with classic SOAP pool configuration.

Endpoint: Enter the endpoint (folder) to be used for web service requests.

Locale: Enter your language and location. (In this example, English.)

User: Enter the user name. In this case **Admin**.

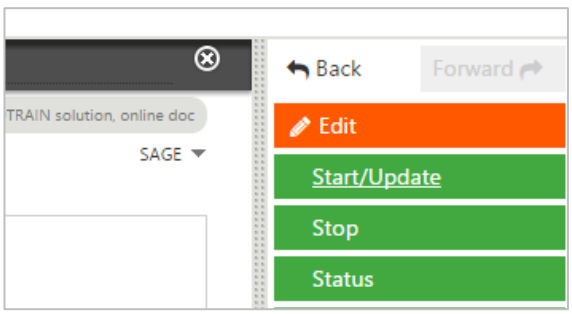


To continue setting up the PHP web portal, you need to start the pool.

After you create the pool based on the previous steps, it displays in the list of soapClassicPools.

Click the name of the pool you just created.

From the Actions panel, click **Start/Update**.



Install and configure the PHP Web portal

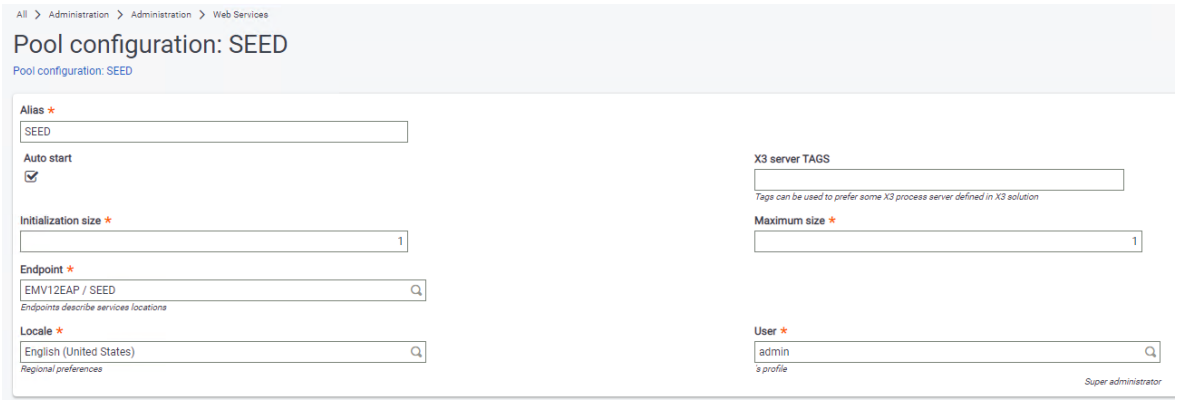
If you have not already done so, start the web service pool you just created. See steps in the previous section for details.

Download the PHP web portal project files

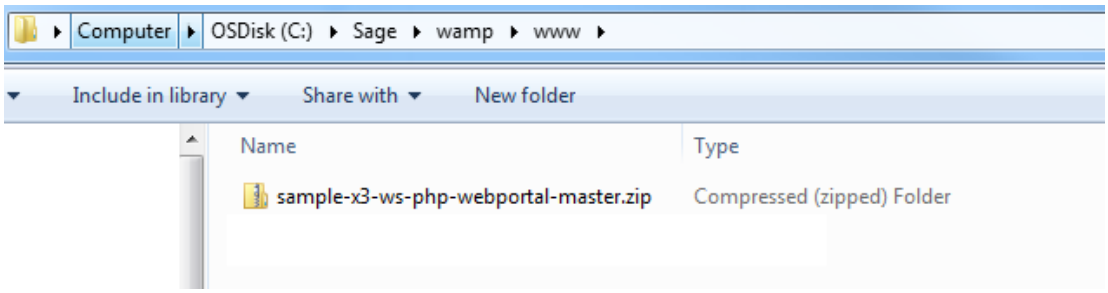
The project file for the PHP web portal is available from GitHub. The project file is open to everyone, so you do not need a GitHub account. The download file contains everything you need to create and configure the portal including the application patch for the YOSOH web service.

From GitHub https://github.com/Sage-ERP-X3/sample-x3-ws-php-webportal, click **Clone or Download**.

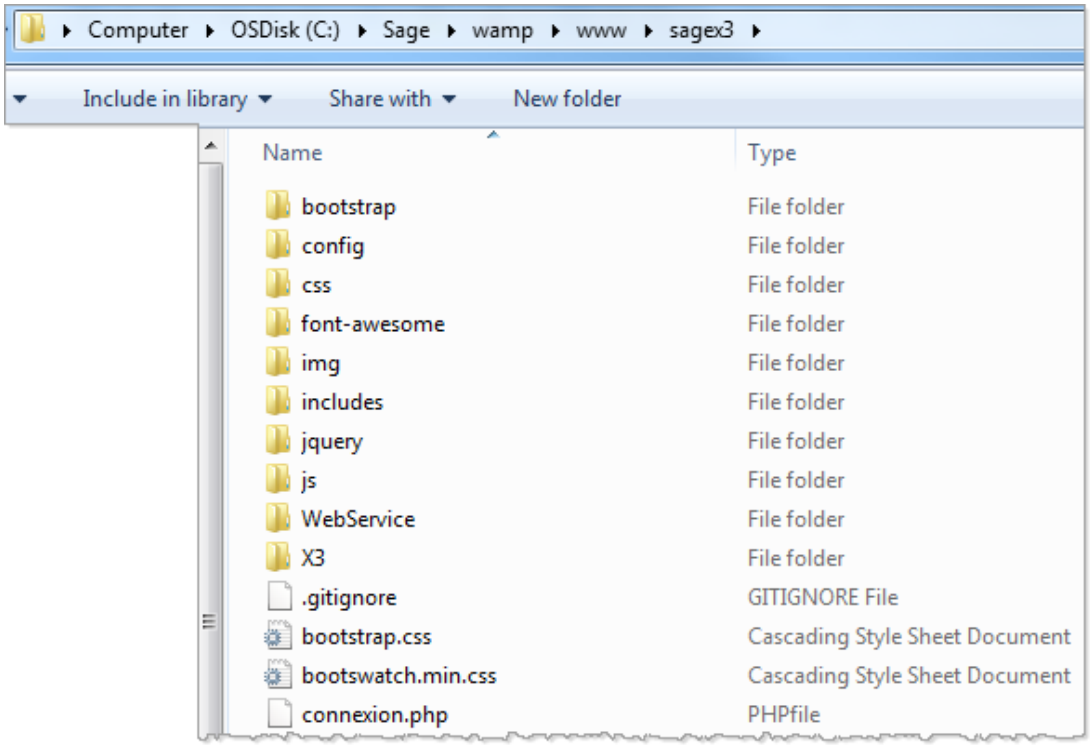
> If you are logged in to GitHub, you have the option **Clone** or **Download** without logged in
> Be sure to download the ZIP file.



Save the **sample-x3-ws-php-webportal-master.zip** file to **C:\Sage\wamp\www**.



Extract all files to **C:\Sage\wamp\www\sagex3**.



Configure the portal

Next, you need to configure the portal to communicate with Enterprise Management.

In the folder **C:\Sage\wamp\www\sagex3\config**, open the **Config.php** file.

The following fields should match what you entered when you configured your web service pool in Enterprise Management:

WSDL : Your server URL

CODE_LANG

POOL_ALIAS

WS_ORDER YOSOH The name of the Sales orders Web services.

WEB_SITE_LOGIN and WEB_SITE_PASSWD represent the credentials you will share with those who will be accessing these web services via the web portal.

```
<?php

class Config {
   public static $WSDL              = "http://x3pu9trainvm:8124/soap-
wsdl/syracuse/collaboration/syracuse/CAdxWebServiceXmlCC?wsdl";
   public static $CODE_LANG         = "ENG";
   public static $CODE_USER         = "…";
```

```
    public static $PASSWORD          = "…";
    public static $POOL_ALIAS        = "SEED";
    public static $REQUEST_CONFIG    = "adxwss.optreturn=XML";
    public static $WS_ORDER          = "YOSOH";

    public static $WEB_SITE_LOGIN    = "websage";
    public static $WEB_SITE_PASSWD   = "websage";

}
?>
```

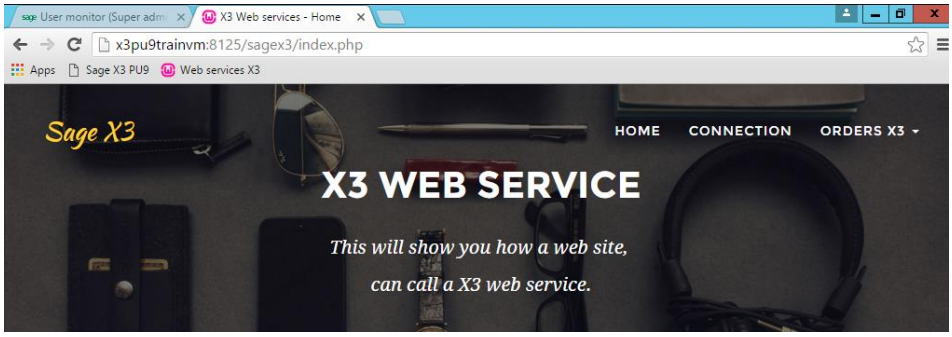
Important! Do not change the punctuation and formatting.

From the WampServer menu**, Restart All Services.**

Enter the URL for your portal in your default browser. In this example the URL is http://x3pu9trainvm:8125/sagex3/

This is the name of Syracuse server and the number was configured in **httpd.conf**.

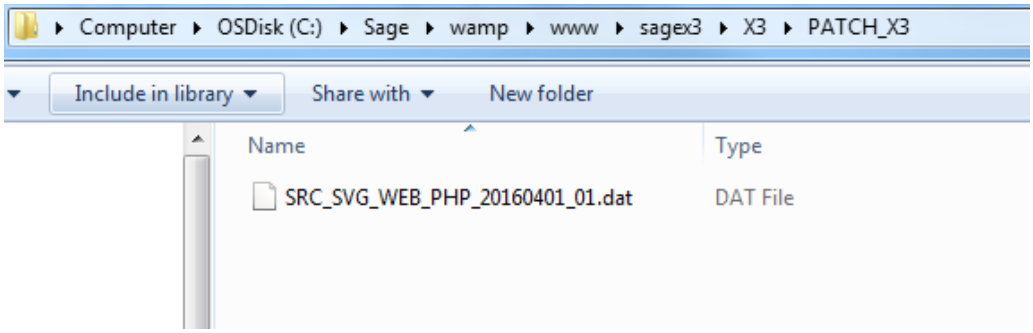
This is an example of what your portal could look like.



Install the application patch

You need to install the patch containing the YOSOH web services. The file was downloaded in the ZIP file from GitHub.

The name of file is **SRC_SVG_WEB_PHP_YYYYMMDD_NN.dat**. It is in the following directory: C:\Sage\wamp\www\X3\PATCH_X3\V12.



Important! You can only install the patch on the SEED folder, not the application folder.

The patch contains the following objects:

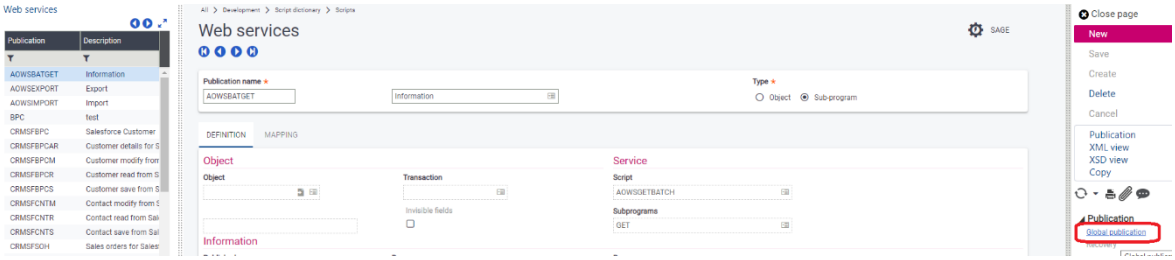
Type	Objects	Comments
ACV	YSWPH	Activity code PHP Web portal
EXE	SUBSLC	Generate Sales entry transaction
TRT	YSWPHPSTOCK	Script Available stock
ASU	YSWPHPSTOCK~STOCK	Sub program YSWPHPSTOCK~STOCK Available stock
AWE	YOSOH	Web service YOSOH Sales orders
AWE	YSSTOCKPHP	Web service YSSTOCKPHP Available stock
SLT	STRTYP=2 & STRNUM='WS'	Sales entry transaction WS: Web service for the web service YOSOH

Publish the Web service

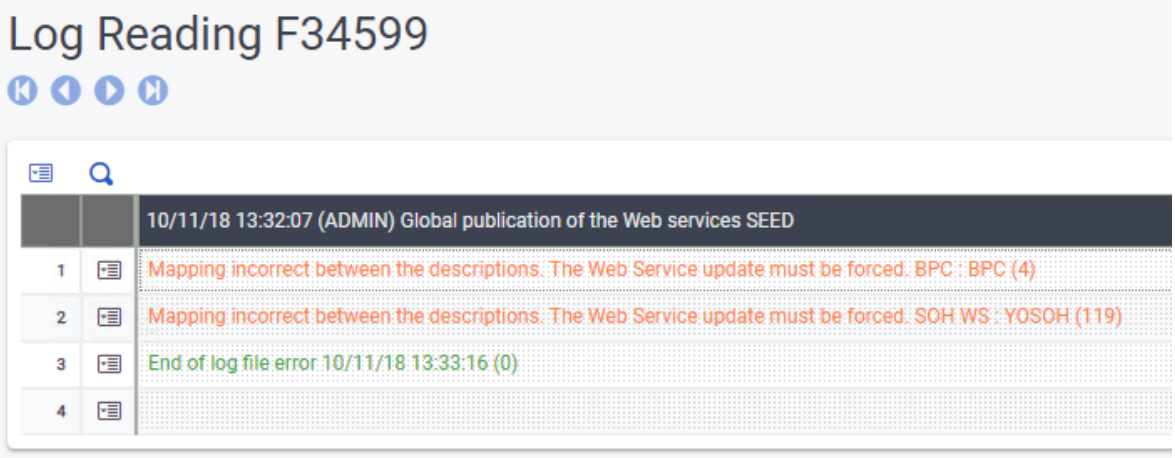
After installing the patch with the web service, you need to publish the service. This validates the web service so that it is visible.

In the application, navigate to **Development > Script dictionary > Scripts** and open **Web services** (GESAWE).

Cliquez sur **Publication globale**.

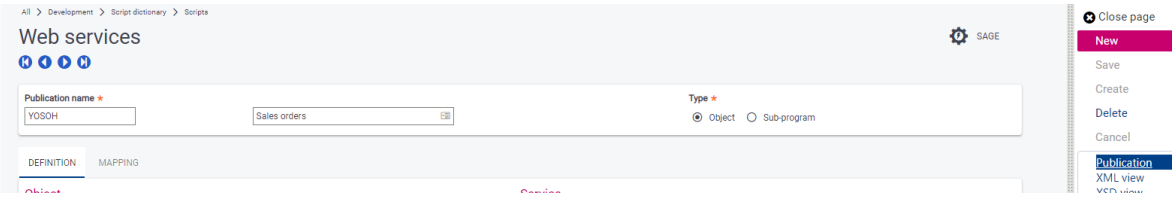


Cas des erreurs dans la trace



Seulement le web service YOSOH est utilisé.

Dans la même fonction aller sur ce web service **YOSOH** et cliquer sur le bouton **Publication.**



Use the portal

Now that the web service has been published, you can begin accessing application data in real-time via the portal.

> **Note**: WampServer needs to be running to access the portal and the application services.

Access the portal

Using the default browser that you set earlier, enter the URL for your web portal.

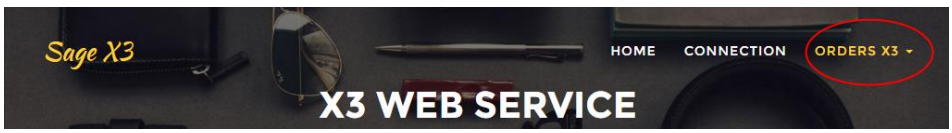
> **Note**: For this example, the URL is http://x3pu9trainvm:8125/sagex3.

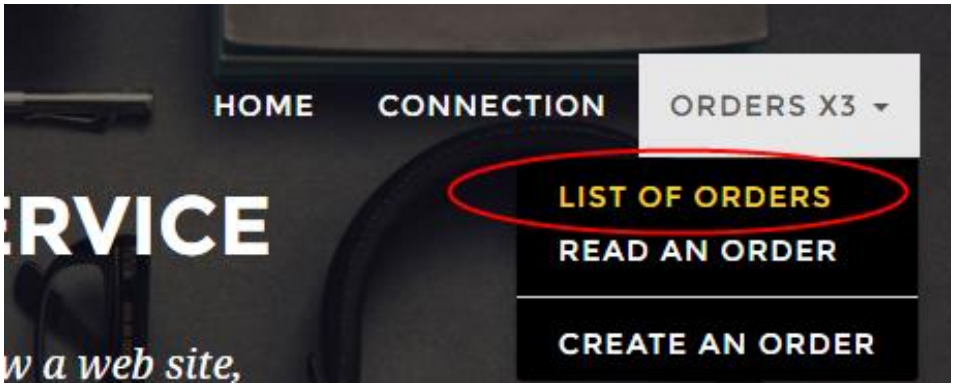
Click **CONNECTION** and log in with the username and password you set up when configured the portal.

> **Important!** You do not need to be logged into the Portal to view the orders.

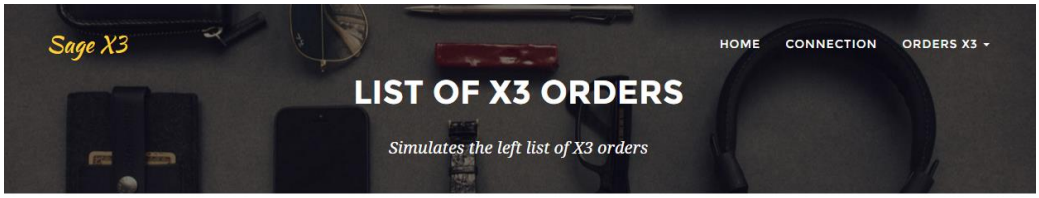
Remember, because this web service is based on YOSOH for sales information, this portal provides access to orders in Enterprise Management.

From the **ORDERS X3** pull-down menu, select **LIST OF ORDERS**.





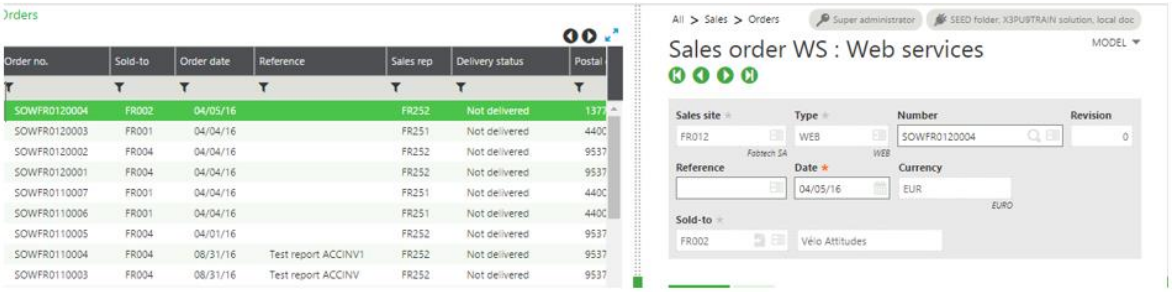
You can now see a list of current orders in your application instance.



RESULT

Order num	Client	Order date	Reference	Sales rep	Delivery status	Postal code
SOWFR0120004	FR002	05/04/2016		FR252	Not delivered	13770
SOWFR0120003	FR001	04/04/2016		FR251	Not delivered	44000
SOWFR0120002	FR004	04/04/2016		FR252	Not delivered	95370
SOWFR0120001	FR004	04/04/2016		FR252	Not delivered	95370

When you look at this data in Enterprise Management, you can see that it is the same.

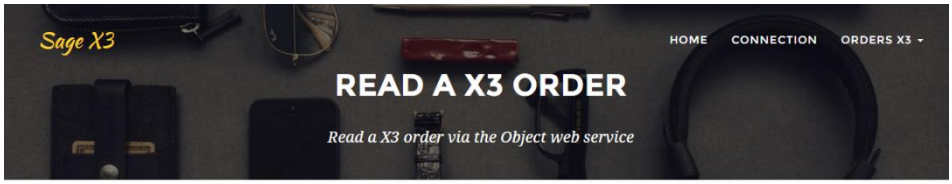


Read an order

You can read orders by selecting from the list or by selecting **READ AN ORDER** from the **ORDERS X3** menu and entering the order number. For either method, you do not need to be logged in to the portal.

Click the order number for one of the orders in the list. This example uses order SOWFR0120004.

Clicking the order number or enter the order number provides detailed information about that order.



Selection

Order number SOWFR0120004

Cancel SUBMIT

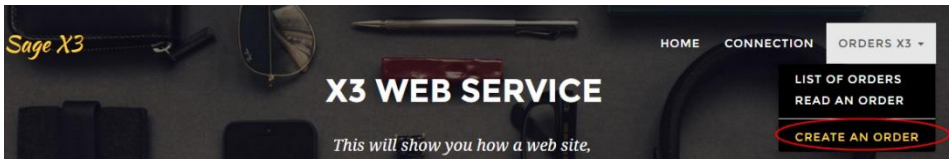
RESULT

Site	Order num	Reference	order date
FR012	SOWFR0120004		05-04-2016

Client	Name of site
FR002	Vélo Attitudes

Product	Designation	Quantity
DIS010	Laser Printer B&W 10ppm	10
DIS009	Standard 22" screen 16:10	20

You can create an order in Enterprise Management using the portal. You need to be logged in to the portal to do this.



Remember, you defined the login and password for your portal in Config.php.

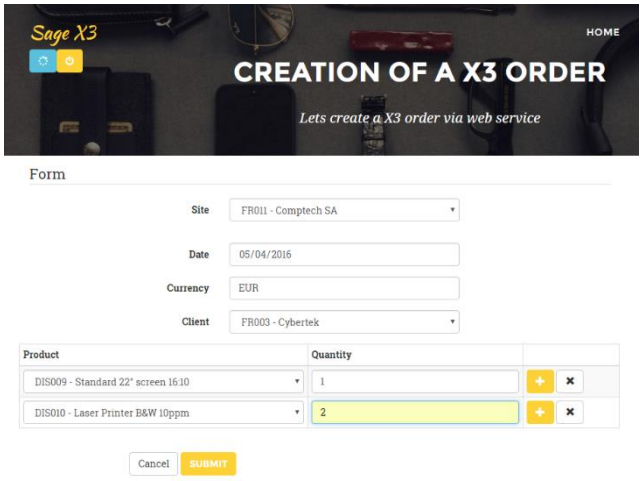
Open the file Config.php

```
need to connect to create an order; don't need to login to view orders
<?php

class Config {
   …

   public static $WEB_SITE_LOGIN    = "websage";
   public static $WEB_SITE_PASSWD   = "websage";

}
?>
```

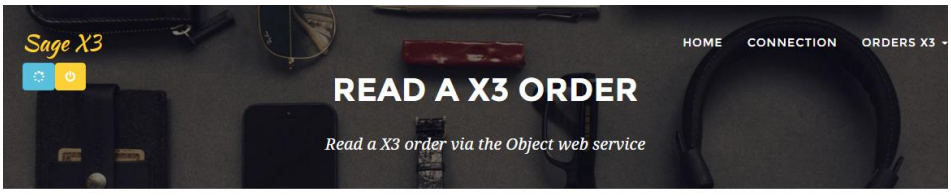
Note: To create a new order, you need to be logged into the Portal.

From the **ORDERS X3** pull-down menu, select **CREATE AN ORDER**.

Enter the relevant information as you would if you were working directly in your application and click **Submit**.

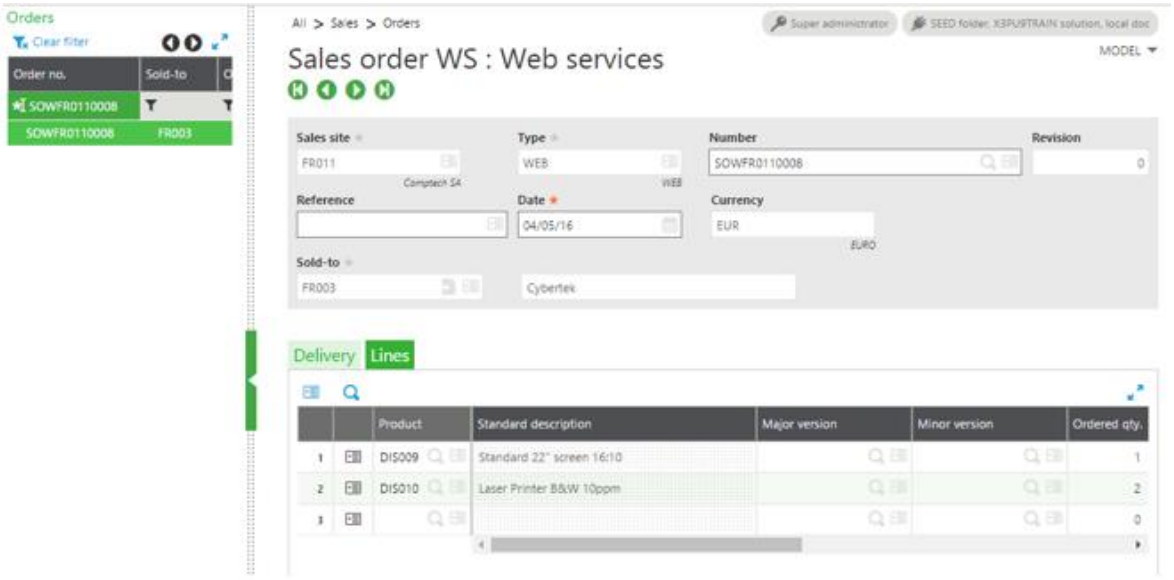


When the order has been created, click the oder number to view details.



RESULT

In the WS entry transaction, you can see the same order:

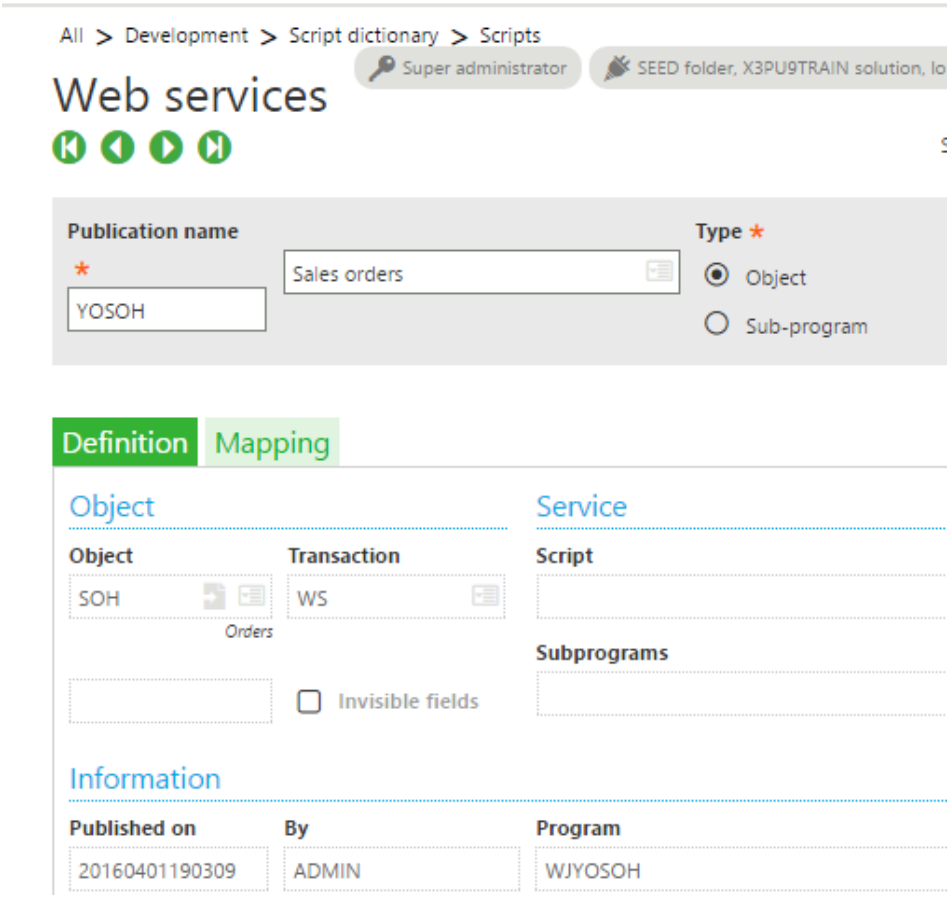


For developers

This section provides details specifically addressed to developers who have an advanced knowledge of coding and web services. The YOSOH web service will still be used as an example.

This section describes how to initiate calls without using an external application, but using the Enterprise Management test tool. You can also see the PHP or C# codes used to call the same web services.

This web service is defined as an object with the WS optimized transaction.



List the orders

In the PHP code:

Remember, the name of the Order web service is SOH.

Config::$WS_ORDER → YSOH

In /sagex3/page_soh_list.php

```
<?php
require_once ('WebService/models/Order.php');
try {
   $order = new Order ();
   echo ($order->showListe ());
} catch ( SoapFault $e ) {
ToolsWS::printError ( "Web service not available" );
}
?>
```

In /sagex3/WebService/models/Order.php

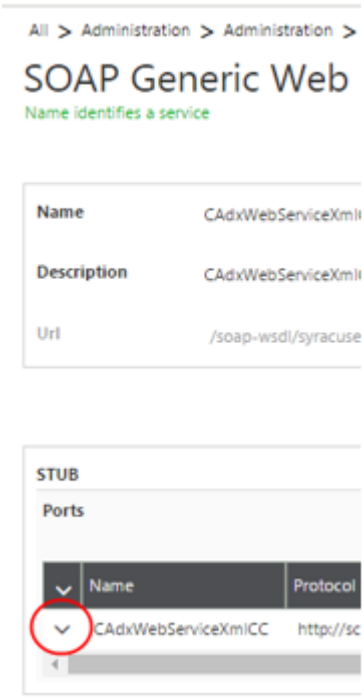
```
function showListe() {
          $WS = "*";
   $this->CAdxResultXml = $this->query ( Config::$WS_ORDER, $WS,100);
          …
   }
```

In the application tool:

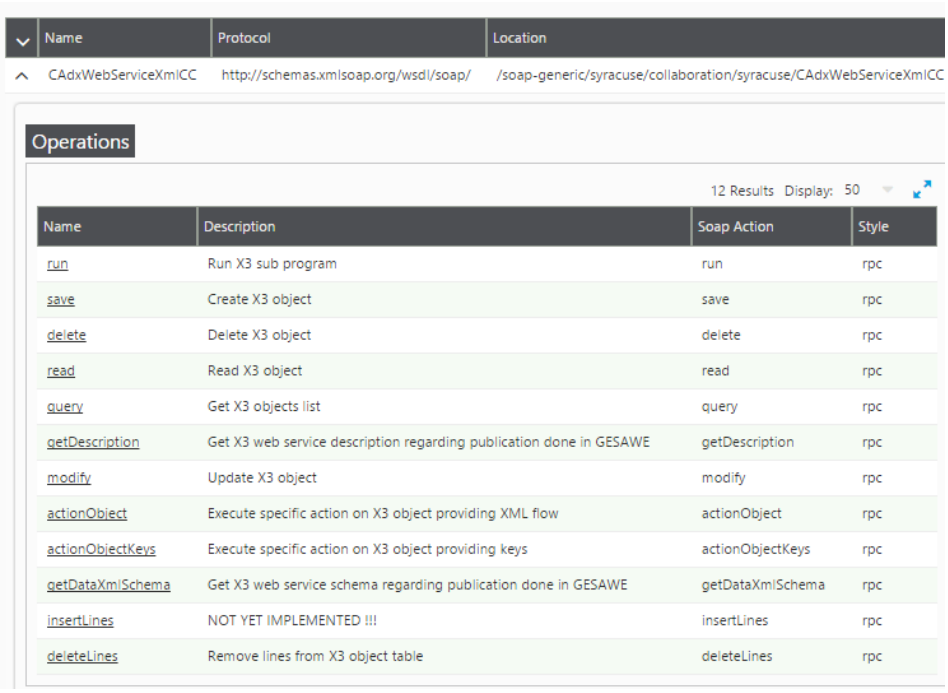
Navigate to **Administration > Administration > Web Services** and select **Classic SOAP Web Services**.

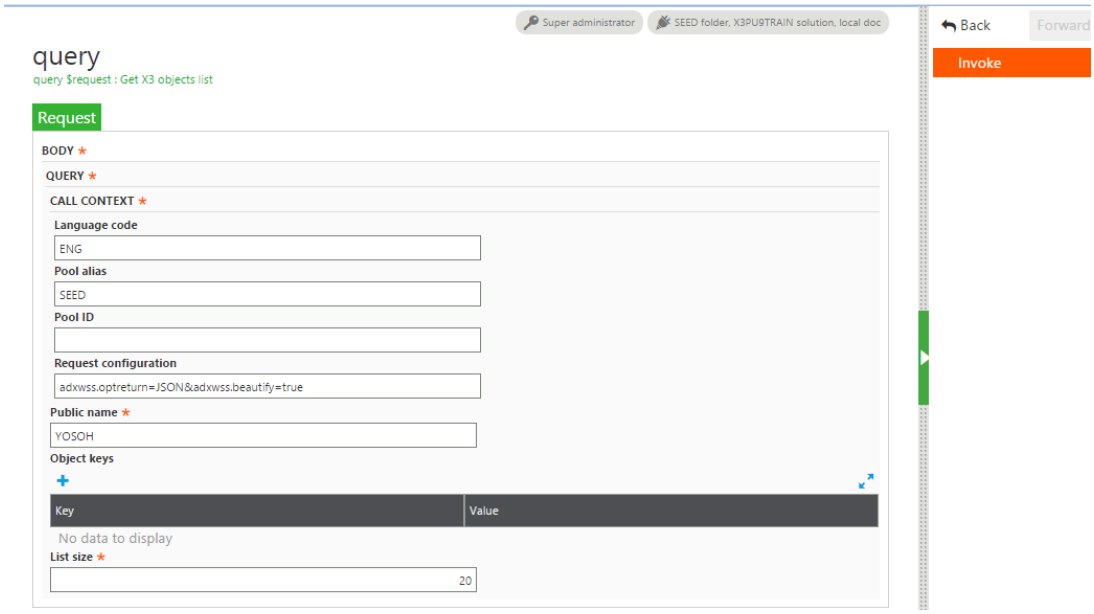
From the list of SOAP Generic Web Services, select this web service.

On the next screen, click the down arrow to see the list of Operations.



From the list of Operations, click **query**.





The request configuration

```
adxwss.optreturn=JSON&adxwss.beautify=true
```

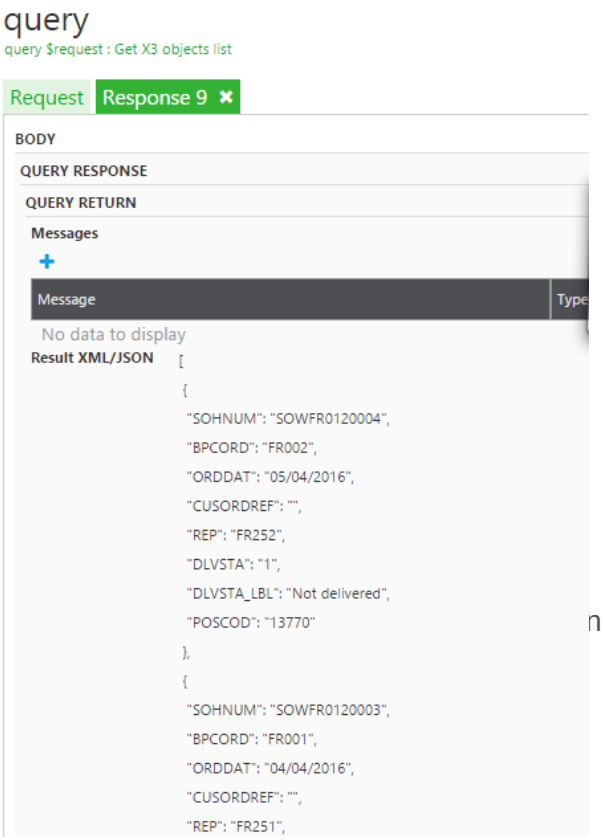
means

```
adxwss.optreturn=JSON
```

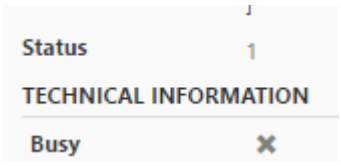
The output data format is JSON or XML, where

```
adxwss.beautify=true
```

This action improves the presentation as shown below.



n the status is 1.



- 0= ERROR

Read an order

In the PHP code:

In /sagex3/page_soh_read.php

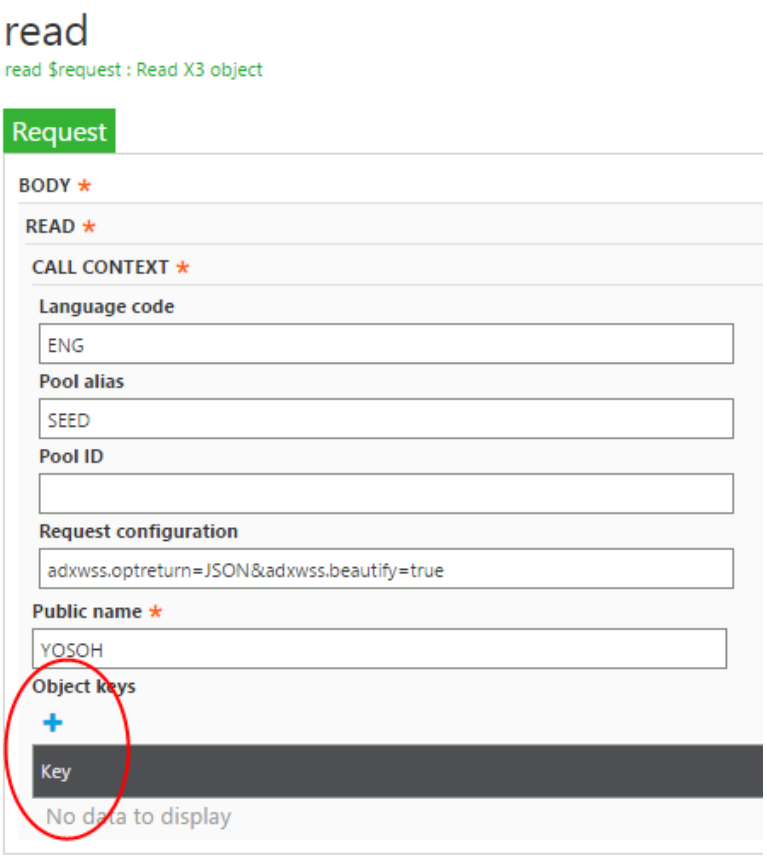
```
<?php

                                                                …
                                                                    echo
($order->showOne ( $sohnum ));
                                                                …

                                                                ?>
```

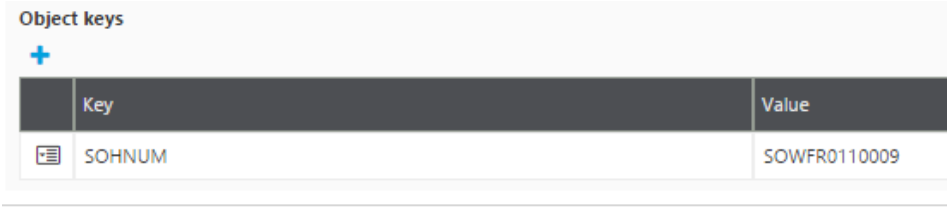
In /sagex3/WebService/models/Order.php

```
function showOne($crit) {
          …
          $cle = new CAdxParamKeyValue ();
          $cle->key = "SOHNUM";
          $cle->value = $crit;

          $this->CAdxResultXml = $this->read
(Config::$WS_ORDER,Array($cle));
          …
   }
```

In the application tool:

You must call the Read operation with the key of the order.





After selecting **Invoke**

Create an order while logged in

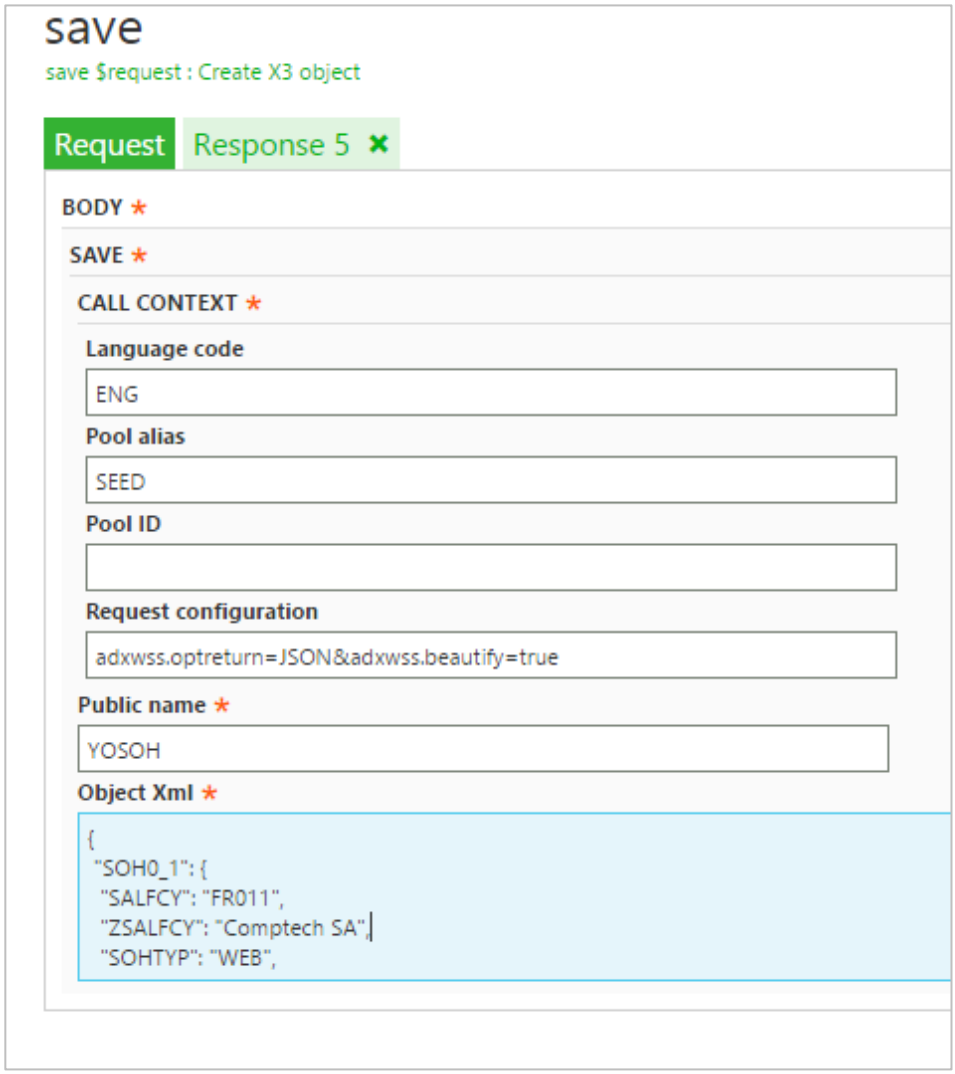
Using the application tool:

At first you can copy the result of the JSON data from the read:

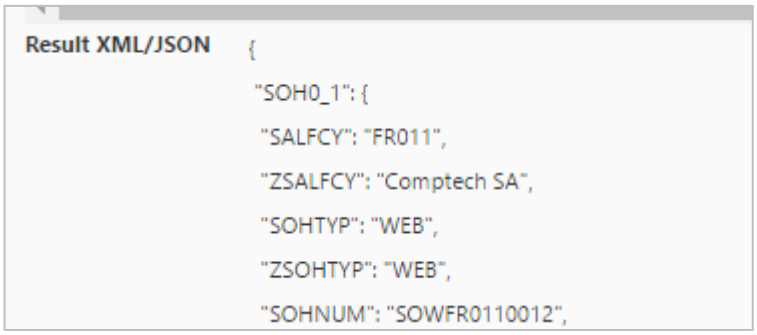
```
{
 "SOH0_1": {
  "SALFCY": "FR011",
  "ZSALFCY": "Comptech SA",
  "SOHTYP": "WEB",
  "ZSOHTYP": "WEB",
  "SOHNUM": " SOWFR0110009 ",
  "REVNUM": "0",
  "CUSORDREF": "",
  "ORDDAT": "20160406",
  "CUR": "EUR",
…
```


Replace the line: "SOHNUM": " SOWFR0110009 ", with "SOHNUM": " ",

In the tool, enter this data into the **Object Xml** field.



Invoke



The code for the order that was created is in the JSON result:
The status have the value 1.

PHP code:

In /sagex3/page_soh_create_action.php

```
<?php
…
try {
          $order = new Order ();
          echo ($order->create ( $WS ));
   } catch ( SoapFault $e ) {
          ToolsWS::printError ( "web service not available" );
   }
…
?>
```

In /sagex3/WebService/models/Order.php

```
function create($WS) {
          $this->CAdxResultXml = $this->save ( Config::$WS_ORDER, $WS );
          …
   }
```

www.sage.com