Window Azure Address:

PHP Office 365 Integration Authentication with Azure

<http://azure.microsoft.com/en-us/documentation/articles/active-directory-php-web-single-sign-on/>

# Step 1: Windows PowerShell cmdlets for Office 365 management and deployment

<https://technet.microsoft.com/en-ca/library/hh974317.aspx>

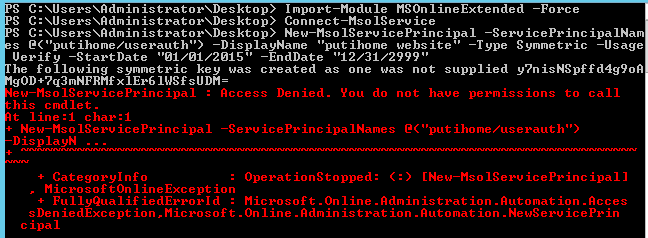
Step 2: Windows Azure Active Directory Module for Windows PowerShell

Run this PowerShell:

Import-Module MSOnlineExtended –Force

Connect-MsolService

New-MsolServicePrincipal -ServicePrincipalNames @("putihome/localhost") -DisplayName "putihome website" -Type Symmetric -Usage Verify -StartDate "01/01/2015" -EndDate "12/31/2999"



# Web Single Sign-On with PHP and Azure Active Directory

Updated:11/20/2014

PHP, Active Directory

By [tfitzmac](https://github.com/tfitzmac) and 7 contributors

Contribute to [this article](https://github.com/Azure/azure-content/blob/master/articles/active-directory-php-web-single-sign-on.md) on GitHub

## Introduction

This tutorial will show PHP developers how to leverage Azure Active Directory to enable single sign-on for users of Office 365 customers. You will learn how to:

* Provision the web application in a customer's tenant
* Protect the application using WS-Federation

### Prerequisites

The following development environment prerequisites are required for this walkthrough:

* [PHP Sample Code for Azure Active Directory](https://github.com/WindowsAzure/azure-sdk-for-php-samples/tree/master/WAAD.WebSSO.PHP)
* [Eclipse PDT 3.0.x All In Ones](http://www.eclipse.org/pdt/downloads/)
* PHP 5.3.1 (via Web Platform Installer)
* Internet Information Services (IIS) 7.5 with SSL enabled
* Windows PowerShell
* [Office 365 PowerShell Commandlets](http://msdn.microsoft.com/en-us/library/azure/jj151815.aspx)

### Table of Contents

* [Introduction](http://azure.microsoft.com/en-us/documentation/articles/active-directory-php-web-single-sign-on/#introduction)
* [Step 1: Create a PHP Application](http://azure.microsoft.com/en-us/documentation/articles/active-directory-php-web-single-sign-on/#createapp)
* [Step 2: Provision the Application in a Company's Directory Tenant](http://azure.microsoft.com/en-us/documentation/articles/active-directory-php-web-single-sign-on/#provisionapp)
* [Step 3: Protect the Application Using WS-Federation for Employee Sign In](http://azure.microsoft.com/en-us/documentation/articles/active-directory-php-web-single-sign-on/#protectapp)
* [Summary](http://azure.microsoft.com/en-us/documentation/articles/active-directory-php-web-single-sign-on/#summary)

## Step 1: Create a PHP Application

This step describes how to create a simple PHP application that will represent a protected resource. Access to this resource will be granted through federated authentication managed by the company's STS, which is described later in the tutorial.

1. Open a new instance of Eclipse.
2. From the **File** menu, click **New**, then click **New PHP Project**.
3. On the **New PHP Project** dialog, name the project phpSample, then click **Finish**.
4. From the **PHP Explorer** menu on the left, right-click phpProject, click **New**, then click **PHP File**.
5. On the **New PHP File** dialog, name the file **index.php**, then click **Finish**.
6. Replace the generated markup with the following, then save **index.php**:
7. <!DOCTYPE>
8. <html>
9. <head>
10. <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
11. <title>Index</title>
12. </head>
13. <body>
14. ##Index Page
15. </body>

</html>

1. Open **Internet Information Services (IIS) Manager** by typing inetmgr at the Run prompt and pressing Enter.
2. In IIS Manager, expand the **Sites** folder in the left menu, right-click **Default Website**, then click **Add Application...**.
3. On the **Add Application** dialog, set the **Alias** value to phpSample and the **Physical path** to the file path where you created the PHP project.
4. In Eclipse, from the **Run** menu, click **Run**.
5. On the **Run PHP Web Application** menu, click **OK**.
6. The **index.php** page will open in a new tab in Eclipse. The page should simply display the text: Index Page.

## Step 2: Provision the Application in a Company's Directory Tenant

This step describes how an administrator of an Azure Active Directory customer provisions the PHP application in their tenant and configures single sign-on. After this step is accomplished, the company's employees can authenticate to the web application using their Office 365 accounts.

The provisioning process begins by creating a new Service Principal for the application. Service Principals are used by Azure Active Directory to register and authenticate applications to the directory.

1. Download and install the Office 365 PowerShell Commandlets if you haven't done so already.
2. From the **Start** menu, run the **Windows Azure Active Directory Module for Windows PowerShell** console. This console provides a command-line environment for configuring attributes about your Office 365 tenant, such as creating and modifying Service Principals.
3. To import the required **MSOnlineExtended** module, type the following command and press Enter:

Import-Module MSOnlineExtended -Force

1. To connect to your Office 365 directory, you will need to provide the company's administrator credentials. Type the following command and press Enter, then enter your credential's at the prompt:

Connect-MsolService

1. Now you will create a new Service Principal for the application. Type the following command and press Enter:

New-MsolServicePrincipal -ServicePrincipalNames @("phpSample/localhost") -DisplayName "Federation Sample Website" -Type Symmetric -Usage Verify -StartDate "12/01/2012" -EndDate "12/01/2013"

This step will output information similar to the following:

The following symmetric key was created as one was not supplied qY+Drf20Zz+A4t2w e3PebCopoCugO76My+JMVsqNBFc=

DisplayName : Federation Sample PHP Website

ServicePrincipalNames : {phpSample/localhost}

ObjectId : 59cab09a-3f5d-4e86-999c-2e69f682d90d

AppPrincipalId : 7829c758-2bef-43df-a685-717089474505

TrustedForDelegation : False

AccountEnabled : True

KeyType : Symmetric

KeyId : f1735cbe-aa46-421b-8a1c-03b8f9bb3565

StartDate : 12/01/2012 08:00:00 a.m.

EndDate : 12/01/2013 08:00:00 a.m.

Usage : Verify

##### NOTE:

You should save this output, especially the generated symmetric key. This key is only revealed to you during Service Principal creation, and you will be unable to retrieve it in the future. The other values are required for using the Graph API to read and write information in the directory.

1. The final step sets the reply URL for your application. The reply URL is where responses are sent following authentication attempts. Type the following commands and press enter:
2. $replyUrl = New-MsolServicePrincipalAddresses –Address "https://localhost/phpSample"

Set-MsolServicePrincipal –AppPrincipalId "7829c758-2bef-43df-a685-717089474505" –Addresses $replyUrl

The web application has now been provisioned in the directory and it can be used for web single sign-on by company employees.

## Step 3: Protect the Application Using WS-Federation for Employee Sign In

This step shows you how to add support for federated login using Windows Identity Foundation (WIF) and the simpleSAML.php libraries you downloaded with the sample code in the prerequisites. You will also add a login page and configure trust between the application and the directory tenant.

1. In Eclipse, right-click the **phpSample** project, click **New**, then click **File**.
2. On the **New File** dialog, name the file **federation.ini**, then click **Finish**.
3. In the new **federation.ini** file, enter the following information, supplying the values with the information you saved in Step 2 when creating your Service Principal:
4. federation.trustedissuers.issuer=https://accounts.accesscontrol.windows.net/v2/wsfederation
5. federation.trustedissuers.thumbprint=qY+Drf20Zz+A4t2we3PebCopoCugO76My+JMVsqNBFc=
6. federation.trustedissuers.friendlyname=Fabrikam
7. federation.audienceuris=spn:7829c758-2bef-43df-a685-717089474505
8. federation.realm=spn:7829c758-2bef-43df-a685-717089474505

federation.reply=https://localhost/phpSample/index.php

##### NOTE:

The **audienceuris** and **realm** values must be prefaced by "spn:".

1. In Eclipse, right-click the **phpSample** project, click **New**, then click **PHP File**.
2. On the **New PHP File** dialog, name the file **secureResource.php**, then click **Finish**.
3. In the new **secureResource.php** file, enter the following code, replacing the **c:\phpLibraries** path with the root location where you downloaded the sample code. The root location should include the **simpleSAML.php** file and **federation** folder:
4. <?php
5. ini\_set('include\_path', ini\_get('include\_path').';c:\phpLibraries\;');
6. require\_once ('federation/FederatedLoginManager.php');
7. session\_start();
8. $token = $\_POST['wresult'];
9. $loginManager = new FederatedLoginManager();
10. if (!$loginManager->isAuthenticated()) {
11. if (isset ($token)) {
12. try {
13. $loginManager->authenticate($token);
14. } catch (Exception $e) {
15. print\_r($e->getMessage());
16. }
17. } else {
18. $returnUrl = "https://" . $\_SERVER['HTTP\_HOST'] . $\_SERVER['PHP\_SELF'];
19. header('Pragma: no-cache');
20. header('Cache-Control: no-cache, must-revalidate');
21. header("Location: " . FederatedLoginManager :: getFederatedLoginUrl($returnUrl), true, 302);
22. exit();
23. }
24. }

?>

1. Open the **index.php** page and update its contents to secure the page, then save it:
2. <?php
3. require\_once (dirname(\_\_FILE\_\_) . '/secureResource.php');
4. ?>
5. <!DOCTYPE html>
6. <html>
7. <head>
8. <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
9. <title>Index Page</title>
10. </head>
11. <body>
12. <h2>Index Page</h2>
13. <h3>Welcome <strong><?php print\_r($loginManager->getPrincipal()->getName()); ?></strong>!</h3>
14. <h4>Claim list:</h4>
15. <ul>
16. <?php
17. foreach ($loginManager->getClaims() as $claim) {
18. print\_r('<li>' . $claim->toString() . '</li>');
19. }
20. ?>
21. </ul>
22. </body>

</html>

1. From the **Run** menu, click **Run**. You should automatically be redirected to the Office 365 Identity Provider page, where you can log in using your directory tenant credentials. For example, john.doe@fabrikam.onmicrosoft.com.

## Summary

This tutorial has shown you how to create and configure a single tenant PHP application that uses the single sign-on capabilities of Azure Active Directory.

A sample that shows how to use Azure Active Directory and single sign-on for PHP websites is available at <https://github.com/WindowsAzure/azure-sdk-for-php-samples/tree/master/WAAD.WebSSO.PHP>.