SMS Sign-in PowerShell Wrapper Documentation

Last updated: 3/13/2020

We have created a PowerShell script that calls the SMS Sign-in API. And will allow you to do the following operations:

* Set an SMS Sign-in number for a user
* Delete an SMS Sign-in number for a user
* Check the SMS Sign-in status for a user
* Bulk add SMS Sign-in numbers for users

In order to use this script, you’ll need to:

1. Register an Azure AD application with permissions to call the API
2. Edit the PowerShell script to call your app
3. Run the PowerShell script

# Register an Azure AD application which has permissions to call the API

The Graph authorization model requires that an application must be consented by a user or administrator prior to accessing an organization’s data. Creating this app will provide the appropriate read/write permissions for the PowerShell code to update your tenant (with appropriately authorized credentials).

1. Sign in to [portal.azure.com](https://microsoft.sharepoint.com/teams/aad/Shared%20Documents/CAT/Phone%20number%20as%20username/Documentation/portal.azure.com) as a global administrator.
2. Navigate to the Azure AD extension, and click on **App registrations** in the **MANAGE** section.
3. Click on **New registration** button at the top of the page.

Provide a name for the application that is different from any other application in your tenant’s directory (e.g., “SMS Sign-in API”), leave the supported account type as “Accounts in this organizational directory only”. Select Public client for the Redirect URI and enter *urn:ietf:wg:oauth:2.0:oob.*

1. Click **register.**
2. When the application is registered, copy the **Application (client) ID**, and save the value for later – we will use it in the script.

A screenshot of a cell phone

Description automatically generated

1. Click on Settings, then click on **Required permissions**.
2. Click on **API permissions** under **Manage**.

A screenshot of a cell phone

Description automatically generated

1. Click on **Add a permission**. Click on **Microsoft Graph**, **Delegated Permissions**. Select the following permissions:
   1. Directory.AccessAsUser.All
   2. Directory.ReadWrite.All
2. Click on **Add a permission**. Click on **Azure Active Directory Graph, Delegated Permissions** and select **Directory.ReadWriteAll.** Click **add permissions.**
3. At the top of the API Permission page, click **Grant admin consent for <your tenant name>**, and select **yes.**

# Install PowerShell Prerequisites

1. The script requires MSAL.PS, which can install by running: *Install-Module -Name MSAL.PS*

# Get the script ready

1. Change the file name of *sign\_in\_wrapper.txt* to *sign\_in\_wrapper.ps1*.

# Run the PowerShell Script

1. Open PowerShell and navigate to the folder where you saved the *sign\_in\_wrapper.ps1* file.

Load the script by running *. .\sign\_in\_wrapper.ps1 -tenant <your tenant id> -clientId <your client id (from app registration)> -isPPE $false* and signing in with Global Admin, Authentication Admin, or Privileged Authentication Admin credentials.

1. You can now use the script, which contains four different modules:
   1. ***Get-UserSMSSignInSettings -userID <desired user’s ID>*** returns whether or not this user is enabled for SMS Sign-in. Returns one of the following states:
      1. **Ready--** The user is successfully enabled to use their phone number for SMS Sign-in.
      2. **notExists--** The user does not have a phone number registered for SMS Sign-in.
   2. ***Set-UserSMSSignInNumber -phoneNumber <userphone number> -userID <user ID>*** sets the phone number for a user. Phone numbers will need to be formatted with as one string including the area code, ex “+15555555555”. Returns one of the following states:
      1. **notAllowedByPolicy—**This user is not enabled for the SMS Sign-in policy.
      2. **phoneNumberNotUnique-**- Sign In provisioning couldn't complete because the phone number is not unique in the tenant. In order to successfully set the user, you’ll either need to remove the number from the colliding account, or set a new, unique number for the user.
      3. **ready-**- The method is properly provisioned to use SMS sign in and the user is allowed by policy.
   3. ***Remove-SMSSignInNumber -userID <user ID>*** removes the SMS Sign-in number for a user. Returns one of the following states:
      1. **success-**- The SMS Sign-in number was successfully removed.
      2. **notExists—** No number was removed, as no number existed. Note that this will be returned if there is a phone number on the account that is not enabled for SMS Sign-in. In this case, the number (ex. MFA number) will not be removed.
   4. ***BulkSet-SMSSignInNumber -filePath <file path>*** bulk sets phone numbers from a properly formatted CSV (template provided in bulkadd.csv). Will return the same values as the set method, but will return an internal server error if phone number is improperly formatted, or User ID doesn’t exist in the directory.
      1. ***Column A: User ID***
      2. ***Column B: Phone number***