

List table content with PowerShell via OAuth 2.0 authentication

Requirements: Permissions to create an App Registration and PowerShell Modules "AzTable" & "Az.Storage".

This tutorial describes how to use content from an Azure Storage Table in a PowerShell script. The authentication against the Azure Storage API is unattended and credentials are handled with an App Registration.

Create App Registration

Create a new App Registration and get the three variables as described in this guide: [Get app details and gr... | LNC DOCS \(lucanoahcaprez.ch\)](#)

Add API permission

The only required permission for this App Registration is "user_impersonation". This permission can be found under the Azure Service Management API.

Request API permissions



< All APIs

 Azure Service Management
<https://management.azure.com/> [Docs](#) 

What type of permissions does your application require?

Delegated permissions

Your application needs to access the API as the signed-in user.



Application permissions

Your application runs as a background service or daemon without a signed-in user.

Select permissions

[expand all](#)

 Start typing a permission to filter these results

 The "Admin consent required" column shows the default value for an organization. However, user consent can be customized per permission, user, or app. This column may not reflect the value in your organization, or in organizations where this app will be used. [Learn more](#) 

Permission	Admin consent required
▼ Permissions (1)	
<input checked="" type="checkbox"/> user_impersonation ⓘ Access Azure Service Management as organization users	No

Grant permissions to Azure Storage Account

After you got all variables (Tenant ID, Client ID & Client Secret) you can add the permissions for Azure RBAC to the created App Registration. You need to go to the corresponding storage account within the azure portal. There you have to add the "Storage Account Contributor" role under "Access Control IAM":

[Role](#) [Members](#) [Review + assign](#)

Selected role Storage Account Contributor

Assign access to
☒ User, group, or service principal
☐ Managed identity

Members [+ Select members](#)

Name	Object ID	Type	
AAP-FUNC-ALL-PERMISSION-GetStorag...	0f9b3930-fda9-48e1-b2d9-a2d511fd6487	App	
Description <input type="text" value="Optional"/>			

Unfortunately, the "Storage Account Contributor" role is mandatory and restricting to a Storage Table Reader for example is not possible, otherwise the data cannot be read.

PowerShell Code

The following code can be used to read the data from the storage table specified. Here it is important that the variables are filled in correctly and that the PowerShell modules are accessible.

```
$TenantID = "<tenantid>"
$ClientId = "<cliendid>"
$ClientSecret = "<clientsecret>"

$SubscriptionId = "<subscriptionid>"
$resourceGroupName = "<resourceGroupName>"
$storageAccName = "<storageaccountname>"
$tableName = "<tablename>"

Import-Module -Name Az.Storage
Import-Module -Name AzTable

$Password = ConvertTo-SecureString -AsPlainText $ClientSecret -Force
$Credential = New-Object System.Management.Automation.PSCredential ($ClientId, $Password)
$ctx = Connect-AzAccount -ServicePrincipal -Credential $Credential -Tenant $TenantId -Subscription
$SubscriptionId
$ctx=(Get-AzStorageAccount -ResourceGroupName $resourceGroupName -Name $storageAccName).Context

$cloudTable = (Get-AzStorageTable -Name $tableName -Context $ctx.context).CloudTable
$TableContent = Get-AzTableRow -table $cloudTable
```

Revision #5

Created 22 May 2023 09:23:22

Updated 21 July 2024 15:11:16