Write table content with PowerShell via OAuth 2.0 authentication

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

Preparations

To gain access to the storage table you need to initialize an App Registration and add permission & a client secret. Then you have to add the Azure RBAC role to the storage account. Everything is documented here in detail: List table content wit... | LNC DOCS (lucanoahcaprez.ch)

PowerShell Code

The following code can be used to write data to 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

Add-AzTableRow -partitionKey "dn" -Rowkey "$DomainName" -table $cloudTable -property @{
"<property1>" = "value1"; "<property2>" = "value2"; "<property3>" = "value3"; } | Out-Null
```

Revision #6
Created 2023-05-22 09:25:26 UTC
Updated 2026-08-23 16:20:49 UTC by Luca Noah Caprez