Download

# Download via git

git clone https://github.com/JanTkacSk/PSZabbix6.0.git

# Download manually

<https://github.com/JanTkacSk/PSZabbix6.0.git> > code > Download.zip

Import

# Import to the current session only

The module will have to be re-imported every time you open powershell. Adjust the path based on the location of your package.

Import-Module C:\Users\YourUser\PSZabbix6.0\PSZabbix6.0.psm1

# Import on PowerShell startup

### Edit powershell profile via a scipt

Open PowerShell or PowerShell\_ise or a PowerShell terminal in VSCode and run the following command. Adjust the path based on the location of your package.

Import-Module C:\Users\YourUser\PSZabbix6.0\ZX.UpdatePSProfile.ps1

### Edit powershell profile manually

Open PowerShell or PowerShell\_ise or Terminal in VSCode

Open the profile script by running the following command. If the profile file does not exist the notepad will offer to create it for you.

notepad $Profile

Add the following line into the file and save it. Adjust the path based on the location of your package.

Import-Module C:\Users\YourUser\PSZabbix6.0\PSZabbix6.0.psm1

New-ZXTokenSession

New-ZXTokenSession -Load -Url https://YourZabbixAPIURL

Get-ZXHost

# Examples

### Example 1: Search for a single host based on an exact Name match.

Return only those results that exactly match the given host name(s). There is no result for “zx\_test\_host\_50” because it does not exist. In Zabbix API, host “**name”** is called “**host**” and “**visible name”** is called “**name**”. In PowerShell module “**name**” is “**Name**” and the “**visible name**” is “**Alias**”

Get-ZXHost -Name zx\_test\_host\_1,zx\_test\_host\_2,zx\_test\_host\_50

hostid : 492558

host : zx\_test\_host\_1

name : zx\_alias\_1

status : 1

proxy\_hostid : 36710

hostid : 494040

host : zx\_test\_host\_2

name : zx\_alias\_2

status : 1

proxy\_hostid : 36710

### Example 2: Search for hosts based on a Name pattern.

Return results that match the given pattern (case-insensitive). Accepts a host name to search for. If no additional options are given, this will perform a LIKE "%…%" search.

Get-ZXHost -NameSearch zx\_test

hostid : 492558

host : zx\_test\_host\_1

name : zx\_alias\_1

status : 1

proxy\_hostid : 36710

hostid : 494040

host : zx\_test\_host\_2

name : zx\_alias\_2

status : 1

proxy\_hostid : 36710

### Example 3: Search for hosts based on an Alias pattern.

Return results that match the given pattern (case-insensitive). Accepts a host name to search for. If no additional options are given, this will perform a LIKE "%…%" search.

Get-ZXHost -AliasSearch some\_alias

### Example 4: Return an “items” property with host items.

By using -IncludeItems you will get a basic set of properties for each item.

Get-ZXHost -NameSearch vm-win -IncludeItems

hostid : 499271

host : vm-win-test-1

name : vm-win-test-1

status : 0

proxy\_hostid : 0

items : {@{itemid=53119623; name=C:: Access; type=0; delay=3m; master\_itemid=0; lastvalue=0}, @{itemid=53119533;

name=caCertificate; type=0; delay=1h; master\_itemid=0; lastvalue=}, @{itemid=53119534; name=caPendingUpdates;

type=7; delay=4h; master\_itemid=0; lastvalue=}, @{itemid=53119625; name=C:: Free space; type=7; delay=3m;

master\_itemid=0; lastvalue=0}…}

### Example 5: Return an “items” property with host items, only show name and type of each item.

Get-ZXHost -NameSearch vm-win -IncludeItems -ItemProperties name,type

### Example 6: Return an “items” property with host items, return all properties of each item.

Get-ZXHost -NameSearch vm-win -IncludeItems -ItemProperties extend