How to get all Azure DevOps organizations in your AzureAD directory

Recently I got a requirement to retrieve list of Azure DevOps organizations. I found few ways to retrieve that. Though of sharing it with you'll.

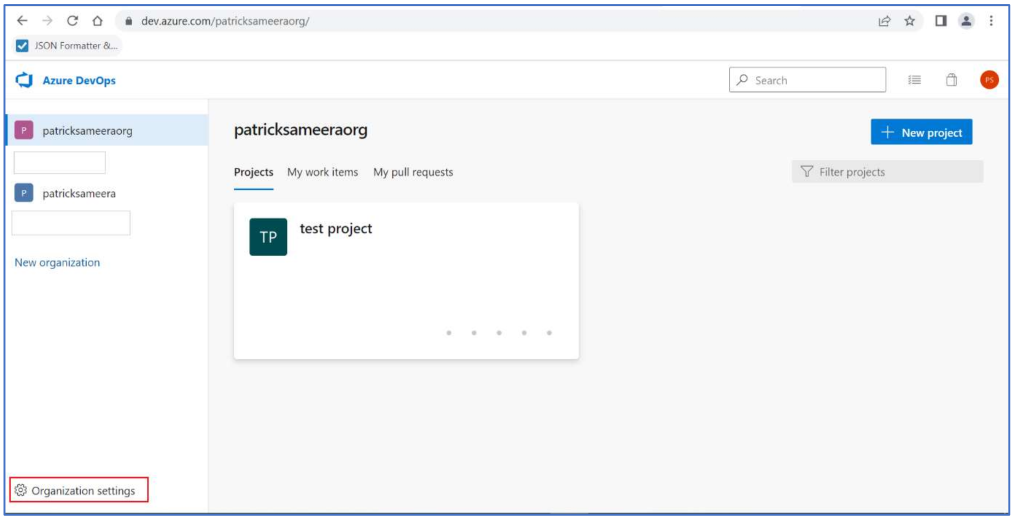
On this list, you can find the following information:

- organization IDs
- organization names
- organization URLs
- organization owners

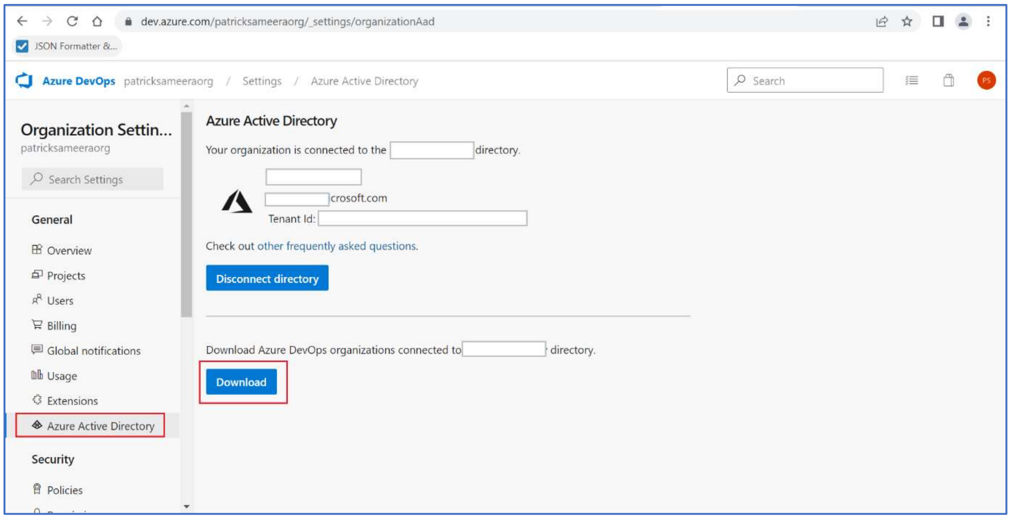
Option 01: From Azure DevOps UI

1. Sign into your organization (https://dev.azure.com/{yourorganization}).

2. Select Organization settings.



3. Select Azure Active Directory, and then Download.



Your organization is connected to the [redacted] directory.

[redacted]

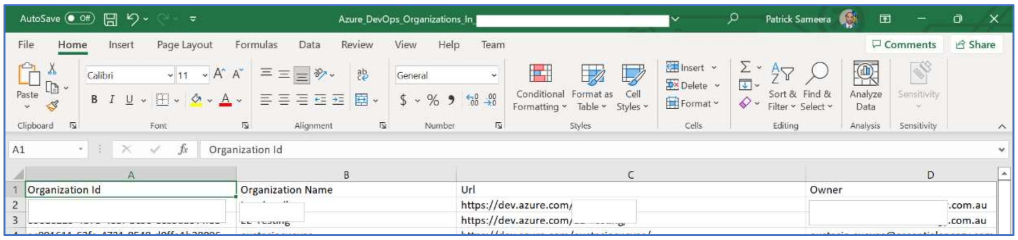
[redacted]crosoft.com

Tenant Id: [redacted]

Check out other frequently asked questions.

Download Azure DevOps organizations connected to [redacted] directory.

It will download you the list of organisations as a excel document.



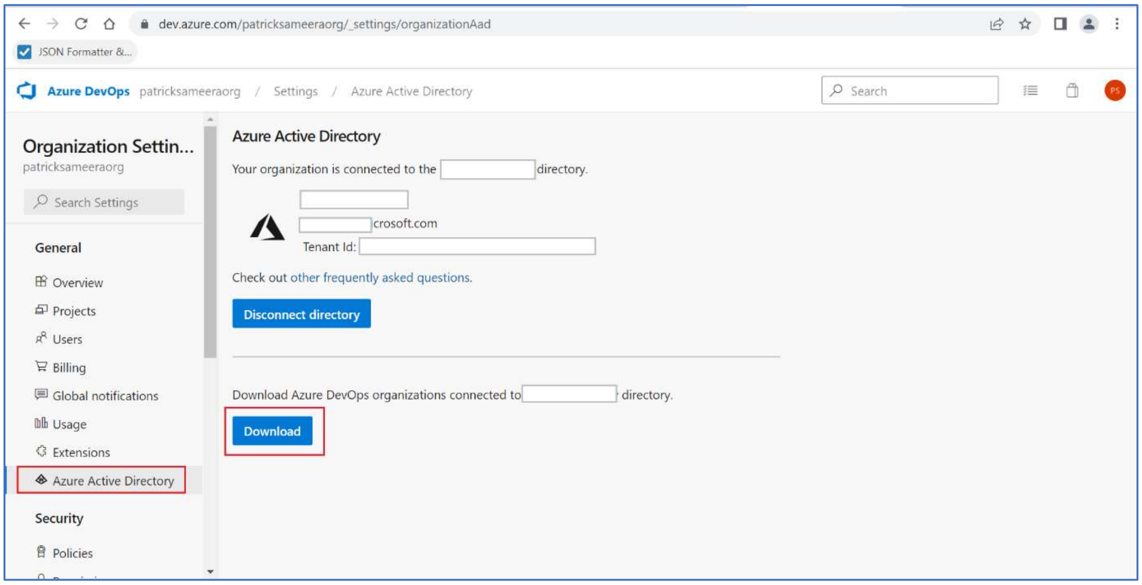
	A	B	C	D
1	Organization Id	Organization Name	Url	Owner
2	[redacted]	[redacted]	https://dev.azure.com/[redacted]	[redacted].com.au
3	[redacted]	[redacted]	https://dev.azure.com/[redacted]	[redacted].com.au

Option 02: From Azure DevOps REST API

Just copy paste below URL to the browser:

https://aexprodeau1.vsaex.visualstudio.com/_apis/EnterpriseCatalog/Organizations?tenantId={tenantID}

Mind you that you must pass the Tenant ID as a parameter, and you can get the Tenant ID by going in to Azure Directory link.



Azure Active Directory

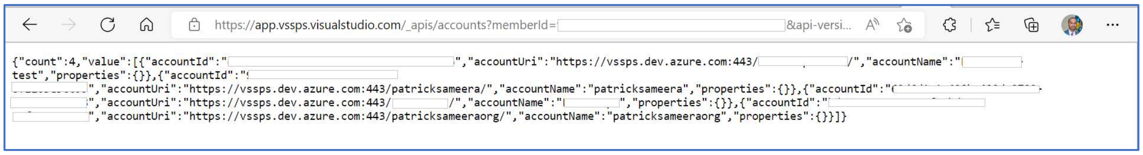
Your organization is connected to the [redacted] directory.

[redacted]
[redacted]crosoft.com
Tenant Id: [redacted]

Check out other frequently asked questions.

Download Azure DevOps organizations connected to [redacted] directory.

It will return you the data in below format:



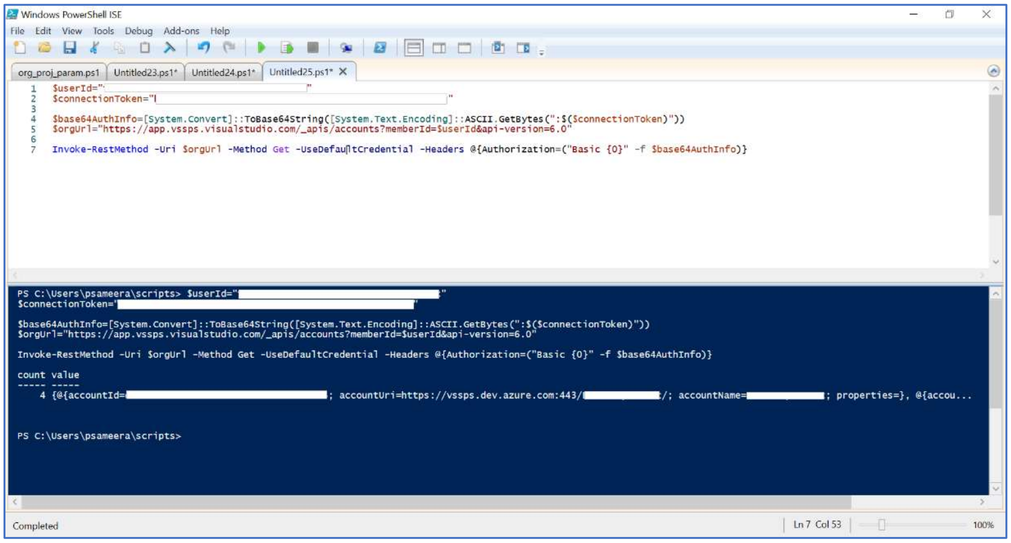
https://app.vssps.visualstudio.com/_apis/accounts?memberId=[redacted]&api-versi...

```
{"count":4,"value":[{"accountId":"[redacted]","accountUri":"https://vssps.dev.azure.com:443/[redacted]/","accountName":"[redacted]
test","properties":{}},{"accountId":"[redacted]
[redacted]","accountUri":"https://vssps.dev.azure.com:443/patricksameera/","accountName":"patricksameera","properties":{}},{"accountId":"[redacted]
[redacted]","accountUri":"https://vssps.dev.azure.com:443/[redacted]/","accountName":"[redacted]","properties":{}},{"accountId":"[redacted]
[redacted]","accountUri":"https://vssps.dev.azure.com:443/patricksameeraorg/","accountName":"patricksameeraorg","properties":{}}]}
```

Option 03: Using PowerShell

```
$userId ="{userId}"
$connectionToken="{token}"

$base64AuthInfo=[System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes(":$($connectionToken)"))
$orgUrl="https://app.vssps.visualstudio.com/_apis/accounts?memberId=$userId&api-version=6.0"

Invoke-RestMethod -Uri $orgUrl -Method Get -UseDefaultCredential -Headers @{Authorization=("Basic {0}" -f $base64AuthInfo)}
```

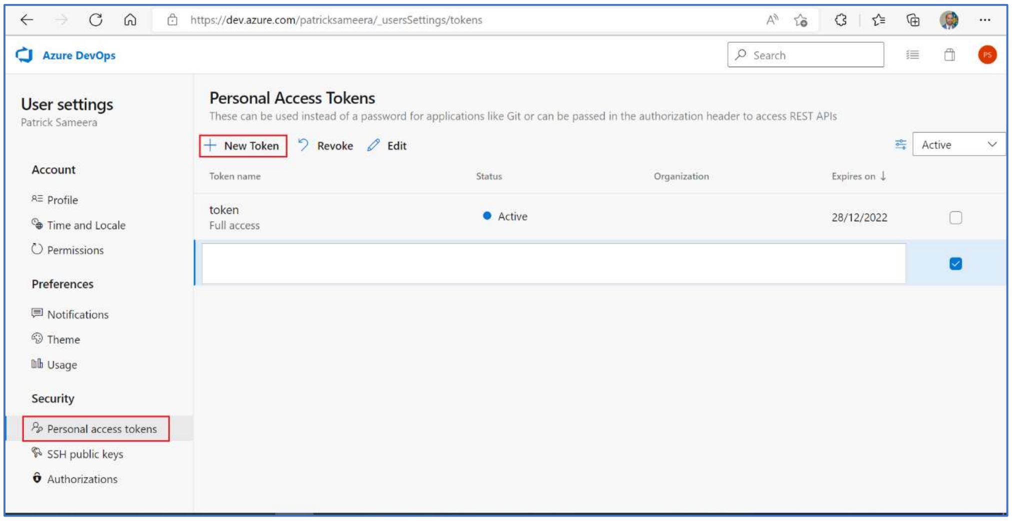
And you get below result:



```
$userId="[redacted]"
$connectionToken="[redacted]"

$base64AuthInfo=[System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes(":$($connectionToken)"))
$orgUrl="https://app.vssps.visualstudio.com/_apis/accounts?memberId=$userId&api-version=6.0"

Invoke-RestMethod -Uri $orgUrl -Method Get -UseDefaultCredential -Headers @{Authorization=("Basic {0}" -f $base64AuthInfo)}
```

```
PS C:\Users\psameera\scripts> $userId="[redacted]"
$connectionToken='[redacted]'

$base64AuthInfo=[System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes(":$($connectionToken)"))
$orgUrl="https://app.vssps.visualstudio.com/_apis/accounts?memberId=$userId&api-version=6.0"

Invoke-RestMethod -Uri $orgUrl -Method Get -UseDefaultCredential -Headers @{Authorization=("Basic {0}" -f $base64AuthInfo)}

count value
----- -----
    4 {@{accountId=[redacted]; accountUri=https://vssps.dev.azure.com:443/[redacted]/; accountName=[redacted]; properties=}, @{accou...


PS C:\Users\psameera\scripts>
```

You can create a personal access token from:



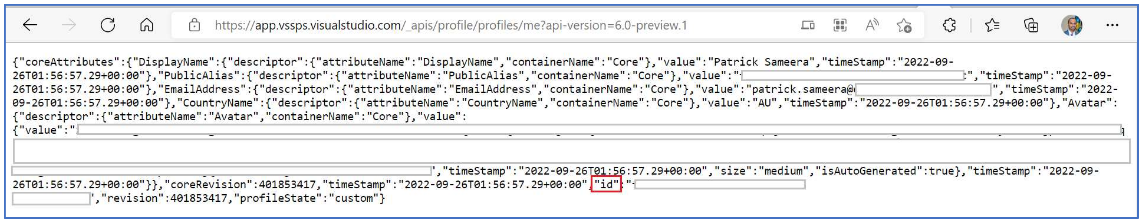
Personal Access Tokens

These can be used instead of a password for applications like Git or can be passed in the authorization header to access REST APIs

Token name	Status	Organization	Expires on ↓	
token Full access	● Active		28/12/2022	□
				☑

You can retrieve your User ID by hitting below URL and by copying the value for “id” field.

https://app.vssps.visualstudio.com/_apis/profile/profiles/me?api-version=6.0-preview.1

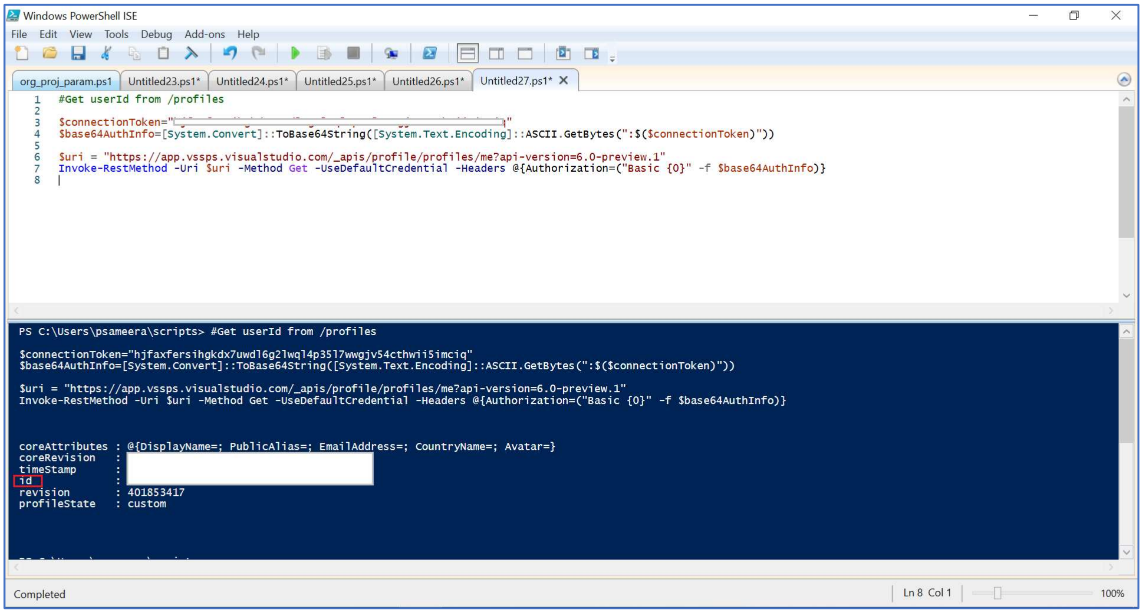


```
{"coreAttributes":{"DisplayName":{"descriptor":{"attributeName":"DisplayName","containerName":"Core"},"value":"Patrick Sameera","timeStamp":"2022-09-26T01:56:57.29+00:00"},"PublicAlias":{"descriptor":{"attributeName":"PublicAlias","containerName":"Core"},"value":"[redacted]","timeStamp":"2022-09-26T01:56:57.29+00:00"},"EmailAddress":{"descriptor":{"attributeName":"EmailAddress","containerName":"Core"},"value":"patrick.sameera@[redacted]","timeStamp":"2022-09-26T01:56:57.29+00:00"},"CountryName":{"descriptor":{"attributeName":"CountryName","containerName":"Core"},"value":"AU","timeStamp":"2022-09-26T01:56:57.29+00:00"},"Avatar":{"descriptor":{"attributeName":"Avatar","containerName":"Core"},"value":
{"value":"[redacted]","timeStamp":"2022-09-26T01:56:57.29+00:00","size":"medium","isAutoGenerated":true},"timeStamp":"2022-09-26T01:56:57.29+00:00"}},"coreRevision":401853417,"timeStamp":"2022-09-26T01:56:57.29+00:00","id":"[redacted]","revision":401853417,"profileState":"custom"}
```

Also, you can retrieve your User ID by below PowerShell script and by copying the value for “id” field.

```
$connectionToken="{token}"
$base64AuthInfo=[System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes(":$($connectionToken)"))

$uri = "https://app.vssps.visualstudio.com/_apis/profile/profiles/me?api-version=6.0-preview.1"
Invoke-RestMethod -Uri $uri -Method Get -UseDefaultCredential -Headers @{Authorization=("Basic {0}" -f $base64AuthInfo)}
```

And you get below result:



```
#Get userId from /profiles

$connectionToken="[redacted]"
$base64AuthInfo=[System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes(":$($connectionToken)"))

$uri = "https://app.vssps.visualstudio.com/_apis/profile/profiles/me?api-version=6.0-preview.1"
Invoke-RestMethod -Uri $uri -Method Get -UseDefaultCredential -Headers @{Authorization=("Basic {0}" -f $base64AuthInfo)}

```

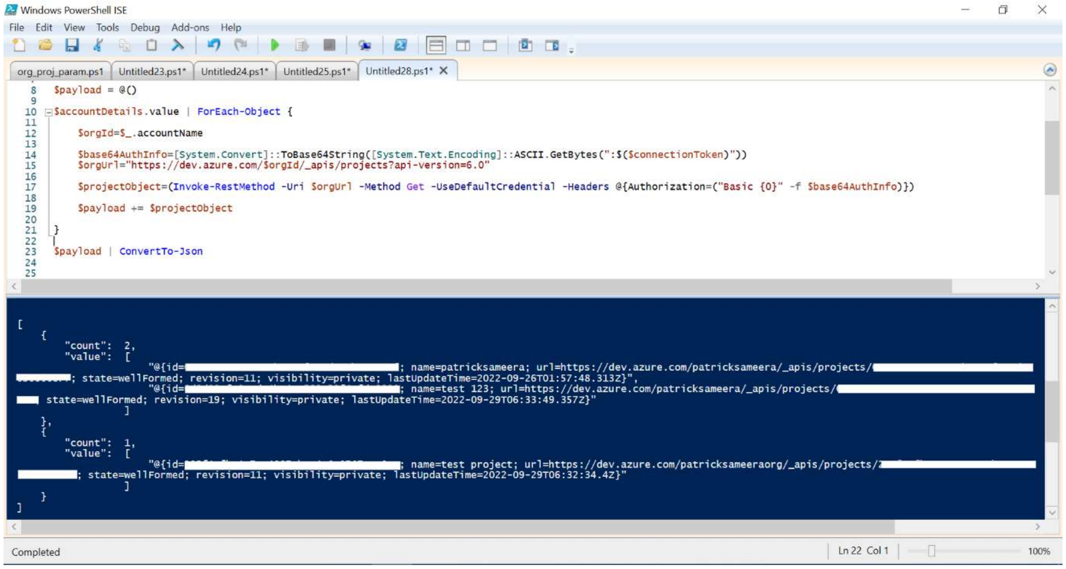
```
PS C:\Users\psameera\scripts> #Get userId from /profiles

$connectionToken="hjfaxfersihgkdx7uwdl6g2lwql4p35l7wwgjv54cthwii5imciq"
$base64AuthInfo=[System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes(":$($connectionToken)"))

$uri = "https://app.vssps.visualstudio.com/_apis/profile/profiles/me?api-version=6.0-preview.1"
Invoke-RestMethod -Uri $uri -Method Get -UseDefaultCredential -Headers @{Authorization=("Basic {0}" -f $base64AuthInfo)}


coreAttributes : @{DisplayName=; PublicAlias=; EmailAddress=; CountryName=; Avatar=}
coreRevision   : [redacted]
timeStamp      :
id             :
revision       : 401853417
profileState   : custom
```

Next, let’s say we want to retrieve all the Projects under each Organisation.

We can use below PowerShell script for that:

```
$userId="{userid}"
$connectionToken="{token}"

$ownerUrl="https://app.vssps.visualstudio.com/_apis/accounts?ownerId=$userId&api-version=6.0"
$base64AuthInfo=[System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes(":$($connectionToken)"))
$accountDetails=Invoke-RestMethod -Uri $ownerUrl -Method Get -ContentType "application/json" -Headers @{Authorization=("Basic {0}" -f $base64AuthInfo)}

$payload = @()

$accountDetails.value | ForEach-Object {

   $orgId=$_.accountName


$base64AuthInfo=[System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes(":$($connectionToken)"))
   $orgUrl="https://dev.azure.com/$orgId/_apis/projects?api-version=6.0"

   $projectObject=(Invoke-RestMethod -Uri $orgUrl -Method Get -UseDefaultCredential -Headers @{Authorization=("Basic {0}" -f $base64AuthInfo)})

   $payload += $projectObject

}

$payload | ConvertTo-Json
```

And you get below result:



```
$payload = @()

$accountDetails.value | ForEach-Object {

    $orgId=$_.accountName

    $base64AuthInfo=[System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes(":$($connectionToken)"))
    $orgUrl="https://dev.azure.com/$orgId/_apis/projects?api-version=6.0"

    $projectObject=(Invoke-RestMethod -Uri $orgUrl -Method Get -UseDefaultCredential -Headers @{Authorization=("Basic {0}" -f $base64AuthInfo)})

    $payload += $projectObject

}

$payload | ConvertTo-Json
```

```
[
    {
        "count":  2,
        "value":  [
                      "@{id=[redacted]; name=patricksameera; url=https://dev.azure.com/patricksameera/_apis/projects/[redacted]
; state=wellFormed; revision=11; visibility=private; lastUpdateTime=2022-09-26T01:57:48.313Z}",
                      "@{id=[redacted]; name=test 123; url=https://dev.azure.com/patricksameera/_apis/projects/[redacted]
 state=wellFormed; revision=19; visibility=private; lastUpdateTime=2022-09-29T06:33:49.357Z}"
                  ]
    },
    {
        "count":  1,
        "value":  [
                      "@{id=[redacted]; name=test project; url=https://dev.azure.com/patricksameeraorg/_apis/projects/[redacted]
; state=wellFormed; revision=11; visibility=private; lastUpdateTime=2022-09-29T06:32:34.4Z}"
                  ]
    }
]
```

By going little furthered ahead, let’s say we want to create a script that we pass Connection Token and User Id and want to retrieve our own formatted Json string that contains both organisation name and project id.

We could use below PowerShell script:

```
param (
[Parameter(Mandatory)]
[String]
[ValidateNotNullOrEmpty()]
$connectionToken,

[Parameter(Mandatory)]
[String]
[ValidateNotNullOrEmpty()]
$userId)


$ownerUrl="https://app.vssps.visualstudio.com/_apis/accounts?ownerId=$userId&api-version=6.0"
$base64AuthInfo=[System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes(":$($connectionToken)"))
$accountDetails=Invoke-RestMethod -Uri $ownerUrl -Method Get -ContentType "application/json" -Headers @{Authorization=("Basic {0}" -f $base64AuthInfo)}

$payload = @()

$accountDetails.value | ForEach-Object {

  $orgId=$_.accountName


$base64AuthInfo=[System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes(":$($connectionToken)"))
  $orgUrl="https://dev.azure.com/$orgId/_apis/projects?api-version=6.0"

  $listProjects=(Invoke-RestMethod -Uri $orgUrl -Method Get -UseDefaultCredential -Headers @{Authorization=("Basic {0}" -f $base64AuthInfo)})

  $listProjects.value | ForEach-Object {

    $props = @{
      OrganisationName = $orgId
      ProjectId = $_.id
      ProjectName = $_.name
      Url = $_.url
      State = $_.state
      Revision = $_.revision
      Visibility = $_.visibility
      LastUpdateTime = $_.lastUpdateTime
    }
```

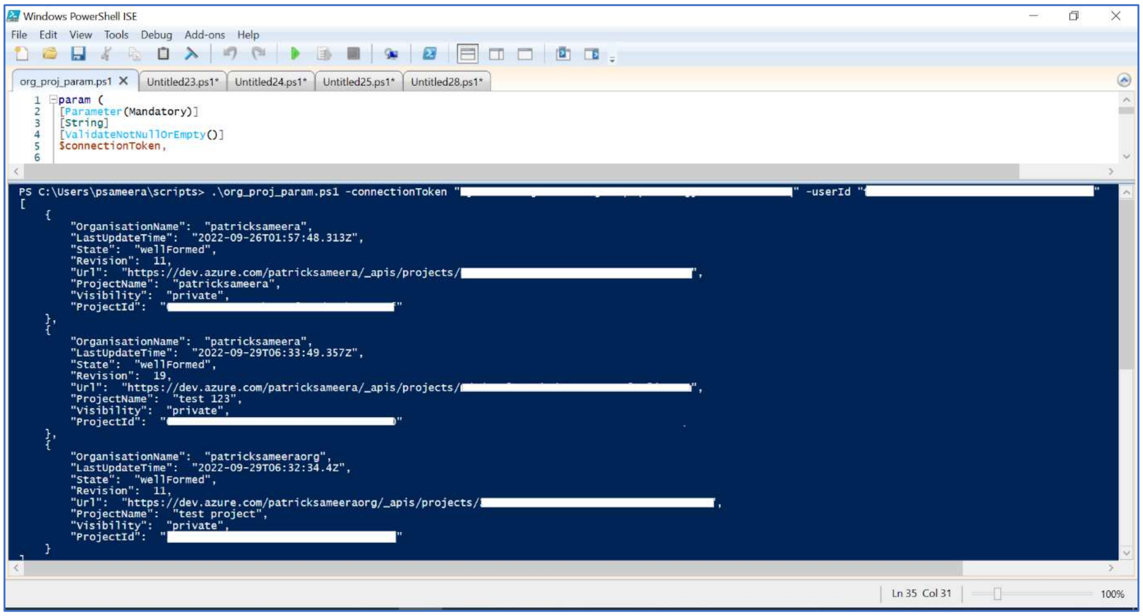
```
        $projectObject = New-Object -TypeName PSObject -Property $props
        $payload += $projectObject
    }
}

$payload | ConvertTo-Json
```

Let’s save this script as org_proj_param.ps1

Now we can execute that script by calling the script and passing correct parameters.

.\org_proj_param.ps1 -connectionToken "{token}" -userId "{userId}"

And you get below result:



```
param (
[Parameter(Mandatory)]
[String]
[ValidateNotNullOrEmpty()]
$connectionToken,
```

```
PS C:\Users\psameera\scripts> .\org_proj_param.ps1 -connectionToken "[redacted]" -userId "[redacted]"
[
    {
        "OrganisationName":  "patricksameera",
        "LastUpdateTime":  "2022-09-26T01:57:48.313Z",
        "State":  "wellFormed",
        "Revision":  11,
        "Url":  "https://dev.azure.com/patricksameera/_apis/projects/[redacted]",
        "ProjectName":  "patricksameera",
        "Visibility":  "private",
        "ProjectId":  "[redacted]"
    },
    {
        "OrganisationName":  "patricksameera",
        "LastUpdateTime":  "2022-09-29T06:33:49.357Z",
        "State":  "wellFormed",
        "Revision":  19,
        "Url":  "https://dev.azure.com/patricksameera/_apis/projects/[redacted]",
        "ProjectName":  "test 123",
        "Visibility":  "private",
        "ProjectId":  "[redacted]"
    },
    {
        "OrganisationName":  "patricksameeraorg",
        "LastUpdateTime":  "2022-09-29T06:32:34.4Z",
        "State":  "wellFormed",
        "Revision":  11,
        "Url":  "https://dev.azure.com/patricksameeraorg/_apis/projects/[redacted],
        "ProjectName":  "test project",
        "Visibility":  "private",
        "ProjectId":  "[redacted]"
    }
]
```