[PowerShell] - Unable to Deserialize Response

Last updated by | Jackie Huang | Jan 4, 2022 at 12:24 AM PST

Contents

- Important Note
- Issue
- Root Cause
- Resolution

Important Note

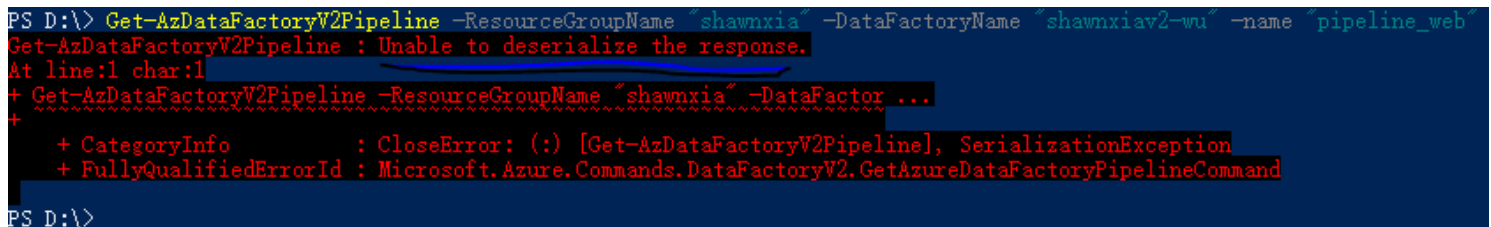Generally, if customer has any PS/SDK issue, please suggest customer upgrade it to the latest version if not yet.

- SDK: https://docs.microsoft.com/en-us/azure/data-factory/quickstart-create-data-factory-dot-net#azure-net-sdk ↗
- PowerShell: https://docs.microsoft.com/en-us/azure/data-factory/quickstart-create-data-factory-powershell#azure-powershell ↗

Issue

When customer using Azure PowerShell to manage ADF resources, they may get following error:

Unable to deserialize the response

An example screenshot(PS cmd could be different from the below one):


```
PS D:\> Get-AzDataFactoryV2Pipeline -ResourceGroupName "shawnxia" -DataFactoryName "shawnxiav2-wu" -name "pipeline_web"
Get-AzDataFactoryV2Pipeline : Unable to deserialize the response.
At line:1 char:1
+ Get-AzDataFactoryV2Pipeline -ResourceGroupName "shawnxia" -DataFactor ...
+
    + CategoryInfo          : CloseError: (:) [Get-AzDataFactoryV2Pipeline], SerializationException
    + FullyQualifiedErrorId : Microsoft.Azure.Commands.DataFactoryV2.GetAzureDataFactoryPipelineCommand

PS D:\>
```

Root Cause

This is usually because customers is not using the latest version of PowerShell Az module or have some properties of the json code with invalid values

Resolution

- Confirm whether customer is using latest Az module (following the doc below, the version showed is the latest version), if not, check whether issue mitigated using the latest one. https://docs.microsoft.com/en-us/powershell/module/az.datafactory/ ↗
  Note: And you can also run below PS command to get what version of the Az module installed in customer's machine and force to load a particular version of the model:
  Get-InstalledModule -Name Az

Get-InstalledModule -Name Az.DataFactory
Import-Module -Name Az.DataFactory -RequiredVersion XXX

- If issue still exists with the latest version, we need to identify which app model object (Pipeline, LinkedService, DataSet, Trigger, etc) caused the issue, so we can work with customer to use the PowerShell command (Get-AzDataFactoryV2Pipeline, Get-AzDataFactoryV2LinkedService, etc) to operate on each single app model object (by using the "-name" option), and check which one caused the error.

```
PS D:\> Get-AzDataFactoryV2Pipeline -ResourceGroupName "shawnxia" -DataFactoryName "shawnxiav2-wu" -name "pipeline_web"
Get-AzDataFactoryV2Pipeline : Unable to deserialize the response.
At line:1 char:1
+ Get-AzDataFactoryV2Pipeline -ResourceGroupName "shawnxia" -DataFactor ...
+
    + CategoryInfo          : CloseError: (:) [Get-AzDataFactoryV2Pipeline], SerializationException
    + FullyQualifiedErrorId : Microsoft.Azure.Commands.DataFactoryV2.GetAzureDataFactoryPipelineCommand

PS D:\>
```

After we identified the problematic one, use the "-Debug" option to execute the same PowerShell command again, then please share the run result with PROD team through ava to confirm what's the real issue and the best solution for it.

```
PS D:\> Get-AzDataFactoryV2Pipeline -ResourceGroupName "shawnxia" -DataFactoryName "shawnxiav2-wu" -name "pipeline_web" -Debug
DEBUG: 2:35:12 PM - GetAzureDataFactoryPipelineCommand begin processing with ParameterSet 'ByFactoryName'.

Confirm
Continue with this operation?
[Y] Yes  [A] Yes to All  [H] Halt Command  [S] Suspend  [?] Help (default is "Y"): A
DEBUG: 2:35:14 PM - using account id 'shawnxia@microsoft.com'...
DEBUG: [Common.Authentication]: Authenticating using Account: 'shawnxia@microsoft.com', environment: 'AzureCloud', tenant: '72f988bf-86f1-41af-91ab-2d7cd011db47'
DEBUG: SharedTokenCacheCredential.GetToken invoked. Scopes: [ https://management.core.windows.net//.default ] ParentRequestId:
DEBUG: Request [9dd55fc5-1a6c-412f-b230-c10075c30be3] POST https://login.microsoftonline.com/72f988bf-86f1-41af-91ab-2d7cd011db47/oauth2/v2.0/token
x-client-SKU:REDACTED
x-client-Ver:REDACTED
x-client-CPU:REDACTED
x-client-OS:REDACTED
```

However, for urgent customers, we can first suggest use generic azure resource PS command and it will always work:

[New-AzResource](#) ⧉

[Get-azresource](#) ⧉

[Get-azresource](#) ⧉

Example of using generic resource cmd:

Step1:

```
$prop=@{
    ResourceName="{Input your DataFactoryName}/{Input your AzureBlobStorage name}"
    ResourceType      = "Microsoft.DataFactory/factories/linkedservices"
    ResourceGroupName = "{input your resource group name}"
    Force             = $true
    Location="{input your location: such as eastus2euap}"
    Properties= @{
        type= "AzureBlobStorage"
        typeProperties=@{
            serviceEndpoint= "https://{your account}.blob.core.windows.net/"
            accountKind="{your account kind}"
        }
    }
}
```

example (using linked service):

```
$prop=@{
    ResourceName="zzytestDF/AzureBlobStorage1205"
    ResourceType      = "Microsoft.DataFactory/factories/linkedservices"
    ResourceGroupName = "zzygroupV2"
    Force             = $true
    Location="eastus2euap"
    Properties= @{
        type= "AzureBlobStorage"
        typeProperties=@{
            serviceEndpoint= "https://myaccount.blob.core.windows.net/"
            accountKind="BlockBlobStorage"
        }
    }
}
```

Step2:

```
New-AzResource @prop
```

Author : Shawn Xiao