



Code generation: From OpenAPI/Swagger schema to PowerShell module in 4 easy steps

Daniele Pecanha & Zhenhao Ye

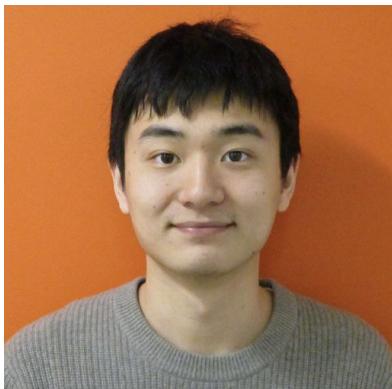
Pure Storage



PURE STORAGE®



Daniele Pecanha



Zhenhao Ye

Contact

Email:

dpecanha@purestorage.com

zye@purestorage.com

Twitter:

[@danielepecanha](https://twitter.com/danielepecanha)

Linkedin:

[danielebpecanha](https://www.linkedin.com/in/danielebpecanha)

[zhenhaoy](https://www.linkedin.com/in/zhenhaoy)

Agenda

- PureStoragePowerShellSDK2 Overview
- Technologies used
- Petstore use case
- Swagger Codegen + Mustache
- Swagger Codegen + Jinja2





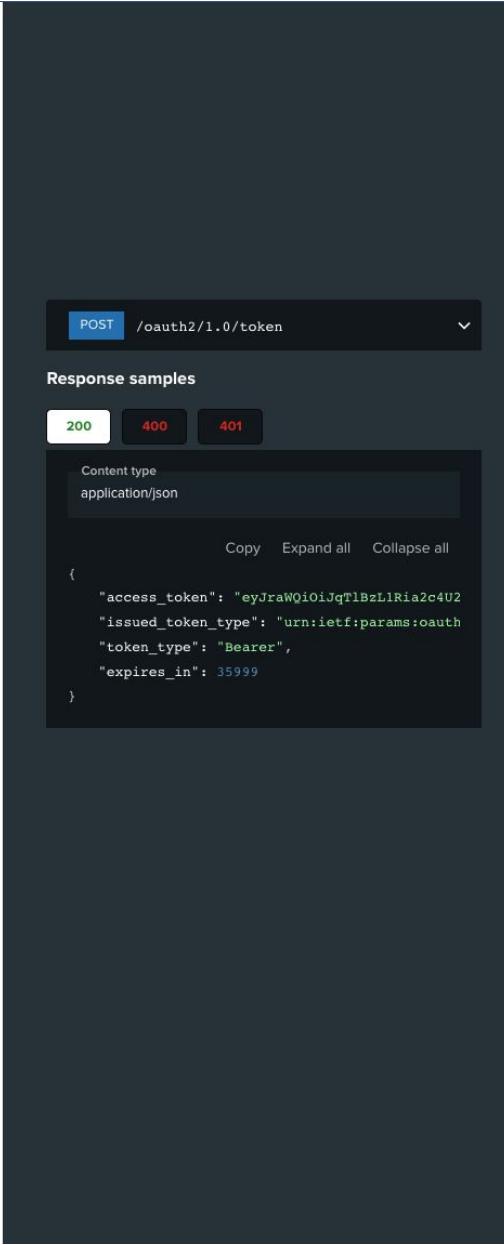
PureStorage PowerShellSDK2

Search...

FlashArray REST API (2.16)

- Authorization > Download OpenAPI specification: [Download](#)
- Active Directory >
- Administrators >
- Alerts >
- Alert Watchers >
- API Clients >
- Apps > Get access token
 - Exchanges an ID Token for an OAuth 2.0 access token.
- Arrays >
- Array Connections >
- Audits > grant_type required
 - string
Default: "urn:ietf:params:oauth:grant-type:token-exchange"
The method by which the access token will be obtained. The Pure Storage REST API supports the OAuth 2.0 "token exchange" grant type, which indicates that a token exchange is being performed. Set `grant_type` to `urn:ietf:params:oauth:grant-type:token-exchange`.
- Certificates >
- Connections > subject_token required
 - string
An encoded security ID Token representing the identity of the party on behalf of whom the request is being made. The token must be issued by a trusted identity provider which must be either a registered application in Pure1 or an enabled API client on the array. The token must be a JSON Web Token and must contain the following claims:

JWT claim	Location	API Client Field	Description	Required By
kid	Header	key_id	Key ID of the API client that issues the identity token.	FlashArray and FlashBlade only.
aud	Payload	id	Client ID of the API client that issues the identity token.	FlashArray and FlashBlade only.
sub	Payload		Login name of the array user for whom the token should be issued. This must be a valid	FlashArray and FlashBlade only.
- Container Default Protections >
- Controllers >
- Directories >
- Directory Exports >
- Directory Quotas >
- Directory Services >
- Directory Snapshots >
- DNS >
- Drives >
- File Systems >
- Hardware >
- Host Groups >



Pure Storage Rest API

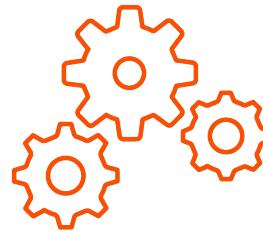
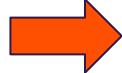
- Manage Pure Storage Flash Arrays
- Create Storage Volumes
- Take Snapshots
- And much more...

Goal

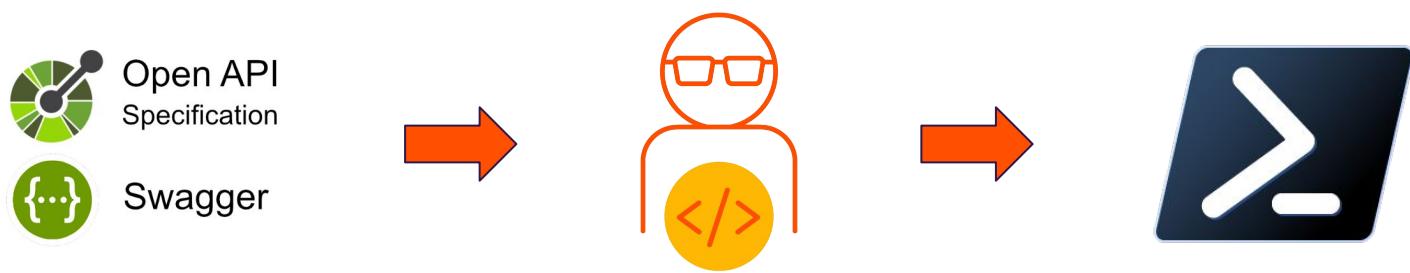


Open API
Specification

Swagger



Manual Approach





Manual Approach

- **50+ Tags**
- **150+ Paths**
- **7 releases in 2022**

Code Generation



- **Faster Releases**
- **Higher Coverage**
- **Less Error Prone**
- **Optimize Dev Resources**



Technologies

Swagger / Open API Specification

"The OpenAPI Specification (OAS) defines a standard, language-agnostic interface to HTTP APIs which allows both humans and computers to discover and understand the capabilities of the service without access to source code, documentation, or through network traffic inspection."¹

The specification can include information such as:

- Operations supported by the API
- API parameters and return types
- Authorization
- Constraints, comments, examples, ...

¹ <https://swagger.io/specification/>

Swagger Codegen

“The Swagger Codegen project, allows generation of API client libraries (SDK generation), server stubs and documentation automatically given an OpenAPI Spec.”²

Some of the languages/frameworks supported for API Clients are:

- C# (.net 2.0, 3.5 or later)
- C++ (cpprest, Qt5, Tizen)
- PowerShell
- Python
- Typescript (Angular1.x, Angular2.x, Fetch, jQuery, Node)

² <https://github.com/swagger-api/swagger-codegen>

Mustache Templates

"Mustache can be used for HTML, config files, source code - anything. It works by expanding tags in a template using values provided in a hash or object.

*We call it "logic-less" because there are no if statements, else clauses, or for loops. Instead there are only tags. Some tags are replaced with a value, some nothing, and others a series of values."*³

³ <http://mustache.github.io/>



Jinja 2

“Jinja is a fast, expressive, extensible templating engine. Special placeholders in the template allow writing code similar to Python syntax. Then the template is passed data to render the final document.”⁴

- A popular and powerful templating engine for Python.
- Jinja2 allows for dynamic generation of multiple file formats
- Jinja2 supports logical operator like for loop and If condition block
- Jinja2 uses a syntax that is familiar to Python developers.

⁴ <https://jinja.palletsprojects.com/>





Petstore

Swagger Petstore

<https://petstore.swagger.io/>

The screenshot shows the Swagger Petstore interface. At the top, it displays the title "Swagger Petstore 1.0.6" and the base URL "[Base URL: petstore.swagger.io/v2]". Below this, the URL "https://petstore.swagger.io/v2/swagger.json" is shown. A note states: "This is a sample server Petstore server. You can find out more about Swagger at <http://swagger.io> or on [irc.freenode.net, #swagger](#). For this sample, you can use the api key **special-key** to test the authorization filters." Navigation links include "Terms of service", "Contact the developer", "Apache 2.0", and "Find out more about Swagger". On the left, there's a "Schemes" dropdown set to "HTTPS" and an "Authorize" button with a lock icon. The main content area is titled "pet Everything about your Pets" with a "Find out more" link and a collapse/expand arrow. It lists four operations: "POST /pet/{petId}/uploadImage uploads an image", "POST /pet Add a new pet to the store", "PUT /pet Update an existing pet", and "GET /pet/findByStatus Finds Pets by status". Each operation has a collapse/expand arrow and a lock icon.

- Sample from swagger.io
- 20 paths
- 3 APIs
- Authentication not required

Swagger Petstore

pet Everything about your Pets [Find out more](#) ^

- POST** `/pet/{petId}/uploadImage` uploads an image [▼](#) [🔒](#)
- POST** `/pet` Add a new pet to the store [▼](#) [🔒](#)
- PUT** `/pet` Update an existing pet [▼](#) [🔒](#)
- GET** `/pet/findByStatus` Finds Pets by status [▼](#) [🔒](#)
- GET** `/pet/findByTags` Finds Pets by tags [▼](#) [🔒](#)
- GET** `/pet/{petId}` Find pet by ID [▼](#) [🔒](#)
- POST** `/pet/{petId}` Updates a pet in the store with form data [▼](#) [🔒](#)
- DELETE** `/pet/{petId}` Deletes a pet [▼](#) [🔒](#)

store Access to Petstore orders

POST `/store/order` Place an order for a pet [▼](#)

GET `/store/order/{orderId}` Find purchase order by ID [▼](#)

DELETE `/store/order/{orderId}` Delete purchase order by ID [▼](#)

GET `/store/inventory` Returns pet inventories by status [▼](#) [🔒](#)

user Operations about user

[Find out more about our store](#) ^

POST `/user/createWithArray` Creates list of users with given input array [▼](#)

POST `/user/createWithList` Creates list of users with given input array [▼](#)

GET `/user/{username}` Get user by user name [▼](#)

PUT `/user/{username}` Updated user [▼](#)

DELETE `/user/{username}` Delete user [▼](#)

GET `/user/login` Logs user into the system [▼](#)

GET `/user/logout` Logs out current logged in user session [▼](#)

POST `/user` Create user [▼](#)

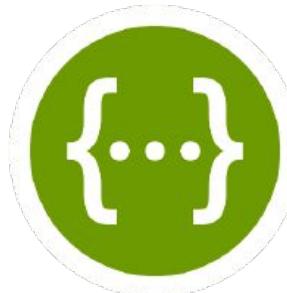
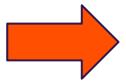


Swagger Codegen + Mustache

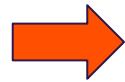
Initial Plan



YAML File



Swagger
Codegen



PowerShell
Module

```
PS Z:\DemoVMShare\src\PowerShellSummitDemo\Demo> Invoke-WebRequest -Uri https://petstore.swagger.io/v2/swagger.yaml -OutFile './swagger.yaml'
PS Z:\DemoVMShare\src\PowerShellSummitDemo\Demo> dir -Recurse

Directory: Z:\DemoVMShare\src\PowerShellSummitDemo\Demo

Mode                LastWriteTime       Length Name
----                -----          ---- 
----        4/10/2023 5:24 PM      16718314 swagger-codegen-cli.jar
----        4/10/2023 5:49 PM       17761 swagger.yaml

PS Z:\DemoVMShare\src\PowerShellSummitDemo\Demo> java -jar swagger-codegen-cli.jar generate -i ./swagger.yaml -l powershell -o petstore/powershell
[main] INFO io.swagger.parser.Swagger20Parser - reading from ./swagger.yaml
[main] WARN io.swagger.codegen.IgnoreProcessor - Output directory does not exist, or is inaccessible. No file (.swagger-codegen-ignore) will be evaluated.
[main] INFO io.swagger.codegen.AbstractGenerator - writing file Z:\DemoVMShare\src\PowerShellSummitDemo\Demo\petstore\powershell\src\IO.Swagger\Model\New-ApiResponse.ps1
[main] INFO io.swagger.codegen.AbstractGenerator - writing file Z:\DemoVMShare\src\PowerShellSummitDemo\Demo\petstore\powershell\test\ApiResponseTest.ps1
[main] INFO io.swagger.codegen.AbstractGenerator - writing file Z:\DemoVMShare\src\PowerShellSummitDemo\Demo\petstore\powershell\src\IO.Swagger\Model\New-Category.ps1
[main] INFO io.swagger.codegen.AbstractGenerator - writing file Z:\DemoVMShare\src\PowerShellSummitDemo\Demo\petstore\powershell\test\CategoryTest.ps1
[main] INFO io.swagger.codegen.AbstractGenerator - writing file Z:\DemoVMShare\src\PowerShellSummitDemo\Demo\petstore\powershell\src\IO.Swagger\Model\New-Order.ps1
[main] INFO io.swagger.codegen.AbstractGenerator - writing file Z:\DemoVMShare\src\PowerShellSummitDemo\Demo\petstore\powershell\test\OrderTest.ps1
[main] INFO io.swagger.codegen.AbstractGenerator - writing file Z:\DemoVMShare\src\PowerShellSummitDemo\Demo\petstore\powershell\src\IO.Swagger\Model\New-Pet.ps1
[main] INFO io.swagger.codegen.AbstractGenerator - writing file Z:\DemoVMShare\src\PowerShellSummitDemo\Demo\petstore\powershell\test\PetTest.ps1
[main] INFO io.swagger.codegen.AbstractGenerator - writing file Z:\DemoVMShare\src\PowerShellSummitDemo\Demo\petstore\powershell\src\IO.Swagger\Model\New-Tag.ps1
[main] INFO io.swagger.codegen.AbstractGenerator - writing file Z:\DemoVMShare\src\PowerShellSummitDemo\Demo\petstore\powershell\test\TagTest.ps1
[main] INFO io.swagger.codegen.AbstractGenerator - writing file Z:\DemoVMShare\src\PowerShellSummitDemo\Demo\petstore\powershell\src\IO.Swagger\Model\New-User.ps1
[main] INFO io.swagger.codegen.AbstractGenerator - writing file Z:\DemoVMShare\src\PowerShellSummitDemo\Demo\petstore\powershell\test\UserTest.ps1
[main] INFO io.swagger.codegen.AbstractGenerator - writing file Z:\DemoVMShare\src\PowerShellSummitDemo\Demo\petstore\powershell\src\IO.Swagger\API\PetApi.ps1
[main] INFO io.swagger.codegen.AbstractGenerator - writing file Z:\DemoVMShare\src\PowerShellSummitDemo\Demo\petstore\powershell\test\PetApiTest.ps1
[main] INFO io.swagger.codegen.AbstractGenerator - writing file Z:\DemoVMShare\src\PowerShellSummitDemo\Demo\petstore\powershell\src\IO.Swagger\API\StoreApi.ps1
[main] INFO io.swagger.codegen.AbstractGenerator - writing file Z:\DemoVMShare\src\PowerShellSummitDemo\Demo\petstore\powershell\test\StoreApiTest.ps1
[main] INFO io.swagger.codegen.AbstractGenerator - writing file Z:\DemoVMShare\src\PowerShellSummitDemo\Demo\petstore\powershell\src\IO.Swagger\API\UserApi.ps1
[main] INFO io.swagger.codegen.AbstractGenerator - writing file Z:\DemoVMShare\src\PowerShellSummitDemo\Demo\petstore\powershell\test\UserApiTest.ps1
[main] INFO io.swagger.codegen.AbstractGenerator - writing file Z:\DemoVMShare\src\PowerShellSummitDemo\Demo\petstore\powershell\README.md
[main] INFO io.swagger.codegen.AbstractGenerator - writing file Z:\DemoVMShare\src\PowerShellSummitDemo\Demo\petstore\powershell\Build.ps1
[main] INFO io.swagger.codegen.AbstractGenerator - writing file Z:\DemoVMShare\src\PowerShellSummitDemo\Demo\petstore\powershell\Private\Get-CommonParameters.ps1
[main] INFO io.swagger.codegen.DefaultGenerator - writing file Z:\DemoVMShare\src\PowerShellSummitDemo\Demo\petstore\powershell\src\IO.Swagger\Private\Get-CommonParameters.ps1
[main] INFO io.swagger.codegen.DefaultGenerator - writing file Z:\DemoVMShare\src\PowerShellSummitDemo\Demo\petstore\powershell\src\IO.Swagger\Private\Out-DebugParameter.ps1
[main] INFO io.swagger.codegen.AbstractGenerator - writing file Z:\DemoVMShare\src\PowerShellSummitDemo\Demo\petstore\powershell\src\IO.Swagger\en-US/about_IO.Swagger.help.txt
[main] INFO io.swagger.codegen.AbstractGenerator - writing file Z:\DemoVMShare\src\PowerShellSummitDemo\Demo\petstore\powershell\.swagger-codegen-ignore
[main] INFO io.swagger.codegen.AbstractGenerator - writing file Z:\DemoVMShare\src\PowerShellSummitDemo\Demo\petstore\powershell\VERSION
PS Z:\DemoVMShare\src\PowerShellSummitDemo\Demo>
```

EXPLORER

DEMO

petstore\powershell

.swagger-codegen

VERSION

src\IO.Swagger

API

PetApi.ps1

StoreApi.ps1

UserApi.ps1

en-US

about_IO.Swagger.help.txt

Model

New-ApiResponse.ps1

New-Category.ps1

New-Order.ps1

New-Pet.ps1

New-Tag.ps1

New-User.ps1

Private

Get-CommonParameters.ps1

Out-DebugParameter.ps1

IO.Swagger.psm1

test

ApiResponseTest.ps1

CategoryTest.ps1

OrderTest.ps1

PetApiTest.ps1

...

StoreApi.ps1

petstore > powershell > src > IO.Swagger > API > StoreApi.ps1 > ...

50

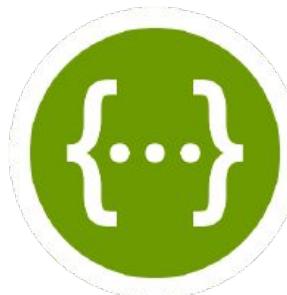
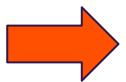
0 references

```
51 function Invoke-StoreApiPlaceOrder {
52     [CmdletBinding()]
53     Param (
54         [Parameter(Position = 0, ValueFromPipeline = $true, ValueFromPipelineByPropertyName = $true, Mandatory = $true)]
55         [IO.Swagger.Model.Order]
56         ${body}
57     )
58
59     Process {
60         'Calling method: StoreApi-PlaceOrder' | Write-Verbose
61         $PSBoundParameters | Out-DebugParameter | Write-Debug
62
63         $Script:StoreApi.PlaceOrder(
64             ${body}
65         )
66     }
67 }
68
69 
```

Initial Plan



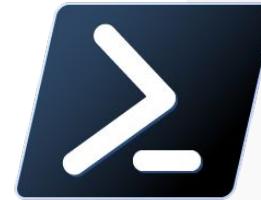
Yaml File



Swagger
Codegen

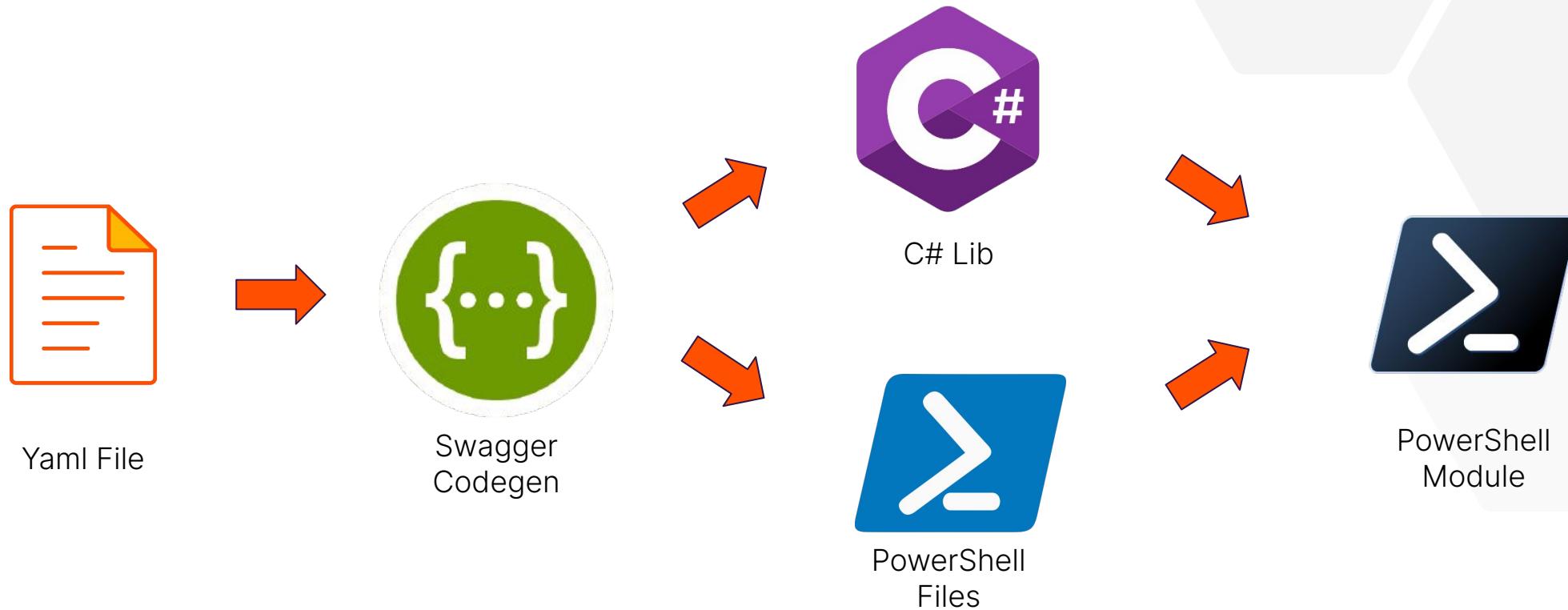


Requires
C# lib



PowerShell
Module

New Plan



Windows PowerShell

Windows PowerShell

```
PS Z:\DemoVMShare\src\PowerShellSummitDemo\Demo> ./petstore/powershell/Build.ps1
Feeds used:
  https://api.nuget.org/v3/index.json

Restoring NuGet package RestSharp.105.1.0.
Restoring NuGet package Newtonsoft.Json.10.0.3.
  GET https://api.nuget.org/v3-flatcontainer/newtonsoft.json/10.0.3/newtonsoft.json.10.0.3.nupkg
  GET https://api.nuget.org/v3-flatcontainer/restsharp/105.1.0/restsharp.105.1.0.nupkg
  OK https://api.nuget.org/v3-flatcontainer/newtonsoft.json/10.0.3/newtonsoft.json.10.0.3.nupkg 10ms
  OK https://api.nuget.org/v3-flatcontainer/restsharp/105.1.0/restsharp.105.1.0.nupkg 48ms
Installed RestSharp 105.1.0 from https://api.nuget.org/v3/index.json with content hash 2DRH1cpTOJwesB0wQ0cIhVHUmpbRkt+tdSatfdvndvsawsGMIegrBq8yv6g3GM1MRVcb37bT8LACVFsqITDmg==.
Installed Newtonsoft.Json 10.0.3 from https://api.nuget.org/v3/index.json with content hash h5xaFmh7hNCuEoC4XNV5DrRkLDzYHqPx/Ik23R4J86Z7PE/X6YidhG602dFvdLBRSdG6xp9NabH3dXpcoxWvw==.
Adding package 'RestSharp.105.1.0' to folder 'Z:\DemoVMShare\src\PowerShellSummitDemo\Demo\petstore\powershell\csharp\SwaggerClient\packages'
Adding package 'Newtonsoft.Json.10.0.3' to folder 'Z:\DemoVMShare\src\PowerShellSummitDemo\Demo\petstore\powershell\csharp\SwaggerClient\packages'
Added package 'Newtonsoft.Json.10.0.3' to folder 'Z:\DemoVMShare\src\PowerShellSummitDemo\Demo\petstore\powershell\csharp\SwaggerClient\packages'
Restoring NuGet package JsonSubTypes.1.2.0.
  GET https://api.nuget.org/v3-flatcontainer/jsonsubtypes/1.2.0/jsonsubtypes.1.2.0.nupkg
  OK https://api.nuget.org/v3-flatcontainer/jsonsubtypes/1.2.0/jsonsubtypes.1.2.0.nupkg 12ms
Installed JsonSubTypes 1.2.0 from https://api.nuget.org/v3/index.json with content hash XY9yuNI19LmHhC/Vz+u35lB7Dyj/H/biKsNybjI9XLuyabBIHPF1PQ3PY/ffADn/LdjiQ0hx/ry+ZsuE04mBnw==.
Adding package 'JsonSubTypes.1.2.0' to folder 'Z:\DemoVMShare\src\PowerShellSummitDemo\Demo\petstore\powershell\csharp\SwaggerClient\packages'
Added package 'RestSharp.105.1.0' to folder 'Z:\DemoVMShare\src\PowerShellSummitDemo\Demo\petstore\powershell\csharp\SwaggerClient\packages'
Added package 'JsonSubTypes.1.2.0' to folder 'Z:\DemoVMShare\src\PowerShellSummitDemo\Demo\petstore\powershell\csharp\SwaggerClient\packages'
  1 file(s) copied.
  1 file(s) copied.
  1 file(s) copied.
Microsoft (R) Visual C# Compiler version 4.7.3190.0
for C# 5
Copyright (C) Microsoft Corporation. All rights reserved.

This compiler is provided as part of the Microsoft (R) .NET Framework, but only supports language versions up to C# 5, which is no longer the latest version. For compilers that support newer versions of the C# programming language, see http://go.microsoft.com/fwlink/?LinkID=533240

src\IO.Swagger\Client\ApiClient.cs(503,90): warning CS1573: Parameter 'collectionFormat' has no matching param tag in the XML comment for 'IO.Swagger.Client.ApiClient.ParameterToKeyValuePairs(string, string, object)' (but other parameters do)
PS Z:\DemoVMShare\src\PowerShellSummitDemo\Demo>
```

DEMO



- petstore\powershell
- .swagger-codegen
- VERSION
- csharp
- src\IO.Swagger
 - API
 - Bin
 - IO.Swagger.dll
 - JsonSubTypes.dll
 - Newtonsoft.Json.dll
 - RestSharp.dll
 - en-US
 - Model
 - Private

IO.Swagger.psd1

IO.Swagger.psm1

test

.swagger-codegen-ignore

Build.ps1

README.md

swagger-codegen-cli.jar

swagger.yaml

```
petstore > powershell > src > IO.Swagger > IO.Swagger.psd1
54 # RequiredModules = @()
55
56 # Assemblies that must be loaded prior to importing this module
57 RequiredAssemblies = 'Bin\JsonSubTypes.dll', 'Bin\Newtonsoft.Json.dll',
58 | | | | 'Bin\RestSharp.dll', 'Bin\IO.Swagger.dll'
59
60 # Script files (.ps1) that are run in the caller's environment prior to importing this module.
61 # ScriptsToProcess = @()
62
63 # Type files (.ps1xml) to be loaded when importing this module
64 # TypesToProcess = @()
65
66 # Format files (.ps1xml) to be loaded when importing this module
67 # FormatsToProcess = @()
68
69 # Modules to import as nested modules of the module specified in RootModule/ModuleToProcess
70 # NestedModules = @()
71
72 # Functions to export from this module, for best performance, do not use wildcards and do not de-
73 FunctionsToExport = 'Invoke-UserApiCreateUser',
74 | | | | 'Invoke-UserApiCreateUsersWithArrayInput',
75 | | | | 'Invoke-UserApiCreateUsersWithListInput',
76 | | | | 'Invoke-UserApiDeleteUser', 'Invoke-UserApiGetUserByName',
77 | | | | 'Invoke-UserApiLoginUser', 'Invoke-UserApiLogoutUser',
78 | | | | 'Invoke-UserApiUpdateUser', 'Invoke-PetApiAddPet',
79 | | | | 'Invoke-PetApiDeletePet', 'Invoke-PetApiFindPetsByStatus',
80 | | | | 'Invoke-PetApiFindPetsByTags', 'Invoke-PetApiGetPetById',
81 | | | | 'Invoke-PetApiUpdatePet', 'Invoke-PetApiUpdatePetWithForm',
82 | | | | 'Invoke-PetApiUploadFile', 'Invoke-StoreApiDeleteOrder',
83 | | | | 'Invoke-StoreApiGetInventory', 'Invoke-StoreApiGetOrderById',
84 | | | | 'Invoke-StoreApiPlaceOrder', 'New-User', 'New-ApiResponse', 'New-Pet',
85 | | | | 'New-Category', 'New-Order', 'New-Tag'
86
```

Let's take a look at the generated module...

```
PS Z:\DemoVMShare\src\PowerShellSummitDemo\Demo> Import-Module .\petstore\powershell\src\IO.Swagger\IO.Swagger.psd1
PS Z:\DemoVMShare\src\PowerShellSummitDemo\Demo> Get-Command -Module IO.Swagger

 CommandType      Name          Version   Source
 -----          ----          -----   -----
 Function        Invoke-PetApiAddPet    1.0      IO.Swagger
 Function        Invoke-PetApiDeletePet  1.0      IO.Swagger
 Function        Invoke-PetApiFindPetsByStatus 1.0      IO.Swagger
 Function        Invoke-PetApiFindPetsByTags   1.0      IO.Swagger
 Function        Invoke-PetApiGetPetById    1.0      IO.Swagger
 Function        Invoke-PetApiUpdatePet   1.0      IO.Swagger
 Function        Invoke-PetApiUpdatePetWithForm 1.0      IO.Swagger
 Function        Invoke-PetApiUploadFile   1.0      IO.Swagger
 Function        Invoke-StoreApiDeleteOrder 1.0      IO.Swagger
 Function        Invoke-StoreApiGetInventory 1.0      IO.Swagger
 Function        Invoke-StoreApiGetOrderById 1.0      IO.Swagger
 Function        Invoke-StoreApiPlaceOrder  1.0      IO.Swagger
 Function        Invoke-UserApiCreateUser   1.0      IO.Swagger
 Function        Invoke-UserApiCreateUsersWithArrayInput 1.0      IO.Swagger
 Function        Invoke-UserApiCreateUsersWithListInput 1.0      IO.Swagger
 Function        Invoke-UserApiDeleteUser   1.0      IO.Swagger
 Function        Invoke-UserApiGetUserByName 1.0      IO.Swagger
 Function        Invoke-UserApiLoginUser   1.0      IO.Swagger
 Function        Invoke-UserApiLogoutUser  1.0      IO.Swagger
 Function        Invoke-UserApiUpdateUser  1.0      IO.Swagger
 Function        New-ApiResponse   1.0      IO.Swagger
 Function        New-Category    1.0      IO.Swagger
 Function        New-Order       1.0      IO.Swagger
 Function        New-Pet        1.0      IO.Swagger
 Function        New-Tag        1.0      IO.Swagger
 Function        New-User        1.0      IO.Swagger

PS Z:\DemoVMShare\src\PowerShellSummitDemo\Demo>
```

Let's take a look at the generated module...

```
PS Z:\DemoVMShare\src\PowerShellSummitDemo\Demo> Get-Help Invoke-StoreApiPlaceOrder -Full

NAME
    Invoke-StoreApiPlaceOrder

SYNTAX
    Invoke-StoreApiPlaceOrder [-body] <Order>  [<CommonParameters>]

PARAMETERS
    -body <Order>

        Required?                true
        Position?                0
        Accept pipeline input?   true (ByValue, ByPropertyName)
        Parameter set name       (All)
        Aliases                  None
        Dynamic?                 false

    <CommonParameters>
        This cmdlet supports the common parameters: Verbose, Debug,
        ErrorAction, ErrorVariable, WarningAction, WarningVariable,
        OutBuffer, PipelineVariable, and OutVariable. For more information, see
        about_CommonParameters (https://go.microsoft.com/fwlink/?LinkID=113216).

INPUTS
    IO.Swagger.Model.Order

OUTPUTS
    System.Object

ALIASES
    None

REMARKS
    None
```

Let's take a look at the generated module...

```
PS Z:\DemoVMShare\src\PowerShellSummitDemo\Demo> $order = New-Order -id 1 -petId 125 -quantity 1 -shipDate (Get-Date) -status "approved" -complete $true
PS Z:\DemoVMShare\src\PowerShellSummitDemo\Demo> $order

Status      : Approved
Id          : 1
PetId       : 125
Quantity    : 1
ShipDate    : 4/10/2023 6:25:24 PM
Complete    : True

PS Z:\DemoVMShare\src\PowerShellSummitDemo\Demo> Invoke-StoreApiPlaceOrder -body $order

Status      : Approved
Id          : 1
PetId       : 125
Quantity    : 1
ShipDate    : 4/10/2023 6:25:24 PM
Complete    : True

PS Z:\DemoVMShare\src\PowerShellSummitDemo\Demo> ■
```

Generated-code Issues

- Syntax not user friendly
- Not enough help/examples
- Complex objects created in multiple calls



Updating Syntax w/ Mustache *logic-less* Templates

```
api.mustache X  
mustache > api.mustache  
1  {{#operations}}  
2  {{#operation}}  
3  function Invoke-{{classname}}{{operationId}} {  
4      [CmdletBinding()]  
5      Param (  
6          {{#allParams}}  
7              [Parameter(Position = {{vendorExtensions.x-index}}, ValueFromPipeline = $true)]  
8              [{{^isContainer}}{{^isPrimitiveType}}{{^isFile}}{{packageName}}].Model.{{= <% %>}}  
9              {{=<% %>}}  
10             ${<%paramName%>}{{^last}},{{^last}}  
11             <%={{ }}=%>  
12             {{/allParams}}  
13     )  
14  
15     Process {  
16         'Calling method: {{classname}}-{{operationId}}' | Write-Verbose  
17         $PSBoundParameters | Out-DebugParameter | Write-Debug  
18  
19         $Script:{{classname}}.{{operationId}}(  
20             {{#allParams}}  
21                 {{= <% %>}}  
22                 ${<%paramName%>}{{^last}},{{^last}}  
23                 <%={{ }}=%>  
24             {{/allParams}}  
25         )  
26     }  
27 }  
28  
29 {{/operation}}  
30 {{/operations}}
```



```
api.mustache X  
mustache > api.mustache  
1  {{#operations}}  
2  {{#operation}}  
3  function Invoke-{{operationId}} {  
4      [CmdletBinding()]  
5      Param (  
6          {{#allParams}}  
7              [Parameter(Position = {{vendorExtensions.x-index}}, ValueFromPipeline = $true)]  
8              [{{^isContainer}}{{^isPrimitiveType}}{{^isFile}}{{packageName}}].Model.{{= <% %>}}  
9              {{=<% %>}}  
10             ${<%paramName%>}{{^last}},{{^last}}  
11             <%={{ }}=%>  
12             {{/allParams}}  
13     )  
14  
15     Process {  
16         'Calling method: {{operationId}}' | Write-Verbose  
17         $PSBoundParameters | Out-DebugParameter | Write-Debug  
18  
19         $Script:{{classname}}.{{operationId}}(  
20             {{#allParams}}  
21                 {{= <% %>}}  
22                 ${<%paramName%>}{{^last}},{{^last}}  
23                 <%={{ }}=%>  
24             {{/allParams}}  
25         )  
26     }  
27 }  
28  
29 {{/operation}}  
30 {{/operations}}
```

Updating Syntax w/ Mustache *logic-less* Templates

```
PS Z:\DemoVMShare\src\PowerShellSummitDemo\Demo> Get-Command -Module IO.Swagger

CommandType      Name                           Version   Source
----           ----
Function        Invoke-AddPet                 1.0       IO.Swagger
Function        Invoke-CREATEUser             1.0       IO.Swagger
Function        Invoke-CREATEUsersWithArrayInput 1.0       IO.Swagger
Function        Invoke-CREATEUsersWithListInput 1.0       IO.Swagger
Function        Invoke-DeleteOrder            1.0       IO.Swagger
Function        Invoke-DeletePet              1.0       IO.Swagger
Function        Invoke-DeleteUser             1.0       IO.Swagger
Function        Invoke-FindPetsByStatus       1.0       IO.Swagger
Function        Invoke-FindPetsByTags         1.0       IO.Swagger
Function        Invoke-GetInventory           1.0       IO.Swagger
Function        Invoke-GetOrderById            1.0       IO.Swagger
Function        Invoke-GetPetById              1.0       IO.Swagger
Function        Invoke-GetUserByName           1.0       IO.Swagger
Function        Invoke-LoginUser              1.0       IO.Swagger
Function        Invoke-LogoutUser             1.0       IO.Swagger
Function        Invoke-PlaceOrder             1.0       IO.Swagger
Function        Invoke-UpdatePet              1.0       IO.Swagger
Function        Invoke-UpdatePetWithForm       1.0       IO.Swagger
Function        Invoke-UpdateUser             1.0       IO.Swagger
Function        Invoke-UploadFile             1.0       IO.Swagger
Function        New-ApiResponse             1.0       IO.Swagger
Function        New-Category                1.0       IO.Swagger
Function        New-Order                  1.0       IO.Swagger
Function        New-Pet                   1.0       IO.Swagger
Function        New-Tag                   1.0       IO.Swagger
Function        New-User                   1.0       IO.Swagger

PS Z:\DemoVMShare\src\PowerShellSummitDemo\Demo>
```

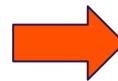


Swagger Codegen + Jinja2

Updating Syntax with Jinja2 *logical* Templates

```
PowerShellModule.pst.jinja2 x
1 # Generated from: PowerShellModule.cs.jinja2
2
3 {# PowerShell parameter definition and related attributes #}
4 {% macro PowerShellParam (param, pipeline_param, path, method, cmdlet) %}
5     {% if 'name' in param.keys() %} (# NOTE: This seems to always be true #)
6         {% set is_required = 'required' in param.keys() and param['required']%}
7         {% set is_pipeline = pipeline_param == param['name'] %}
8         # {{ param['originalName'] }}, name={{ param['name'] }}, path: {{ param['source_path'] }}}
9         {{ param | get_param_attributes_by_path(path, method, cmdlet) | safe }}
10        {{ param | get_type_name | get_type_override_by_path(path, method, param['name'])| safe}}
11        ${{ param['name'] }}
12    {% endif %}
13 {% endmacro %}
14
15 {# Call the constructor of a non-basic type parameter #}
16 {% macro ComplexParamInitialization(param, method, path) %}
17 New-{{ param['mapped-type'] | safe }} {{ for complex_param in param |get_complex_type_params if not param['mapped-type'] | is_constructor_param_excluded (complex_param, method) }} -{{ complex_param['originalName'] }} ${{ complex_param['name'] }} {{% endfor %}}
18 {% endmacro %}
19
20 {% macro ParamComment(param, pipeline_param) %}
21 {%- if 'parent_reference_type' in param %}
22     {% set reference_text = ' reference='''+param['parent_reference_type']+ ''' %}
23 {%- else %}
24     {% set reference_text =''' %}
25 {%- endif %}
26 .PARAMETER {{ param['name'] }}
27     {% if 'description' in param %} {{ param['description'].replace('\n', ' ') | filter_description}}{% endif %}
28 {% endmacro %}
29
30 {# PowerShell Module Generation #}
31 {% macro Cmdlet(verb, method, class_name, path, item) %}
32     {% set param_map = params_map[path][method.lower()] %}
33     {% set pipeline_param = param_map | get_pipeline_param_name %}
34     {% set object_name = path | get_class_name %}
35     # Path: {{ path }}
36     function {{ verb }}-PetStore{{object_name}} {
37         # Function description help #
38         #.SYNOPSIS
39         {{item.summary}}
40         .DESCRIPTION
41         {{item.description}}
42         {% set list_reference_types = param_map | get_list_of_references_types %}
43         {% for param_name, param in param_map.items() %}
44             {{ ParamComment(param, pipeline_param) }}
45             {% endfor %}
46         #>
47     
```

Jinja2 template



```
PetStoreSDK-order.ps1 M x
powershell > src > Pure.Swagger > scripts > build > PetStoreSDK-order.ps1 > function New-PetStoreOrder()
1 # Generated from: PowerShellModule.cs.jinja2
2 # Path: /store/order
0 references
3 function New-PetStoreOrder {
4     <#
5     .SYNOPSIS
6     | Place an order for a pet
7     .DESCRIPTION
8     | Place an order for a pet. Full fill the order information as parameter.
9     .PARAMETER Id
10    Order id
11    .PARAMETER Petid
12    Order pet id
13    .PARAMETER Quantity
14    Order quantity
15    .PARAMETER Shipdate
16    date of shipping
17    .PARAMETER Status
18    Order Status
19    .PARAMETER Complete
20    Order completion Status
21    #>
22
23 Param[
24     # id, name=Id, path: |
25     [Parameter(Mandatory=$false, ValueFromPipeline = $false, ValueFromPipelineByPropertyName = $false )]
26     [long]
27     $Id,
28     # petid, name=Petid, path:
29     [Parameter(Mandatory=$false, ValueFromPipeline = $false, ValueFromPipelineByPropertyName = $false )]
30     [long]
31     $Petid,
32     # quantity, name=Quantity, path:
33     [Parameter(Mandatory=$false, ValueFromPipeline = $false, ValueFromPipelineByPropertyName = $false )]
34     [int]
35     $Quantity,
36     # shipdate, name=Shipdate, path:
37     [Parameter(Mandatory=$false, ValueFromPipeline = $false, ValueFromPipelineByPropertyName = $false )]
38     [string]
39     $Shipdate,
40     # status, name>Status, path:
41     [Parameter(Mandatory=$false, ValueFromPipeline = $false, ValueFromPipelineByPropertyName = $false )]
42     [string]
43     $Status,
44     # complete, name=Complete, path:
45     [Parameter(Mandatory=$false, ValueFromPipeline = $false, ValueFromPipelineByPropertyName = $false )]
46     [boolean]
```

Rendered PowerShell scripts

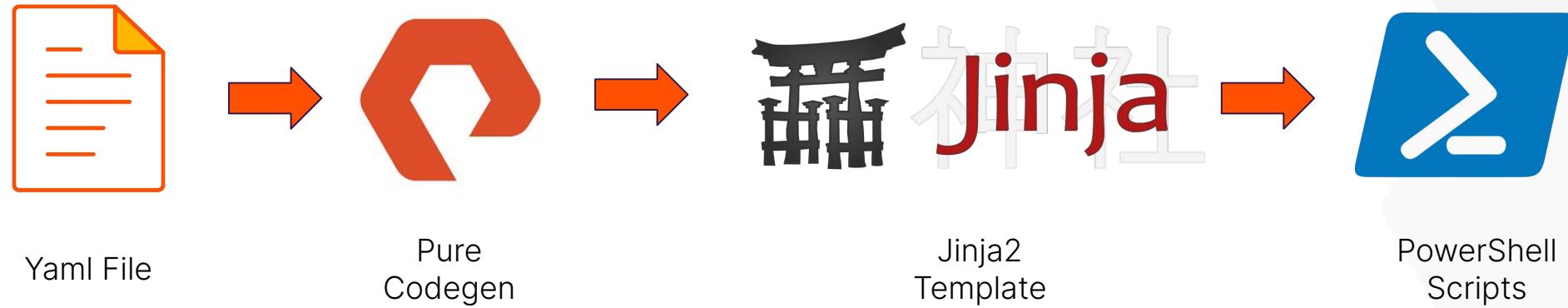
Overall Workflow

Generate C# Library



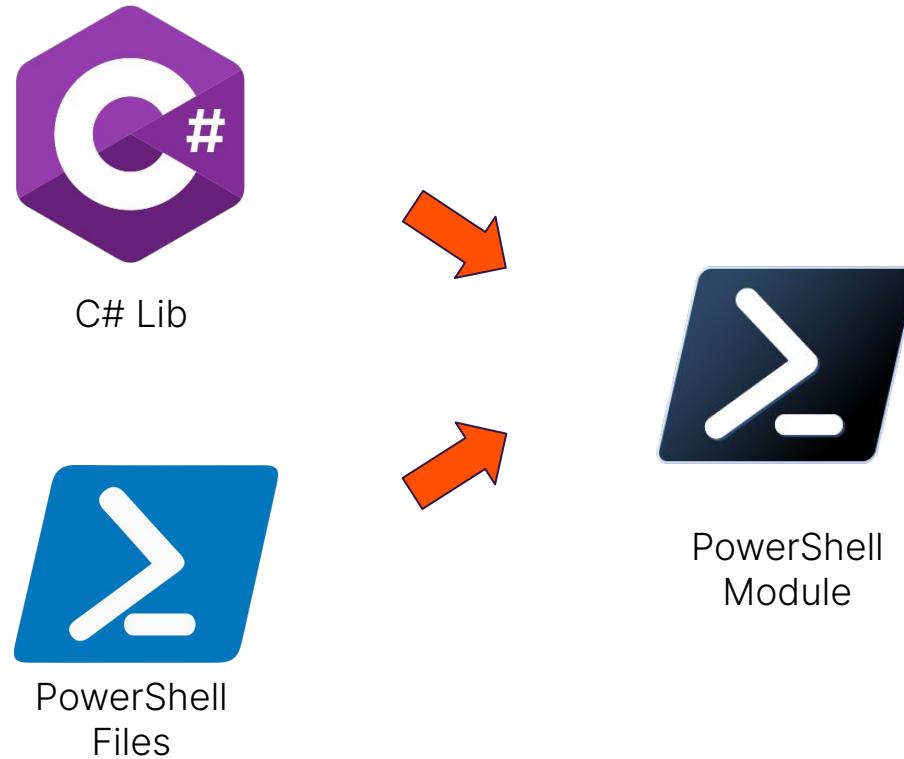
Overall Workflow

Generate PowerShell Scripts



Overall Workflow

Generate PowerShell Modules

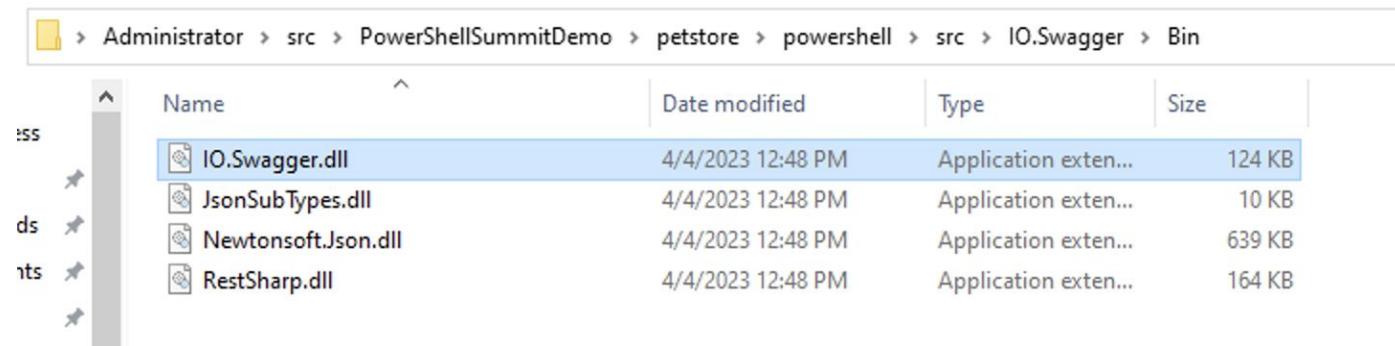


Step One - Generate C# Lib



Step One - Generate C# Lib

How does C# dll look like?



The screenshot shows a Windows File Explorer window with the following path: Administrator > src > PowerShellSummitDemo > petstore > powershell > src > IO.Swagger > Bin. The list view displays four files:

	Name	Date modified	Type	Size
ss	IO.Swagger.dll	4/4/2023 12:48 PM	Application exten...	124 KB
ds	JsonSubTypes.dll	4/4/2023 12:48 PM	Application exten...	10 KB
nts	Newtonsoft.Json.dll	4/4/2023 12:48 PM	Application exten...	639 KB
	RestSharp.dll	4/4/2023 12:48 PM	Application exten...	164 KB

Step One - Generate C# Lib

What does C# dll include?

```
C# StoreApis x
powershell > csharp > SwaggerClient > src > IO.Swagger > Api > C# StoreApi.cs

691     /// <summary>
692     /// Place an order for a pet
693     /// </summary>
694     /// <exception cref="IO.Swagger.Client.ApiException">Thrown when fails to make API call</exception>
695     /// <param name="body">order placed for purchasing the pet</param>
696     /// <returns>Order</returns>
697     public Order PlaceOrder (Order body)
698     {
699         ApiResponse<Order> localVarResponse = PlaceOrderWithHttpInfo(body);
700         return localVarResponse.Data;
701     }
702 }
```

```
C# StoreApis x
powershell > csharp > SwaggerClient > src > IO.Swagger > Api > C# StoreApi.cs

703     /// <summary>
704     /// Place an order for a pet
705     /// </summary>
706     /// <exception cref="IO.Swagger.Client.ApiException">Thrown when fails to make API call</exception>
707     /// <param name="body">order placed for purchasing the pet</param>
708     /// <returns>ApiResponse of Order</returns>
709     public ApiResponse< Order > PlaceOrderWithHttpInfo (Order body)
710     {
711         // verify the required parameter 'body' is set
712         if (body == null)
713             throw new ApiException(400, "Missing required parameter 'body'");
714
715         var localVarPath = "/store/order";
716         var localVarPathParams = new Dictionary<String, String>();
717         var localVarQueryParams = new List<KeyValuePair<String, String>>();
718         var localVarHeaderParams = new Dictionary<String, String>(this.Configuration.DefaultHeader);
719         var localVarFormParams = new Dictionary<String, String>();
720         var localVarFileParams = new Dictionary<String, FileParameter>();
721         Object localVarPostBody = null;
722
723         // to determine the Content-Type header
724         String[] localVarHttpContentTypes = new String[] {
725             "application/json"
726         };
727         String localVarHttpContentType = this.Configuration.ApiClient.SelectHeaderContentType(localVarHttpContentTypes);
728
729         // to determine the Accept header
730         String[] localVarHttpHeaderAccepts = new String[] {
```

Step One - Generate C# Lib

How do we use C# dll within PowerShell?

Create Api object from
C# Lib

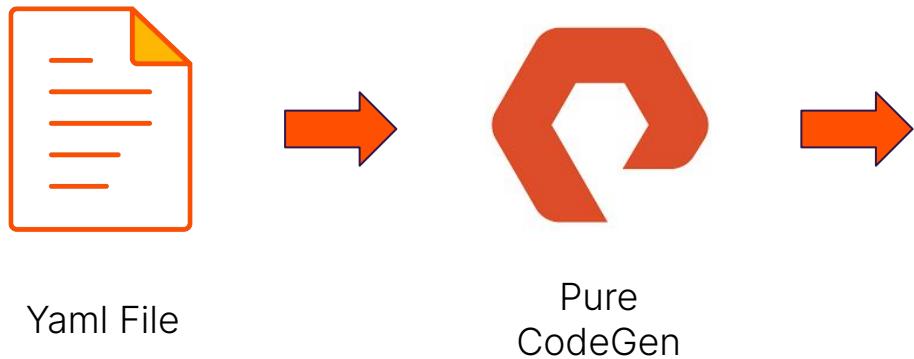
```
> IO.Swagger.psm1 X
powershell > src > IO.Swagger > > IO.Swagger.psm1 > #region Initialize APIs
16
17     #region Initialize APIs
18
19     'Creating object: IO.Swagger.Api.PetApi' | Write-Verbose
20     $Global:PetApi= New-Object -TypeName IO.Swagger.Api.PetApi -ArgumentList @($null)
21
22     'Creating object: IO.Swagger.Api.StoreApi' | Write-Verbose
23     $Global:StoreApi= New-Object -TypeName IO.Swagger.Api.StoreApi -ArgumentList @($null)
24
25     'Creating object: IO.Swagger.Api.UserApi' | Write-Verbose
26     $Global:UserApi= New-Object -TypeName IO.Swagger.Api.UserApi -ArgumentList @($null)
27
28
29     #endregion
30
```

Invoke API from
C# Lib

```
Process {
    # Reference Params
    $OrderObj = New-Order -id $Id -petid $Petid -quantity $Quantity -shipdate $Shipdate -status $Status -complete $Complete

    $Global:StoreApi.placeOrder(
        $OrderObj
    )
}
```

Step Two - Translate Yaml file into in-memory data



```
pureSwagger: <pure_swagger.PureSwagger object at 0x0000025403C50550>
  > special variables
  > function variables
param_map: {'/store/order': {'post': {...}}}
  > special variables
  > function variables
'/store/order': {'post': {'Id': {...}, 'Petid': {...}, 'Quantity': {...}, 'Shipdate': {...}, 'Status': {...}, 'Complete': {...}}}
  > special variables
  > function variables
  > 'post': {'Id': {'type': 'long', 'description': 'Order id', 'format': 'int'}, 'Petid': {'type': 'long', 'description': 'Order pet id', 'format': 'int'}, 'Quantity': {'type': 'int', 'description': 'Order quantity', 'format': 'int'}, 'Shipdate': {'type': 'string', 'description': 'date of shipping', 'format': 'date'}, 'Status': {'type': 'string', 'description': 'Order Status', 'enum': ['QUEUED', 'PENDING_PAYMENT', 'PROCESSING', 'SHIPPED', 'DELIVERED', 'REFUNDED', 'CANCELLED']}, 'Complete': {'description': 'Order completion Status', 'type': 'boolean'}}}
```

Python Directory

Step Two - Translate Yaml file into in-memory data

Resolve yaml reference

```
paths:  
  /store/order:  
    post:  
      tags:  
        - "store"  
      summary: "Place an order for a pet"  
      description: "Place an order for a pet. Full fill the order information as parameter."  
      operationId: "placeOrder"  
      consumes:  
        - "application/json"  
      produces:  
        - "application/json"  
        - "application/xml"  
      parameters:  
        - name: host  
          in: body  
          description: "order placed for purchasing the pet"  
          required: true  
      schema:  
        $ref: ./models/order.yaml  
    responses:  
      200:  
        description: "successful operation"  
      400:  
        description: "Invalid Order"
```

Reference starting from "\$ref"

```
#####  
## __resolve_ref__  
## Recursive scan of data.  
## For each "$ref" node that references another YAML file,  
## read that file in.  
## Replace this node in the tree with what was read from the YAML file.  
## Add "source_path" key to the node with the path to the YAML file.  
#####  
def __resolve_ref__(self, data, root='root', level = 0):  
    if isinstance(data, dict):  
        keys = data.keys()  
        for key in keys:  
            self.__process_ref__(data, key, root, level + 1)  
    if isinstance(data, list):  
        for key in range(len(data)):  
            self.__process_ref__(data, key, root, level + 1)
```

Function for processing reference

Step Two - Translate Yaml file into in-memory data

Mark complex parameter type

```
parameters:
- name: host
  in: body
  description: "order placed for purchasing the pet"
  required: true
  schema:
    $ref: ./models/order.yaml
```

Complex type in yaml

```
#####
# __identify_complex_params__
# Mark complex (non simple types) parameters along with the parameter metadata
#####
def __identify_complex_params__(self, data):
    for path in data['paths']:
        for action_name, action in data['paths'][path].items():
            if PARAMETERS in data['paths'][path][action_name]:
                for param in data['paths'][path][action_name][PARAMETERS]:
                    param['isComplexType'] = is_complex_type(param)
```

Identify complex type and mark it in directory

```
    <schema>: {'description': 'Order for the pet store', 'properties': {'id': {...}, 'petId': {...}, 'quantity': {...}, 'shipDate': {...}}, 'type': 'object'}
    > special variables
    > function variables
      'description': 'Order for the pet store'
    <properties>: {'id': {'type': 'long', 'description': 'Order id', 'format': 'int'}, 'petId': {'type': 'long', 'description': 'Order pet id', 'format': 'int'}, 'quantity': {'type': 'int', 'description': 'Order quantity', 'format': 'int'}, 'shipDate': {'type': 'string', 'description': 'date of shipping', 'format': 'date-time'}, 'status': {'type': 'string', 'description': 'Order Status', 'enum': ['placed', 'approved', 'delivered']}, 'complete': {'description': 'Order completion Status', 'type': 'boolean'}}
    > special variables
    > function variables
      > 'id': {'type': 'long', 'description': 'Order id', 'format': 'int'}
      > 'petId': {'type': 'long', 'description': 'Order pet id', 'format': 'int'}
      > 'quantity': {'type': 'int', 'description': 'Order quantity', 'format': 'int'}
      > 'shipDate': {'type': 'string', 'description': 'date of shipping', 'format': 'date-time'}
      > 'status': {'type': 'string', 'description': 'Order Status', 'enum': ['placed', 'approved', 'delivered']}
      > 'complete': {'description': 'Order completion Status', 'type': 'boolean'}
    len(): 6
    'type': 'object'
    'source_path': 'C:/Users/Administrator/src/PowerShellSummitDemo/petstore/powershell/src/Pure.Swagger/scripts/yaml/models/order.yaml'
    'required': False
    len(): 5
    'schemaName_actual': 'host'
    'schemaName': 'host'
    'originalName': 'host'
    'type': 'Order'
    'mapped-type': 'Order'
    'isExcluded': False
    'path': '/store/order'
    'method': 'post'
    'isComplexType': True
    len(): 14
```

Step Two - Translate Yaml file into in-memory data

Mark complex parameter type

```
Process {
    {# Resolved param initialization #}
    {% set reference_types = param_map | get_reference_type_params %}
        {# Create complex type #}
        # Reference Params
        {% for param in item['parameters'] if param['mapped-type'] is not none%}
            |  {% if param['isComplexType'] and not param['mapped-type'].startswith('List') %} 
                ${{ param['mapped-type'] }}Obj = {{ ComplexParamInitialization(param, method, path) }}
            |  {% endif %}
        
```

Jinja2 checks complex type and loop through parameters of the type

```
Process {
    # Reference Params
    $OrderObj = New-Order -id $Id -petid $Petid -quantity $Quantity -shipdate $Shipdate -status $Status -complete $Complete
```

PowerShell scripts. Parameter flattened now!

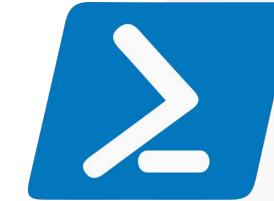
Step Three - Render Powershell files with Jinja2

```
pureSwagger: <pure_swagger.PureSwagger object at 0x0000025403C50550>
> special variables
> function variables
> param_map: {'/store/order': {'post': {...}}}
> special variables
> function variables
> '/store/order': {'post': {'Id': {...}, 'PetId': {...}, 'Quantity': {...}, 'Status': {...}, 'Complete': {...}}, 'put': {...}, 'patch': {...}, 'delete': {...}}, 'get': {...}}
> special variables
> function variables
> 'post': {'Id': {'type': 'long', 'description': 'Order id', 'format': 'int32'}, 'PetId': {'type': 'long', 'description': 'Order pet id', 'format': 'int32'}, 'Quantity': {'type': 'int', 'description': 'Order quantity', 'format': 'int32'}, 'Status': {'type': 'string', 'description': 'Order status', 'enum': ['placed', 'processing', 'shipped', 'delivered', 'canceled']}, 'Complete': {'description': 'Order completion Status', 'type': 'boolean'}}}
```

In-memory data
Python Directory



Jinja2
Template



PowerShell
Scripts

Step Three - Render Powershell scripts with Jinja2

PowerShell scripts Example

```
> PetStoreSDK-order.ps1 M X
powershell > src > Pure.Swagger > scripts > build > PetStoreSDK-order.ps1 > function New-PetStoreOrder () {
  1  # Generated from: PowerShellModule.cs.jinja2
  2
  3  # Path: /store/order
  4  0 references
  5  function New-PetStoreOrder {
  6    <#
  7    .SYNOPSIS
  8    | Place an order for a pet
  9    .DESCRIPTION
 10    | Place an order for a pet. Full fill the order information as parameter.
 11    .PARAMETER Id
 12    Order id
 13    .PARAMETER Petid
 14    Order pet id
 15    .PARAMETER Quantity
 16    Order quantity
 17    .PARAMETER Shipdate
 18    date of shipping
 19    .PARAMETER Status
 20    Order Status
 21    .PARAMETER Complete
 22    Order completion Status
 23    #>
 24
 25    Param(
 26      # id, name=Id, path:
 27      [Parameter(Mandatory=$false, ValueFromPipeline = $false, ValueFromPipelineByPropertyName = $false )]
 28      [long]
 29      $Id,
 30      # petid, name=Petid, path:
 31      [Parameter(Mandatory=$false, ValueFromPipeline = $false, ValueFromPipelineByPropertyName = $false )]
 32      [long]
 33      $Petid,
 34      # quantity, name=Quantity, path:
 35      [Parameter(Mandatory=$false, ValueFromPipeline = $false, ValueFromPipelineByPropertyName = $false )]
 36      [int]
 37      $Quantity,
 38      # shipdate, name=Shipdate, path:
 39      [Parameter(Mandatory=$false, ValueFromPipeline = $false, ValueFromPipelineByPropertyName = $false )]
 40      [string]
 41      $Shipdate,
```

Step Four - Export PowerShell module



C# Lib



PowerShell
Module



PowerShell
Scripts



Step Four - Export PowerShell module

Define psd1 file

```
> IO.Swagger.psd1 X
powershell > src > IO.Swagger > > IO.Swagger.psd1

46
47 # Minimum version of the common language runtime (CLR) required by this module. This pre
48 # CLRVersion = ''
49
50 # Processor architecture (None, X86, Amd64) required by this module
51 # ProcessorArchitecture = ''
52
53 # Modules that must be imported into the global environment prior to importing this module
54 # RequiredModules = @()
55
56 # Assemblies that must be loaded prior to importing this module
57 RequiredAssemblies = 'Bin\IO.Swagger.dll', 'Bin\JsonSubTypes.dll',
58 | | | | 'Bin\Newtonsoft.Json.dll', 'Bin\RestSharp.dll'
59
```

The new help description of the new module...

```
PS C:\Users\Administrator> Get-Help New-PetStoreOrder -Detailed

NAME
  New-PetStoreOrder

SYNOPSIS
  Place an order for a pet

SYNTAX
  New-PetStoreOrder [[-Id] <Int64>] [[-Petid] <Int64>] [[-Quantity] <Int32>] [[-Shipdate] <String>] [[-Status] <String>] [[-Complete] <Boolean>] [<CommonParameters>]

DESCRIPTION
  Place an order for a pet. Full fill the order information as parameter.

PARAMETERS
  -Id <Int64>
    Order id

  -Petid <Int64>
    Order pet id

  -Quantity <Int32>
    Order quantity

  -Shipdate <String>
    date of shipping

  -Status <String>
    Order Status

  -Complete <Boolean>
    Order completion Status

  <CommonParameters>
    This cmdlet supports the common parameters: Verbose, Debug,
    ErrorAction, ErrorVariable, WarningAction, WarningVariable,
    OutBuffer, PipelineVariable, and OutVariable. For more information, see
    about_CommonParameters (https://go.microsoft.com/fwlink/?LinkID=113216).

REMARKS
  To see the examples, type: "get-help New-PetStoreOrder -examples".
  For more information, type: "get-help New-PetStoreOrder -detailed".
  For technical information, type: "get-help New-PetStoreOrder -full".
```



Let's run the new module...

```
PS C:\Users\Administrator> New-PetStoreOrder -Id 1 -Petid 11 -Quantity 11 -Shipdate (Get-Date) -Status "Approved" -Complete $true

Status      : Approved
Id          : 1
PetId       : 11
Quantity    : 11
ShipDate   : 4/18/2023 10:53:33 AM
Complete    : True
```





Demo



Q&A



Review

1. Generate C# Lib
2. Translate YAML files into in-memory python directory
3. Render Jinja2 Template and generate Powershell scripts
4. Export module from Powershell scripts



**** Stop by our Booth ****

**Thank
You!**



Uncomplicate Data Storage, Forever