
.NET Conf China
2022
线上+线下 2022.12.03~04
开源 · 安全 · 赋能

.NET Conf China
2022

云计算SDK最佳实践-以.NET为例

郑宇
微软亚太-Azure SDK 产品经理
周峰
微软亚太-Azure SDK 资深软件工程师

云计算SDK最佳实践-以.NET为例

- 您可以从我们的分享中学到什么
- 云计算SDK是什么?
- 谁会用到云计算SDK
- 云计算SDK的分类
- 我们为什么需要云计算SDK
- 构建Azure SDK面临的挑战
- Azure SDK for .NET

云计算SDK最佳实践-以.NET为例

您可以从我们的分享中学到什么

- 超大型平台类系统的API工程实践
- 微软如何设计.NET框架类项目（造轮子）

云计算SDK最佳实践-以.NET为例

云计算SDK是什么?

传统SDK

- Windows SDK/.NET SDK/JAVA SDK
- 和操作系统或运行时的编程接口
- 二级制级别协议
- 通常都在进程内部通讯

云计算SDK

- Azure SDK, AWS SDK, 等等
- 大部分基于OpenAPI，为HTTP协议
- 跨进程通讯
- 少部分为其他通讯协议，如MQTT

云计算SDK最佳实践-以.NET为例

谁会用到云计算SDK

微软第一方工具

- Azure CLI – Python SDK
- Azure PowerShell - .NET SDK
- VSCode Azure Extension – JS SDK

微软第一方服务

- AKS(Azure Kubernates Service) – Golang SDK

第三方多云管理平台

第三方Infrastructure as Code Provider

企业内部使用

云计算SDK最佳实践-以.NET为例

云计算SDK的分类

云计算SDK最佳实践-以.NET为例

我们为什么需要云计算SDK

集成的认证服务

- SDK:直接引用认证库，适配多种认证模式
- Rest API: 需要自己想办法完成认证

更高效的HTTP Client生命周期管理

- SDK:HTTP Client对象池，自建请求管线
- Rest API:需要自己管理HTTP Client对象生命周期，没有原生请求管线支持

统一的异常处理

- SDK:预定义的HTTP Status Code 到异常的映射，符合服务端对客户端行为的预设
- Rest API: 需要自己来判断返回的HTTP Status Code是否代表异常

高耗时操作处理（Long running operation）

- SDK:自动处理高耗时操作
- Rest API: 需要手动处理高耗时操作的结果拉取

开发者友好

- SDK：面向对象的设计，各个语言IDE均可提供智能提示
- Rest API: 非面向对象，无智能提示

云计算SDK最佳实践-以.NET为例

构建Azure SDK面临的挑战

- Azure服务众多，200+的Azure服务
- OpenAPI本身的局限性
- 服务版本更新频繁
- 上万份的swagger spec，如何保证不同团队写出来的swagger都符合一定的规范
- 五大主流语言的支持（.NET,JAVA,JS,PYTHON,GO）,要符合该语言使用者的习惯，但并不是每个Azure服务的开发团队都有相应语言的工程师
- 文档，Sample Code等的整个开发者生态建设

云计算SDK最佳实践-以.NET为例

我们如何解决这些问题

x-ms-extension 来拓展原生OpenAPI的局限

- x-ms-pageable
- x-ms-long-running-operation
- x-ms-examples
- autorest/readme.md at main · Azure/autorest (github.com)

Swagger Quality

- Linter rule
- Swagger correctness

Swagger版本的管理

Swagger Breaking Change Detection

- Swagger PR Pipeline

SDK Breaking Change Detection

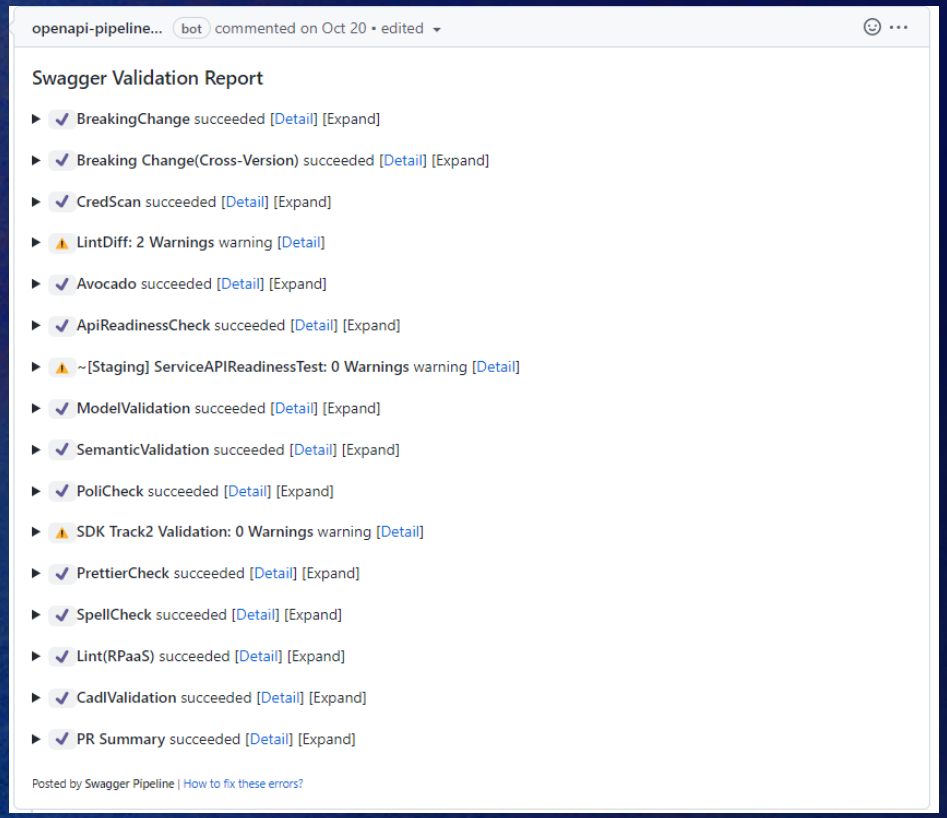
- API View

上下游生态建设

- 样例代码随SDK生成自动生成
- 文档Reference随Build Pipeline自动生成

云计算SDK最佳实践-以.NET为例

我们如何解决这些问题-示例 Swagger校验
十月份提交的Azure Cost Management Service 新版本API以及相应SDK的Build Pipeline
[CosmosDB] DTS: Add CosmosDBMongo enum by niteshvijay1995 · Pull Request #21199 · Azure/azure-rest-api-specs (github.com)

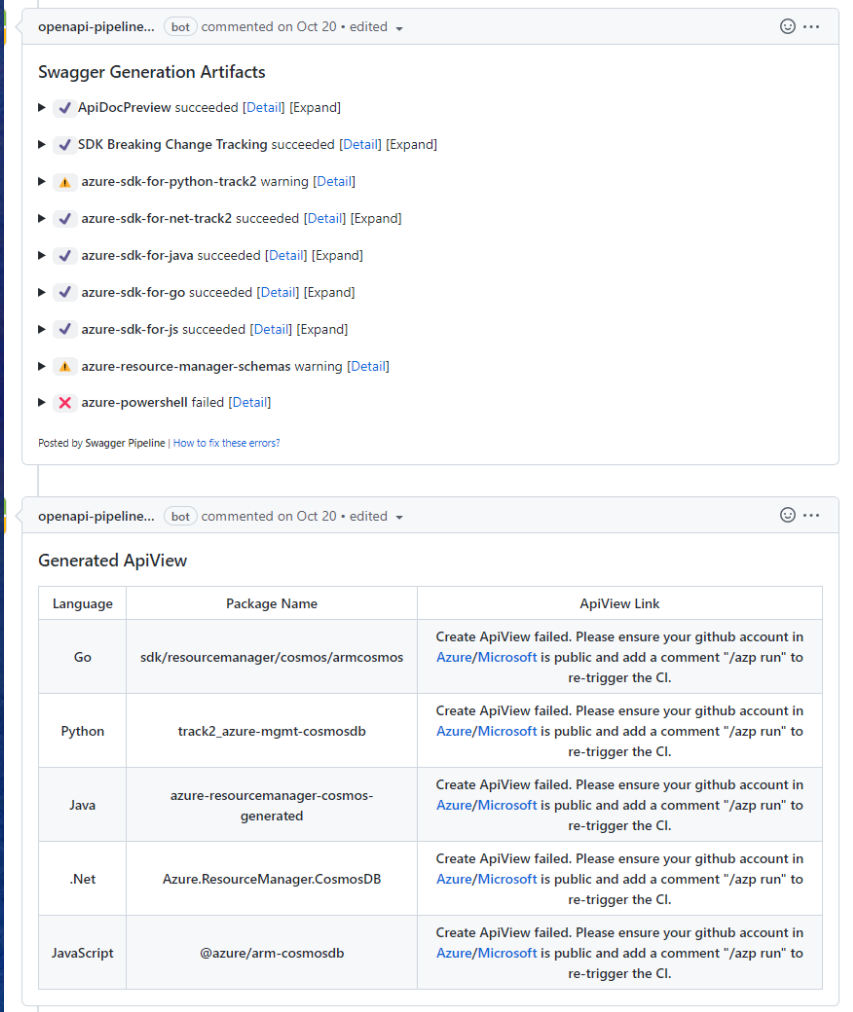


这个构建管线做了什么？

- 检查了Swager是否有Breaking change(全量，增量检查)
- 检查了PR中是否不小心包括了敏感的身份凭证等信息
- LintRule, 检查了Swagger设计是否符合Rest API的设计原则，比如 A PUT operation request body schema should be the same as its 200 response schema, to allow reusing the same entity between GET and PUT. If the schema of the PUT request body is a superset of the GET response body, make sure you have a PATCH operation to make the resource updatable. Operation: 'DataTransferJobs_Create' Request Model: 'CreateJobRequest' Response Model: 'DataTransferJobGetResults'
- 拼写检查
- Swagger本身的语法
- Swagger文件之间互相引用是否正确

云计算SDK最佳实践-以.NET为例

我们如何解决这些问题-示例 SDK校验



openapi-pipeline... bot commented on Oct 20 • edited

Swagger Generation Artifacts

- ApiDocPreview succeeded [Detail] [Expand]
- SDK Breaking Change Tracking succeeded [Detail] [Expand]
- azure-sdk-for-python-track2 warning [Detail]
- azure-sdk-for-net-track2 succeeded [Detail] [Expand]
- azure-sdk-for-java succeeded [Detail] [Expand]
- azure-sdk-for-go succeeded [Detail] [Expand]
- azure-sdk-for-js succeeded [Detail] [Expand]
- azure-resource-manager-schemas warning [Detail]
- azure-powershell failed [Detail]

Posted by Swagger Pipeline | How to fix these errors?

openapi-pipeline... bot commented on Oct 20 • edited

Generated ApiView

Language	Package Name	ApiView Link
Go	sdk/resourcemanager/cosmos/armcosmos	Create ApiView failed. Please ensure your github account in Azure/Microsoft is public and add a comment "/azp run" to re-trigger the CI.
Python	track2_azure-mgmt-cosmosdb	Create ApiView failed. Please ensure your github account in Azure/Microsoft is public and add a comment "/azp run" to re-trigger the CI.
Java	azure-resourcemanager-cosmos-generated	Create ApiView failed. Please ensure your github account in Azure/Microsoft is public and add a comment "/azp run" to re-trigger the CI.
.Net	Azure.ResourceManager.CosmosDB	Create ApiView failed. Please ensure your github account in Azure/Microsoft is public and add a comment "/azp run" to re-trigger the CI.
JavaScript	@azure/arm-cosmosdb	Create ApiView failed. Please ensure your github account in Azure/Microsoft is public and add a comment "/azp run" to re-trigger the CI.

这个构建管线做了什么

- 各个语言的SDK自动在Swagger PR管线中生成
- 检测新生成的SDK是否包含Breaking Change (这将决定我们在Release SDK的时候是否升大版本号)
- 运行测试用例测试SDK，覆盖不同平台
- 生成API View来对生成的SDK进行Review
- 生成PowerShell组件

云计算SDK最佳实践-以.NET为例

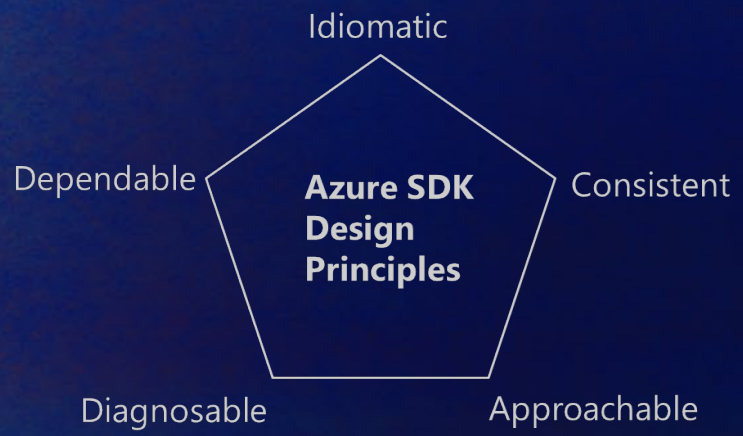
Azure SDK for .NET - 构成

基础SDK	数据交互类SDK	资源管理类SDK
Azure.Core	Azure.Messaging.ServiceBus	Azure.ResourceManager.ServiceBus
Azure.Identity	Azure.Analytics.Purview.Administration	Azure.ResourceManager.Compute
Azure.ResourceManager	Azure.Analytics.Synapse.Artifacts	Azure.ResourceManager.Storage
	其他服务的数据SDKs...	其他服务的管理SDKs...

https://azure.github.io/azure-sdk/releases/latest/all/dotnet.html

云计算SDK最佳实践-以.NET为例

Azure SDK for .NET - 设计原则



Idiomatic 符合语言习惯

- 遵循.NET框架设计指南
- SDK的设计，版本管理遵循.NET标准库规范
- 拥抱.NET生态系统，包括它的优势与缺陷

Consistent 一致性

- 所有.NET SDK的一致性>客户端与服务端的一致性>所有语言SDK的一致性
- 所有Azure SDK就像是同一个组的同一系列产品，而不是一堆无关的NuGet包
- 用户只需第一次学习公共的概念，然后可以将其应用到所有SDK

Approachable 易用性

- 使用少量步骤即可入门；同时为高级用户提供强大的配置功能
- 使用少量的概念，类型和成员
- 用户可以轻松找到出色的入门指南和示例

Diagnosable 可诊断

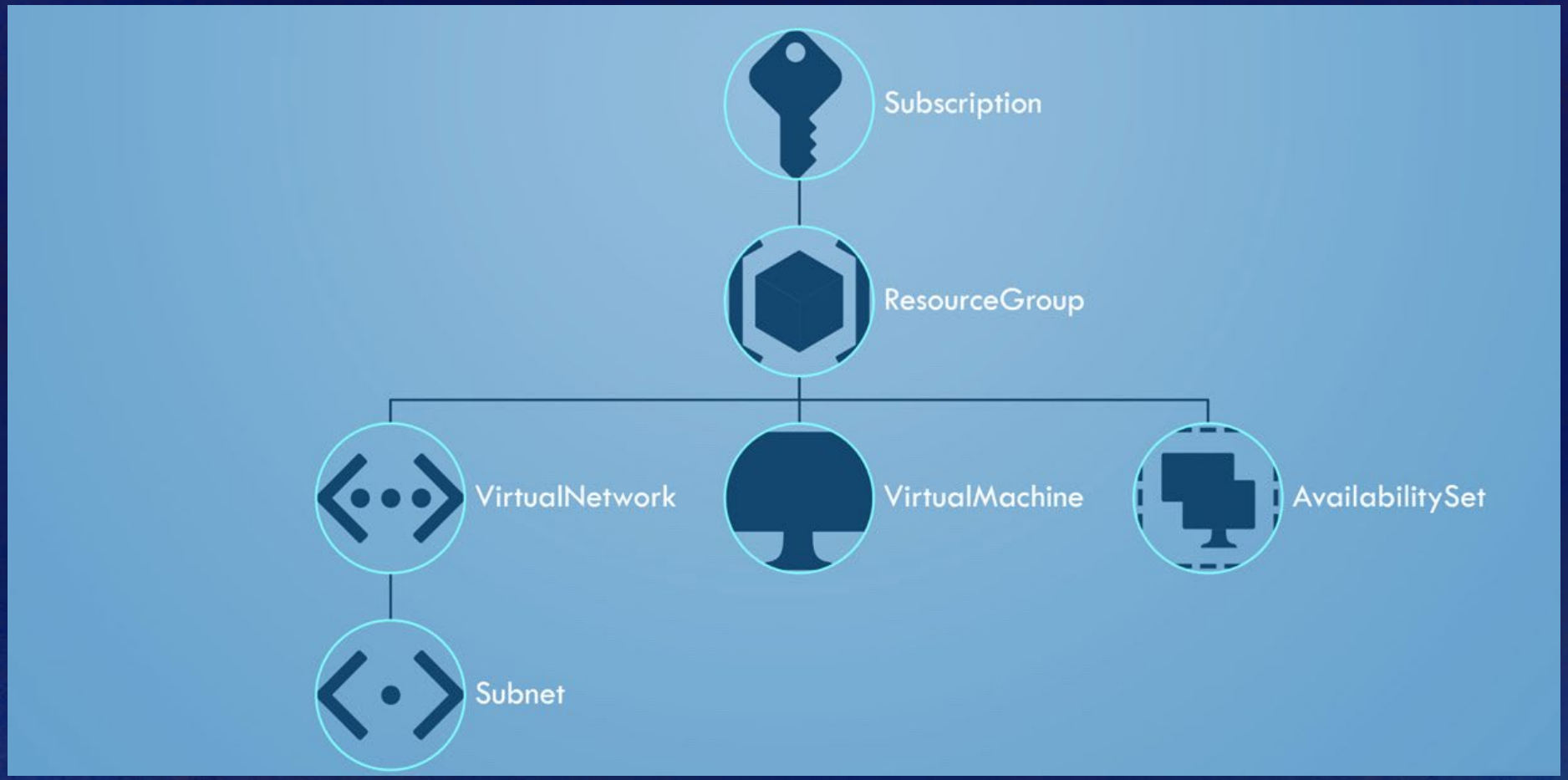
- 可靠的日志记录，链路追踪，错误消息

Dependable 可靠性

- 100 % 向后兼容
- 可预测的SDK支持生命周期，功能覆盖及质量

云计算SDK最佳实践-以.NET为例

Azure SDK for .NET - 资源层级



云计算SDK最佳实践-以.NET为例

Azure SDK for .NET -资源层级的结构化表示

```
string vmId = "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/myrg/providers/Microsoft.Compute/virtualMachines/myvm";
ResourceIdentifier vmResourceId = new ResourceIdentifier(vmId);

// Structured data
Console.WriteLine(vmResourceId.Name);              // myvm
Console.WriteLine(vmResourceId.ResourceGroupName); // myrg
Console.WriteLine(vmResourceId.SubscriptionId);    // 00000000-0000-0000-0000-000000000000
Console.WriteLine(vmResourceId.ResourceType);      // Microsoft.Compute/virtualMachines

// Hierarchical data
Console.WriteLine(vmResourceId.Parent.Name);                // myrg
Console.WriteLine(vmResourceId.Parent.ResourceType);        // Microsoft.Resources/resourceGroups
Console.WriteLine(vmResourceId.Parent.Parent.Name);         // 00000000-0000-0000-0000-000000000000
Console.WriteLine(vmResourceId.Parent.Parent.ResourceType); // Microsoft.Resources/subscriptions
```

云计算SDK最佳实践-以.NET为例

Azure SDK for .NET - 资源的类表示

类	描述
[Resource]Data	数据模型
[Resource]Resource	包含资源Id，资源数据，操作单个资源的方法, 获取子类资源Collection的方法
[Resource]Collection	包含父资源的Id以及该父资源的子类资源的集体操作方法

云计算SDK最佳实践-以.NET为例

Azure SDK for .NET - 两代SDK代码示例与对比

Track 2:

```
ArmClient client = new ArmClient(new DefaultAzureCredential());

SubscriptionCollection subscriptionCollection = client.GetSubscriptions();
SubscriptionResource subscriptionResource = await subscriptionCollection.GetAsync("00000000-0000-0000-0000-000000000000");

ResourceGroupCollection rgCollection = subscriptionResource.GetResourceGroups();
ResourceGroupResource rgResource = (await rgCollection.CreateOrUpdateAsync(WaitUntil.Completed, "myrg", new
ResourceGroupData(AzureLocation.WestUS))).Value;

VirtualMachineCollection vmCollection = rgResource.GetVirtualMachines();
var vmData = new VirtualMachineData(AzureLocation.WestCentralUS){...};
VirtualMachineResource vmResource = (await vmCollection.CreateOrUpdateAsync(WaitUntil.Completed, "myvm", vmData)).Value;
await vmResource.PowerOffAsync(WaitUntil.Started);
```

Track 1:

```
var resourceClient = new ResourcesManagementClient("00000000-0000-0000-0000-000000000000", credentials);
var resourceGroups = resourceClient.ResourceGroups;
var resourceGroup = new ResourceGroup("westus");
resourceGroup = await (await resourceGroups.CreateOrUpdateAsync("myrg", resourceGroup)).WaitForCompletionAsync();

var computeManagementClient = new ComputeManagementClient(subscriptionId, credential);
var virtualMachines = computeManagementClient.VirtualMachines;
var vm = new VirtualMachine(location){...};
vm = await (await virtualMachines.StartCreateOrUpdateAsync("myrg", "myvm", vm)).WaitForCompletionAsync();
await virtualMachines.StartPowerOffAsync("myrg", "myvm");
```

Track 2 特点

- 只需创建一次客户端
- 资源对象的方法自带来源于对象属性的默认参数值
- 资源方法的返回对象是资源对象而不是数据模型

云计算SDK最佳实践-以.NET为例

Azure SDK for .NET - 管理现存资源优化前后代码示例与对比

任务：删除一个现有的虚拟机

```
ArmClient client = new ArmClient(new DefaultAzureCredential());
SubscriptionCollection subscriptionCollection = client.GetSubscriptions();
SubscriptionResource subscriptionResource = await subscriptionCollection.GetAsync("00000000-0000-0000-0000-000000000000"); // API call
ResourceGroupCollection rgCollection = subscriptionResource.GetResourceGroups();
ResourceGroupResource rgResource = await rgCollection.GetAsync("myrg"); // API call
VirtualMachineCollection vmCollection = rgResource.GetVirtualMachines();
VirtualMachineResource vmResource = await vmCollection.GetAsync("myvm"); // API call
await vmResource.DeleteAsync(WaitUntil.Completed); // API call
```

优化后：

```
ArmClient client = new ArmClient(new DefaultAzureCredential());
ResourceIdentifier vmResourceId = VirtualMachineResource.CreateResourceIdentifier("00000000-0000-0000-0000-000000000000", "myrg", "myvm");
VirtualMachineResource vmResource = client.GetVirtualMachineResource(vmResourceId);
await vmResource.DeleteAsync(WaitUntil.Completed); // API call
```

云计算SDK最佳实践-以.NET为例

Azure SDK for .NET - 资源层级设计遇到的挑战

非资源节点	位于资源层级中，但其本身并非资源，例如location
Singleton资源	单例资源，且资源名固定，通常为default, current, latest等
Extension资源	可以挂在多个父资源下
Scope资源	父资源通常为（限定条件下）的任意资源

云计算SDK最佳实践-以.NET为例

Azure SDK for .NET - 公共类的替换

概念

不同服务会使用到一些公共的资源相关或功能相关的数据模型, 它们可以被提取为公共类供不同服务使用。SDK生成时需要将由本服务swagger生成的对应类替换成公共类。

来源

基于swagger公共类定义, 例如TrackedResource, ErrorResponse，ManagedServiceIdentity

方法

通过递归比较类的属性, 除可选匹配属性外，其余属性的名称和类型全部匹配成功，即可替换为公共类。

挑战

如何处理不同版本的公共类定义?

观察

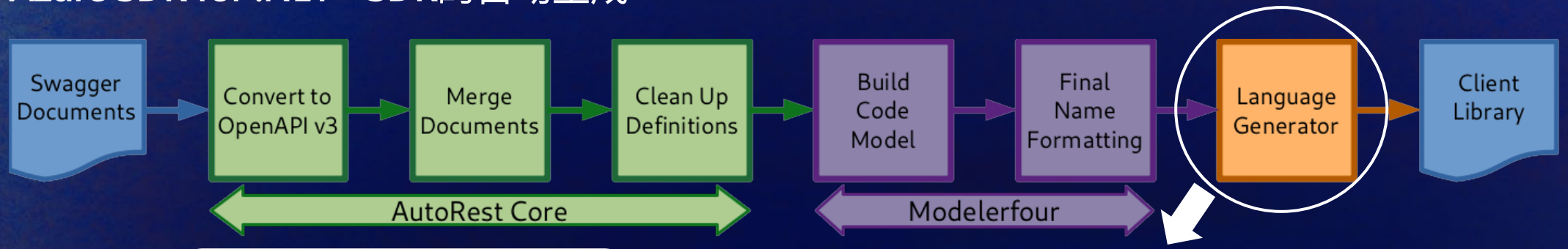
- v1 => v2: 重命名 ErrorResponse 为 ErrorDetail；添 加新的类ErrorResponse；添 加了其他新的类；
- v2 => v3: Resource类添加 systemData属性；

方案

- 将 v1、v2、v3 的定义合并为一组向后兼容的通用类型，也即使用v3即可。对于v3才增加的只读属性，在属性匹配时作为可选项，使公共类中的v3可替换掉被匹配的v1或v2，只是相应只读属性永远为null。

云计算SDK最佳实践-以.NET为例

Azure SDK for .NET - SDK的自动生成



前期处理

- 反序列化构建autorest pipeline 传过来的类，例如code model, 配置项
- 读入已有的代码

生成原始代码

- 根据code model生成中间态的内部模型
- 写入原始代码

后期处理

- 调整访问控制(Internalizer)，并删除未用到的类（Remover）
- 根据手写代码删除相应的自动生成的成员
- 简化并设定格式

云计算SDK最佳实践-以.NET为例

Azure SDK for .NET - 生成代码的定制

Swagger directives：

在生成代码之前对Swagger进行修改。
通用于所有语言，功能灵活。

.NET SDK生成器的配置项：

可作用于.NET SDK生成阶段的任意时期。
可根据需要增加配置，语法相对简洁。

示例

```
directive:
  - from: QueryPackQueries.json
    where: $.definitions.LogAnalyticsQueryPackQueryProperties
    transform: >
      $.properties.id.format = 'uuid';
      $.properties.id['x-ms-client-name'] = 'ApplicationId';
```

```
rename-mapping:
  LogAnalyticsQueryPackQuery.properties.id: ApplicationId|uuid
```

```
directive:
  - from: swagger-document
    where: $.definitions..enabled
    transform: >
      if ($['type'] === 'boolean')
        $['x-ms-client-name'] = 'IsEnabled'
```

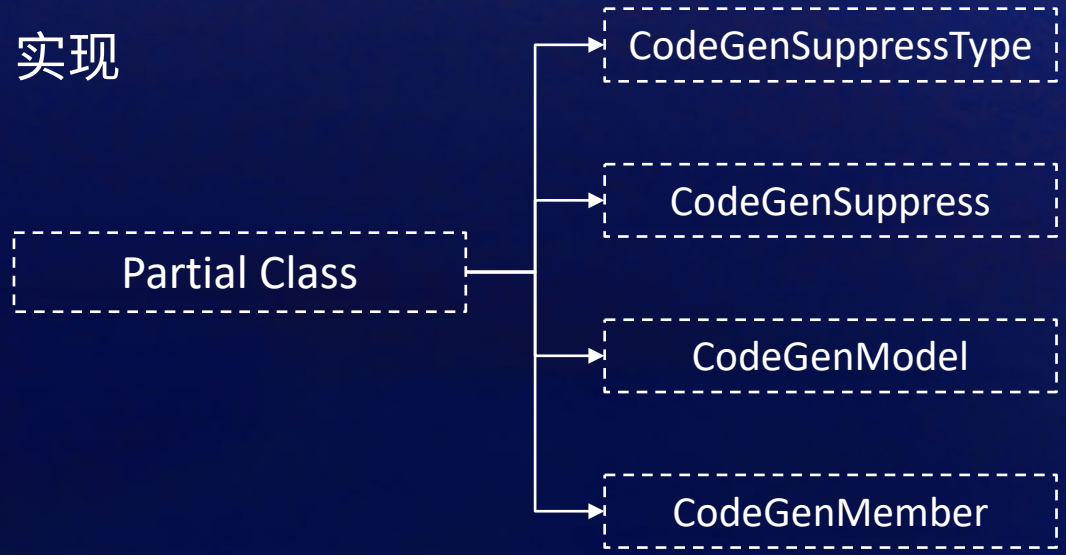
```
format-by-name-rules:
  'tenantId': 'uuid'
  'location': 'azure-location'
override-operation-name:
  DeletedWorkspaces_List: GetDeletedWorkspaces
  DeletedWorkspaces_ListByResourceGroup: GetDeletedWorkspaces
```

云计算SDK最佳实践-以.NET为例

Azure SDK for .NET - 手写代码支持

应用场景

实现



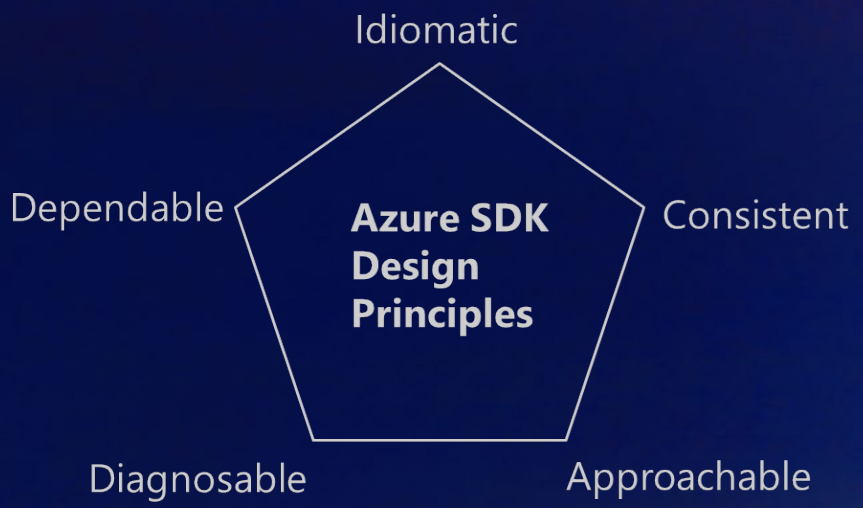
示例

```
1  [assembly:CodeGenSuppressType("ModelToBeSkipped")]
2
3  namespace CustomNamespace
4  {
5      [CodeGenModel("Model")]
6      [CodeGenSuppress("CustomizedModel", typeof(CustomFruitEnum), typeof(CustomDaysOfWeek))]
7      internal partial class CustomizedModel
8      {
9
10         [CodeGenMember("ModelProperty")]
11         public int? CustomizedModelProperty { get; set; }
12     }
13 }
```

云计算SDK最佳实践-以.NET为例

Azure SDK for .NET - 设计总结

- 结构化资源id (**易用性**)
- 资源的对象化层级结构(**符合语言习惯**)
- 单客户端(**一致性**)
- 方法自带默认值(**易用性**)
- 公共类替换(**一致性/易用性**)
- 手写代码支持向后兼容(**可靠性**)



Thank you!

Let's build amazing apps with .NET 7

get.dot.net/7