

# API First在构建云服务 Python SDK中的实践

石辰杰 微软开发平台事业部高级软件工程师



# 面临的挑战

Azure SDK团队需要为所有的Azure云服务提供Python SDK

- 规模
  - 云服务数量在不断增长，服务由不同团队维护，Python SDK的发布如何完全自动化？
- 规范
  - 对于API的审查落后于服务开发，Azure API规范，Azure Python SDK规范如何落地？
- 一致性
  - 不同服务间的、同一服务不同版本的Python SDK一致性如何保证？
  - 同一服务Python SDK如何和其他语言的SDK保持一致性？



# 过去的解决方案

## 围绕OpenAPI的工具链

- 生成Python SDK，检查Python SDK可能遇到的问题
  - 和规范的一致性
  - 是否有SDK Breaking Change

## 无法完全解决前述挑战

- OpenAPI的限制
  - 无法从Spec层面定义Python SDK的细节
- 事后审查的弊端
  - 遇到的Issue大部分都来源于服务API的定义问题



# API First w/TypeSpec

TypeSpec: 一种可扩展的API定义语言

- 语法很简单，扩展很灵活
  - 定义各种协议的API
  - 定义生成的客户端SDK风格
  - 模板化&模块化
  - Linter检查
- Playground Demo
  - [Rest](#), [Protobuf](#)

通过TypeSpec让API First在Azure落地

```
main.tsp

import "@typespec/http";

using TypeSpec.Http;

model Store {
  name: string;
  address: Address;
}

model Address {
  street: string;
  city: string;
}

@route("/stores")
interface Stores {
  list(@query filter: string): Store[];
  read(@path id: Store): Store;
}
```



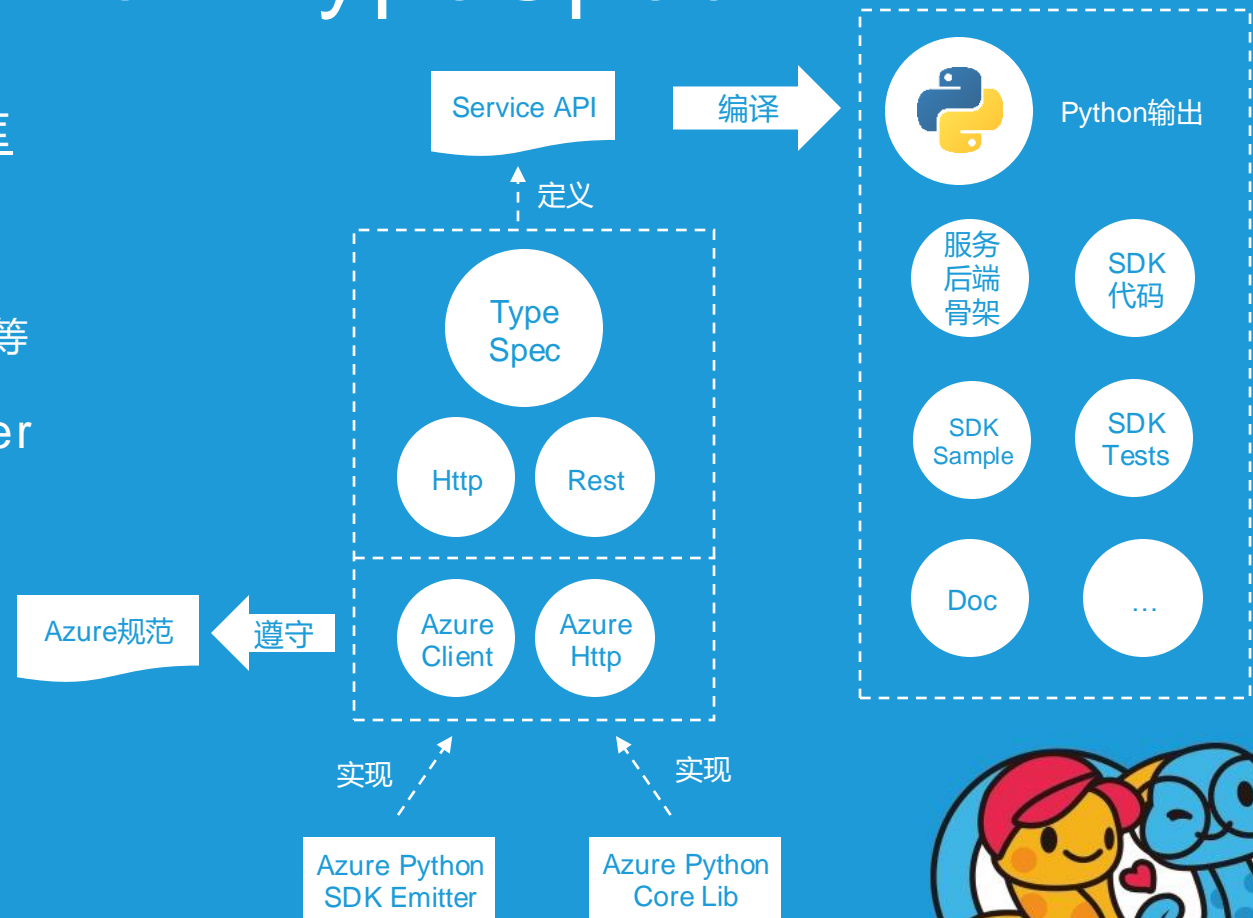
# Azure Python SDK on TypeSpec

Azure相关的TypeSpec库和Python库

- 落实Azure服务相关规范
  - 认证、通用资源操作、分页操作、LRO等

基于TypeSpec的Python SDK Emitter

- 生成SDK代码
- 生成示例、测试、文档等

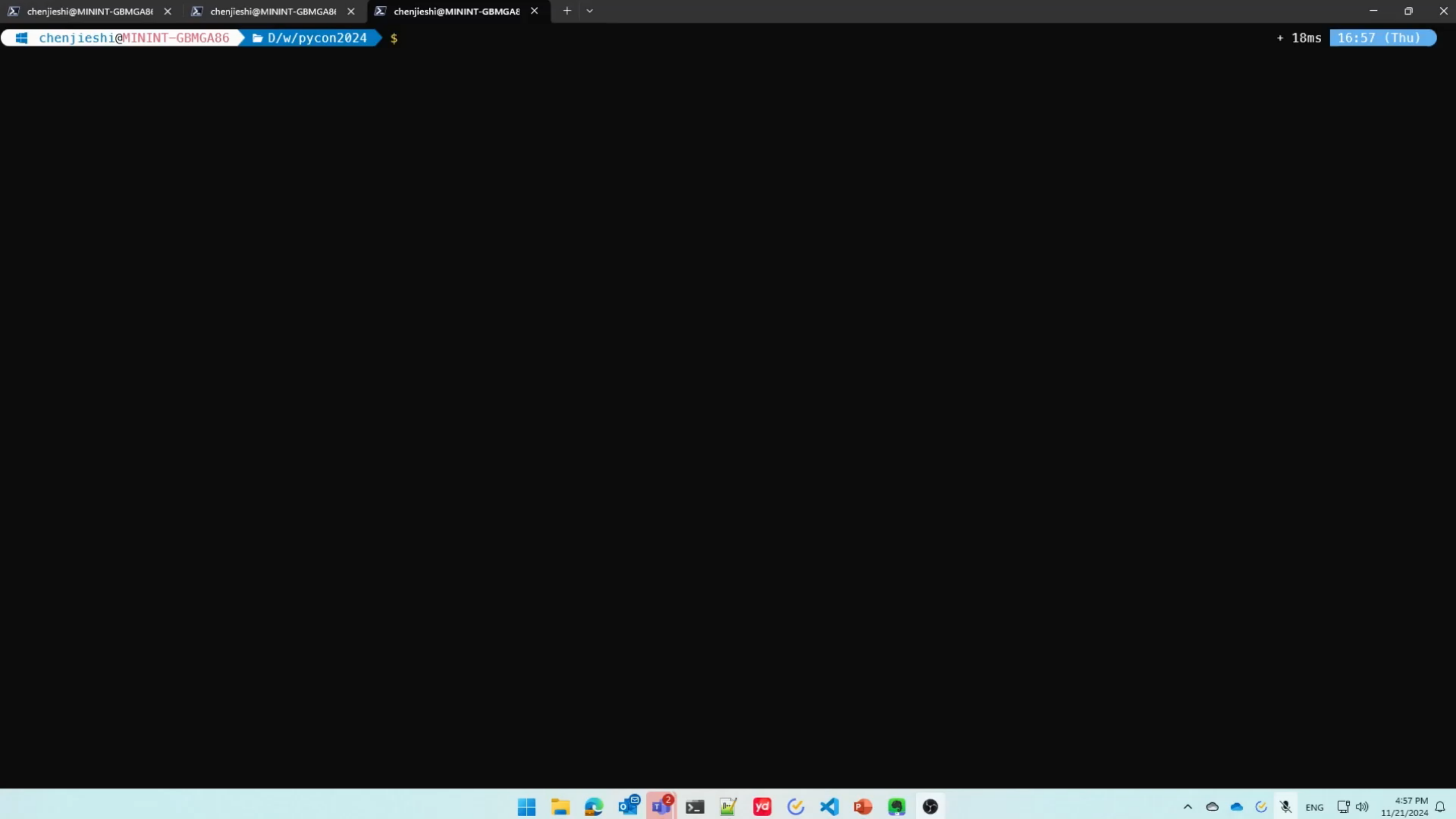


# Demo

## Azure AI Face服务

- API定义
- 生成Python SDK
  - `tsp compile client.tsp --emit "@azure-tools/typespec-python"`
  - 源码链接, Pypi链接
- 调用API
  - 示例链接





# 相关资源

- [Azure SDK Blog](#)
- Azure SDK for Python repo: [Azure/azure-sdk-for-python](#)
- TypeSpec官网: [typespec.io](#)
- Playground: [aka.ms/trytypespec](#)
- TypeSpec repo: [microsoft/typespec](#)





# TypeSpec介绍

# TypeSpec

