API设计与实现

主讲人：陈长兵

目 录

CONTENTS

7 API实现-基于REST

- REST基础
- REST API设计标准
- 基于Swagger的REST API实战

7.1 REST基础

REST起源

- Representational State Transfer（表述性状态转移）
  - Roy Fielding，2000年博士学位论文
  - Architectural Styles and the Design of Network-based Software Architectures

7.1 REST基础

REST作用

- REST是Web的架构风格
  - 指导Web架构的设计与开发
  - 关注组件的可伸缩性，接口的通用性，降低延迟等

7.1 REST基础

软件架构相关术语

- 运行时抽象
  - 软件架构是软件系统在其运行过程中某个阶段的运行时元素的抽象

7.1 REST基础

软件架构相关术语

- 架构元素
  - 组件
  - 连接器
  - 数据

7.1 REST基础

软件架构相关术语

- 配置
  - 软件系统在运行期间和组件、连接器和数据之间的架构关系的结构
  - 如何组织架构元素之间的关系

7.1 REST基础

软件架构相关术语

- 架构属性
  - 软件架构对组件、连接器和数据的选择以及排列所产生的属性
  - 系统的功能性属性以及非功能性属性

7.1 REST基础

软件架构相关术语

- 架构风格
  - 架构风格是一组相互协作的架构约束，这些架构约束限制了架构元素的角色和功能，以及在任何一个遵循该架构风格的架构中允许存在的元素之间的关系。
  - 架构风格是一种对架构进行分类并且定义它们的公共特征的机制。

7.1 REST基础

基于网络应用的关键架构属性

- 性能
- 可伸缩性
- 简单性
- 可修改性
- 可见性
- 可移植性
- 可靠性

7.1 REST基础

REST架构风格约束

- 客户端-服务器
  - 分离关注点
  - 用户界面 和 数据存储
  - 独立部署

7.1 REST基础

REST架构风格约束

- 无状态
  - 客户端到服务的每个请求，必须包含理解改请求所必须的所有信息
  - 会话状态全部保存在客户端

7.1 REST基础

REST架构风格约束

- 缓存
  - 通过缓存架构约束，改善网络效率
  - 消除一部分网络交互，

7.1 REST基础

REST架构风格约束

- 统一接口
  - 强调组件之间要有一个统一的接口
  - 简化整体的系统架构，改善交互的可见性

7.1 REST基础

REST架构风格约束

- 分层系统
  - 通过限制组件的行为，将架构分解成若干层级
  - 每个组件只能看到与其交互的相邻层

7.1 REST基础

REST架构风格约束

- 按需代码
  - 允许下载与执行部分代码，对客户端的功能进行扩展
  - 可选的加工约束

7.1 REST基础

REST架构元素

- 数据元素
  - 资源
  - 资源描述符
  - 表述
  - 表述元数据
  - 资源元数据
  - 控制数据

7.1 REST基础

REST架构元素

- 连接器
  - 客户端
  - 服务器
  - 缓存
  - 解析器
  - 隧道

7.1 REST基础

REST架构元素

- 组件
  - 源服务器
  - 网关
  - 代理
  - 用户代理

7.1 REST基础

什么是HATEOAS

- Hypermedia As The Engine Of Application State
  - 超媒体作为应用状态的引擎，超文本驱动
  - 超媒体不仅包括数据，还包含状态迁移的语义
  - 以超媒体作为引擎，驱动Web应用的状态迁移

7.1 REST基础

REST成熟度

- Richardson Maturity Model
  - Leonard Richardson，2008 Qcon大会
  - 最初的REST只是一种架构风格，不是规范，缺少直接参考的依据
  - 将REST的实现划分不同的等级

7.1 REST基础

REST成熟度

- Level0
  - 使用HTTP作为传输方式
  - RPC的一种具体形式，SOAP和XML-RPC属于此级别
  - 表现形式：一个URI，一个HTTP方法

7.1 REST基础

REST成熟度

- Level1
  - 引入资源的概念，每个资源都有对应的标识符和表达
  - 分而治之
  - 表现形式：多个URI，一个HTTP方法

7.1 REST基础

REST成熟度

- Level2
  - 根据语义使用HTTP动词
  - 使用不同的HTTP方法来进行不同的操作，并使用HTTP状态码来表示不同的结果
  - 使用相同的方法处理类似的情况
  - 表现形式：多个URI，多个HTTP方法

7.1 REST基础

REST成熟度

- Level3
  - 使用HATEOAS
  - 在资源的表述中包含了链接信息，客户端可根据链接发现可执行的动作
  - 引入可发现性，提供一种使协议更具有描述性的方法
  - RESTful的终极状态

7.1 REST基础

REST与HTTP协议

- HTTP/0.9
  - 1991年，Timothy Berners-Lee
- HTTP/1.0
  - 1996年，rfc1945，Berners-Lee，Roy Fielding，...
- HTTP/1.1
  - 1997年，rfc2068，Roy Fielding，...，Berners-Lee
  - 1999年，rfc2616，Roy Fielding，...，Berners-Lee

7.1 REST基础

REST与HTTP协议

- HTTP/0.9
  - 基于客户端服务端的请求响应协议
  - 没有首部，没有状态码
  - 只支持GET方法
  - ...

7.1 REST基础

REST与HTTP协议

- HTTP/1.0升级
  - 增加首部
  - 增加16个响应状态码
  - 更多的请求方法：GET、HEAD、POST
  - ...

7.1 REST基础

REST与HTTP协议

- HTTP/1.1升级
  - 新增更多请求方法
  - 强化缓存管理和控制
  - ...

7.1 REST基础

REST与HTTP协议

- 从Fielding视角来看HTTP协议
  - HTTP不是RPC
  - HTTP不是传输协议
  - ...

7.2 REST设计标准

三大设计标准

- Swagger
  - API Development for Everyone
- RAML
  - A simple but powerful syntax for modelling APIs
- API Blueprint
  - A powerful high-level API description language for web APIs

7.2 REST设计标准-Swagger设计标准

Swagger介绍

- 最早出现、最成熟的API标准规范。
- 官网，https://swagger.io
- 2015年被SmartBear公司收购，Swagger规范重命名为OpenAPI规范(OAS)
- OAS3.0规范，https://swagger.io/specification/

7.2 REST设计标准-Swagger设计标准

Swagger版图

- OpenAPI Specification
  - RESTful API设计的工业标准
- Open Source Tools
  - 用于OpenAPI定义的创建、更新和分享的开源工具集
- SwaggerHub
  - 支持OpenAPI规模化工作流的商业化平台方案

7.2 REST设计标准-Swagger设计标准

Swagger商业版

◆ API开发生命周期覆盖

阶段	开源版本	商户化版本
API设计	Swagger Editor	SwaggerHub
API开发	Swagger Codegen	SwaggerHub
API文档	SwaggerUI	SwaggerHub
API测试		SwaggerHub/ ReadyAPI
API Mock		ReadyAPI Virtualization
API治理		SwaggerHub
API监控		AlertSite

7.2 REST设计标准-RAML设计标准

RAML介绍

- REST API Modeling Language，REST API建模语言
- 官网，https://raml.org
- MuleSoft、PayPal、思科等业界知名公司组的强力支持

7.2 REST设计标准-RAML设计标准

RAML生态

- RAML Specification
- 社区项目
  - raml-design
  - raml-document
  - raml-build
  - raml-parser
  - raml-test
  - raml-utilities

7.2 REST设计标准-RAML设计标准

RAML生态

- RAML Specification 1.0版本
  - 使用YAML1.2规范

```
#%RAML 1.0
title: Mobile Order API
baseUri: http://localhost:8081/api
version: 1.0

uses:
  assets: assets.lib.raml

annotationTypes:
  monitoringInterval:
    type: integer

/orders:
  displayName: Orders
  get:
    is: [ assets.paging ]
    (monitoringInterval): 30
    description: Lists all orders of a specific user
    queryParameters:
      userId:
        type: string
        description: use to query all orders of a user
  post:
  /{orderId}:
    get:
      responses:
        200:
          body:
            application/json:
              type: assets.Order
            application/xml:
              type: !include schemas/order.xsd
```

Name your API, specify its version and base URL

Specify reusable types to avoid duplication and redundancy

Model your endpoints with access information, HTTP verbs, parameters, example responses and more

Model multiple response types including JSON & XML within a single interface

7.2 REST设计标准-RAML设计标准

RAML生态

- raml-design工具
  - https://raml.org/developers/design-your-api
  - API Workbench（deprecated）
  - API Designer, https://github.com/mulesoft/api-designer

7.2 REST设计标准-RAML设计标准

RAML生态

- raml-build工具
  - https://raml.org/developers/build-your-api
  - NodeJS
  - Java
  - .NET
  - Python

7.2 REST设计标准-RAML设计标准

RAML生态

- raml-test工具
  - https://raml.org/developers/test-your-api
  - 测试类工具，Abao/Vigia/Postman
  - 测试类平台，API Fortress/API Science/SmartBear

7.2 REST设计标准-RAML设计标准

RAML生态

- raml-document工具
  - https://raml.org/developers/document-your-api
  - API Console
  - RAML to HTML
  - RAML2HTML for PHP

7.2 REST设计标准-API Blueprint设计标准

API Blueprint介绍

- 是为Web API设计的一种强大的高级API设计语言
- 官网，https://apiblueprint.org/
- Apiary公司在支持，旨在帮助构建和记录API

7.2 REST设计标准-API Blueprint设计标准

API Blueprint介绍

- 是为Web API设计的一种强大的高级API设计语言
- 官网，https://apiblueprint.org/
- Apiary公司在支持，旨在帮助构建和记录API

7.2 REST设计标准-API Blueprint设计标准

API Blueprint生态

- API Blueprint Specification
- 社区项目
  - Editors
  - Testing
  - Parsers
  - MockServers
  - Renderers
  - Converters

7.2 REST设计标准-API Blueprint设计标准

API Blueprint生态

- API Blueprint Specification
  - 使用Markdown语言
  - MSON：Markdown Syntax for Object Notation
  - 定义关键字：Request / Response / Parameters / Attributes / Model / Data Structures等

7.2 REST设计标准

设计标准对比

- 设计理念
  - API使用方尽早参与
  - 在具体实现前尽早迭代

7.2 REST设计标准

设计标准对比

- 商业模式
  - 背靠大公司支持

设计标准	支持公司	公司行业	运营模式
Swagger	SmartBear	自动化测试	开源 + 商业版本
RAML	Mulesoft	软件集成	社区项目 + 商业(免费 + 付费)
API Blueprint	Oracle	数据库 + 云基础设施	社区项目 + 商业版本

7.2 REST设计标准

设计标准对比

- 活跃度
  - 标准规范

标准规范	最新版本	发布时间	贡献者
OpenAPI	3.0	2017	Linux基金会
	3.1	2021	Linux基金会
RAML	1.0	2016	社区
	1.0.1	2021	社区
API Blueprint	Format 1A9	2015	社区

7.2 REST设计标准

设计标准对比

- 活跃度
  - 工具平台

设计标准	开源版本/社区项目	商业版本
Swagger	非常活跃(ui/editor/codegen等)	SwaggerHub + ReadyAPI
RAML	不活跃	Mulesoft Anypoint Platform
API Blueprint	不活跃	Oracle Apiary API Flow

7.2 REST设计标准

设计标准对比

- 活跃度
  - 开发框架支持

设计标准	开发框架
Swagger	友好支持spring框架，支持OAS3.x SpringFox，Knife4J，Springdoc-openapi

7.3 基于Swagger的设计实战

OpenAPI Specification

- 定义“接口的接口”
  - 一套用于定义RESTful API的标准的，与具体编程语言无关的规范
  - 允许使用者和计算机都可以发现和理解服务的能力
- 发布版本
  - OAS v2.0, OAS v3.0, OAS v3.1

7.3 基于Swagger的设计实战

OpenAPI Specification

- 版本约定
  - 使用符合semver规范的版本号
  - major.minor.patch，alpha/beta/rc
- 基本格式
  - JSON或YAML格式
  - 所有字段名都是小写
  - 字段分两类：固定字段 和 模式字段

7.3 基于Swagger的设计实战

OpenAPI Specification

- 文档结构
  - 单个文件 或 被拆分为多个文件
  - 根文档命名一般为openapi.json 或 openapi.yaml

7.3 基于Swagger的设计实战

OpenAPI Specification

- 数据类型
  - 原始数据类型参考JSON Schema Specification
  - 原始数据类型有一个可选的修饰属性：format
  - integer：int32、int64 format
  - number：float、double format
  - string：byte、binary、date、date-time、password format
  - boolean

7.3 基于Swagger的设计实战

OpenAPI Specification

- 富文本格式
  - 整个规范的description字段被标识为支持CommonMark markdown格式
  - CommonMark 0.27中描述的markdown 语法
- URL的关联引用
  - 一般而言，所有URL类型的属性值都可以是相对地址

7.3 基于Swagger的设计实战

OpenAPI Specification

◆ 数据结构-OpenAPI对象

□ OpenAPI文档的根文档对象

字段名	类型	描述
openai	string	必选，openapi规范的版本号
info	Info对象	必选，API相关的元数据
servers	[Server对象]	提供到服务器的连接信息
paths	Paths对象	必选，对所提供的API有效的路径和操作
components	Components对象	包含多种结构的元素
security	[安全需求对象]	声明API使用的安全机制
tage	[Tag对象]	提供更多元数据的一系列标签
externalDocs	外部文档对象	附件文档

7.3 基于Swagger的设计实战

OpenAPI Specification

- 数据结构-Info对象
  - 提供API的元数据

字段名	类型	描述
title	string	必选
description	string	可使用CommonMark词法
termsOfService	string	指向服务条款的URL地址
contact	Contact对象	
license	License对象	
version	string	必选，API文档的版本信息

7.3 基于Swagger的设计实战

OpenAPI Specification

- 数据结构-Server对象
  - 表示一个服务器的对象

字段名	类型	描述
url	string	必选，指向目标主机的URL地址。模板化支持，可包含变量
description	string	可使用CommonMark词法
variables	Map<string, Server变量对象>	一组变量和值的映射

7.3 基于Swagger的设计实战

OpenAPI Specification

- 数据结构-Server Variable对象
  - 用于服务器URL地址模板变量替换的对象

字段名	类型	描述
enum	[string]	替换项只能固定的某些值
default	string	必选，默认值
description	string	可使用CommonMark词法

7.3 基于Swagger的设计实战

OpenAPI Specification

- 数据结构-Components对象
  - 包含开放API规范固定的各种可重用组件
  - schemas，Map<string, Schema对象>
  - responses，parameters，examples，requestBodies
  - headers，securitySchemes，links，callbacks
  - 均可引用

7.3 基于Swagger的设计实战

OpenAPI Specification

- 数据结构-Path对象
  - 定义各个端点和操作的相对路径
  - 模型字段：/{path}
  - 类型：PathItem对象

7.3 基于Swagger的设计实战

OpenAPI Specification

◆ 数据结构-PathItem对象

- 描述对一个路径可执行的有效操作

字段名	类型	描述
$ref	[string]	替换项只能固定的某些值
summary	string	必选，默认值
description	string	可使用CommonMark词法
get/put/post/delete/options/head/patch/trace，Operation对象		
servers	[Server对象]	可用于替代根server
parameters	[Parameter对象]	此路径下所有操作的参数的列表，可引用

7.3 基于Swagger的设计实战

OpenAPI Specification

- 数据结构-Operation对象
  - 描述对路径的某个操作

字段名	类型	描述
tags	[string]	
summary	string	
description	string	
externalDocs		
opertaionId	string	
parameters	[Parameter对象]	可覆盖PathItem中的同名参数
requestBody	RequestBody对象	
responses	Response对象	必须

7.3 基于Swagger的设计实战

OpenAPI Specification

- 数据结构-Operation对象
  - 描述对路径的某个操作

字段名	类型	描述
callbacks	Map<string, Callback对象>	一组相对于父操作的回调映射
deprecated	boolean	声明是否已被废弃
security	[Security需求对象]	
servers	[Server对象]	可覆盖上游的定义

7.3 基于Swagger的设计实战

OpenAPI Specification

- 数据结构-Parameter对象
  - 描述一个参数的参数

字段名	类型	描述
name	string	必选，参数名称，区分大小写
in	string	必选，参数位置，query/header/path/cookie
description	string	
required	boolean	可覆盖上游的定义
deprecated	boolean	
allowEmptyValue	boolean	

7.3 基于Swagger的设计实战

OpenAPI Specification

- 数据结构-Parameter对象
  - 描述一个参数的参数

字段名	类型	描述
style	string	form，simple
explode	boolean	
allowReserved	boolean	
schema	Schema对象	
example	Any	
examples	Map<string, Example对象>	不同媒体类型的示例
content	Map<string, MediaType对象>	不同媒体类型

7.3 基于Swagger的设计实战

OpenAPI Specification

- 数据结构-RequestBody对象
  - 定义请求体

字段名	类型	描述
description	string	form，simple
content	Map<String, MediaType对象>	必选，请求体的内容
required	boolean	

7.3 基于Swagger的设计实战

OpenAPI Specification

- 数据结构-MediaType对象
  - 每种媒体类型对象都有相应的结构和示例来描述它

字段名	类型	描述
schema	Schema对象	定义此种媒体类型的结构，可引用
example	Any	
examples	Map<string, Example对象>	可引用
encoding	Map<string, Enconding对象>	各属性字段的编码信息

7.3 基于Swagger的设计实战

OpenAPI Specification

- 数据结构-Responses对象
  - 描述一个操作可能发生的响应码与响应包含的响应体的对象
  - 固定字段

字段名	类型	描述
default	Response对象	未被明确声明的HTTP状态码的响应的文档

  - 模式字段

字段名	类型	描述
HTTP状态码	Response对象	每个HTTP状态码只能使用一次，4XX通配符

7.3 基于Swagger的设计实战

OpenAPI Specification

- 数据结构-Response对象
  - 单个API操作的响应

字段名	类型	描述
description	string	
headers	Map<string, Header对象>	
content	Map<string, MediaType对象>	
links	Map<string, Link对象>	可关联的操作连接

7.3 基于Swagger的设计实战

OpenAPI Specification

- 数据结构-Example对象
  - 单个API操作的响应

字段名	类型	描述
summary	string	
description	string	
value	Any	
externalValue	string	指向存有示例数据的一个url，与Value互斥

7.3 基于Swagger的设计实战

OpenAPI Specification

- 数据结构-Link对象
  - 一个响应关联的操作

字段名	类型	描述
operationRef	string	
operationId	string	与operationRef互斥
parameters	Map<String, Any>	从当前操作的请求或响应中获取参数
requestBody	Any	
description	string	
server	Server对象	

7.3 基于Swagger的设计实战

OpenAPI Specification

- 数据结构-Header对象
  - 与Parameter对象基本一致，除了一下几点：
  - name不能被指定
  - in不能被指定
  - 所有被location影响的特性必须适合header中的一个location

7.3 基于Swagger的设计实战

OpenAPI Specification

- 数据结构-Reference对象
  - 一个允许引用规范内部的其他部分或外部规范的对象
  - 可以引用对象、外部文档或外部文档的对象

字段名	类型	描述
$ref	string	必选

7.3 基于Swagger的设计实战

OpenAPI Specification

- 数据结构-Schema对象
  - 用于定义输入和输出的数据类型，可以是对象，或原始值和数组
  - 是JSON Schema Specification扩展后的子集

7.3 基于Swagger的设计实战

OpenAPI Specification

- 数据结构-Security Scheme对象
  - 定义一个用于Operations的安全方案对象

字段名	类型	描述
type	string	必选，apiKey，http，oauth2，openIdConnect
description	string	
name	string	适用于apiKey
in	string	适用于apiKey，query/header/cookie
scheme	string	适用于http
baererFormat	string	适用于http
flows	OAuthFlows对象	适用于oauth2
openIdConnectUrl	string	适用于openIdConnect

7.3 基于Swagger的设计实战

OpenAPI Specification

- 数据结构-OAuthFlows对象
  - 允许配置支持的OAuthFlows

字段名	类型	描述
implicit	OAuthFlow对象	
password	OAuthFlow对象	
clientCredentials	OAuthFlow对象	
authorizationCode	OAuthFlow对象	

7.3 基于Swagger的设计实战

OpenAPI Specification

- 数据结构-OAuthFlow对象
  - 允许OAuthFlow的配置详情

字段名	类型	描述
authorizationUrl	string	
tokenUrl	string	
refreshUrl	string	
scopes	Map<string, string>	

7.3 基于Swagger的设计实战

OpenAPI Specification

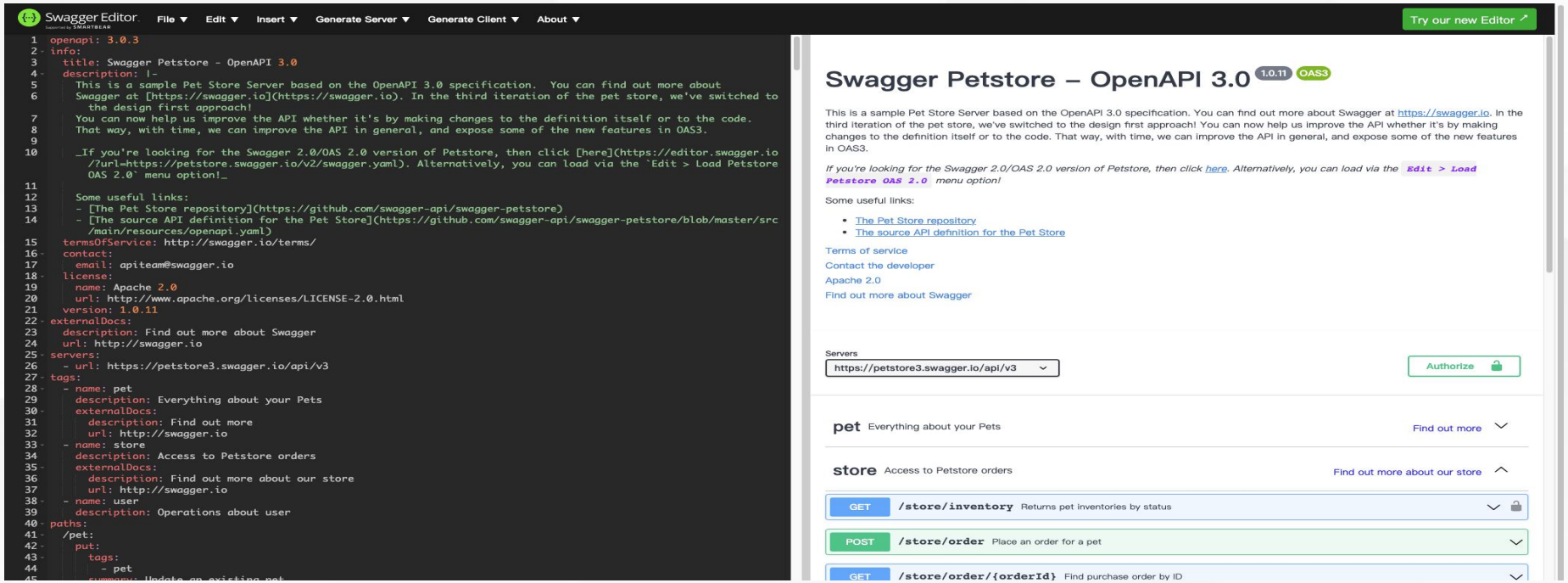
- 数据结构-SecurityRequirement对象
  - 列出某个operation所需的security schemes
  - 模式字段

字段名	类型	描述
{name}	[string]	

7.3 基于Swagger的设计实战

Open Source Tools

- Design工具
  - Swagger Editor



7.3 基于Swagger的设计实战

Open Source Tools

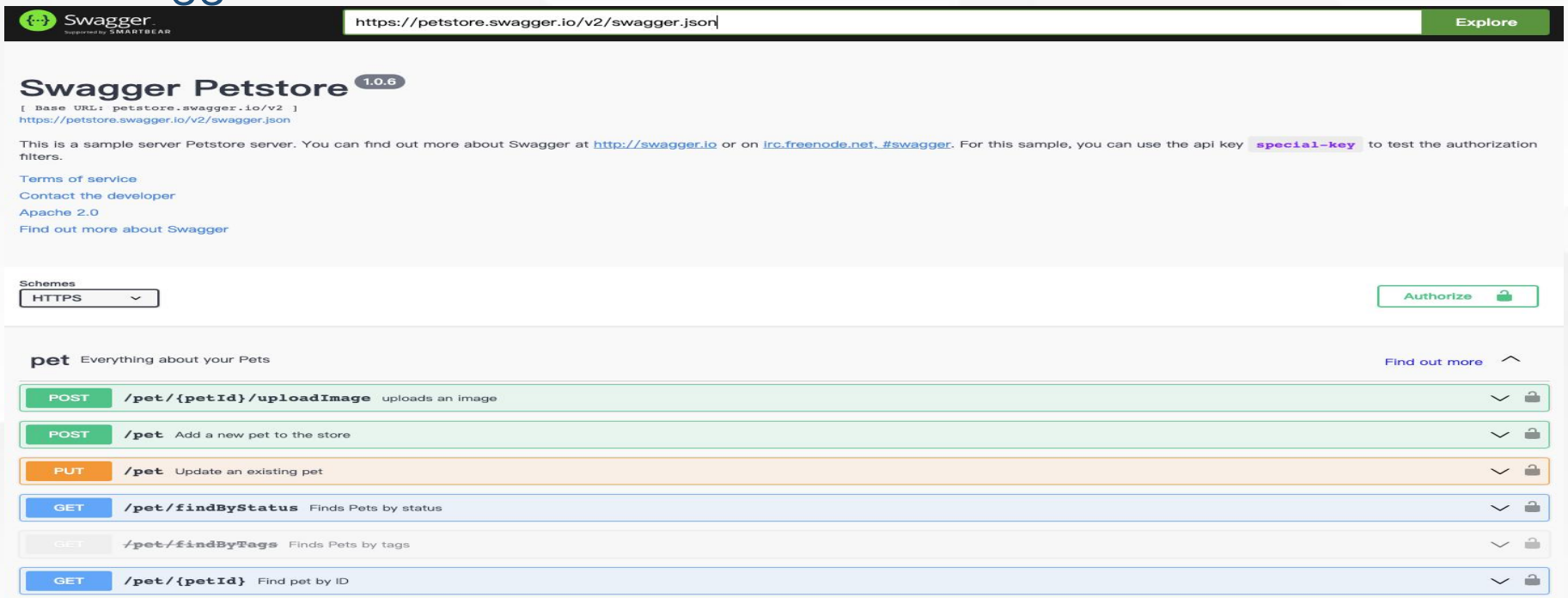
- Build工具
  - Swagger Codegen

	Codegen Download	SwaggerHub Sign Up Free
General Features		
Automated Server Stub Generation	✓	✓
Automated SDK Generation	✓	✓
Advanced Editing		
Codeless Mocking		✓
Style Guide Enforcement		✓
Reusable Components		✓
Hosting & Maintenance		
Cloud Hosting		✓
Centralized Definition Storage		✓
Versioning for API Definitions		✓
Collaboration & Sharing		
Access Control and Secure Sharing		✓
Issue Tracking and Commenting		✓
Change Notifications		✓
Third-Party Integrations		

7.3 基于Swagger的设计实战

Open Source Tools

- Document工具
  - Swagger UI



7.3 基于Swagger的设计实战

案例分析与实现一

- 从设计到实现
  - Swagger Editor
  - Swagger Codegen

7.3 基于Swagger的设计实战

案例分析与实现二

- SpringBoot项目集成Knife4j
  - Knife4j + openapi3

QA