

Savorboard

『代码如诗』...这是我们为世界写的诗歌。

[博客园](#)[首页](#)[联系](#)[订阅](#)[管理](#)

随笔 - 94 文章 - 0 评论 - 1691

谈谈微服务中的 API 网关 (API Gateway)

前言

又是很久没写博客了，最近一段时间换了新工作，比较忙，所以没有抽出来太多的时间写给关注我的粉丝写一些干货了，就有人问我怎么最近没有更新博客了，在这里给大家抱歉。

那么，在本篇文章中，我们就一起来探讨一下 API 网关在整个微服务分布式架构中的一个作用。

背景

我们知道在微服务架构风格中，一个大应用被拆分成为了多个小的服务系统提供出来，这些小的系统他们可以自成体系，也就是说这些小系统可以拥有自己的数据库，框架甚至语言等，这些小系统通常以提供 Rest Api 风格的接口来被 H5, Android, IOS 以及第三方应用程序调用。

但是在UI上进行展示的时候，我们通常需要在界面上展示很多数据，这些数据可能来自于不同的微服务中，举个例子。

在一个电商系统中，查看一个商品详情页，这个商品详情页包含商品的标题，价格，库存，评论等，这些数据对于后端来说可能是位于不同的微服务系统之中，可能我后台的系统是这样来拆分我的服务的：

公告

访问统计:

01008642



姓名: 杨晓东

所在位置: 四川 - 成都

Email: yangxiaodong1214@126.com

联系QQ: 615709110

Software Engineer / FOSS Developer

昵称: Savorboard

园龄: 8年10个月

荣誉: 推荐博客

粉丝: 2026

关注: 18

关注我

105

0

- 产品服务 - 负责提供商品的标题，描述，规格等。
- 价格服务 - 负责对产品进行定价，价格策略计算，促销价等。
- 库存服务 - 负责产品库存。
- 评价服务 - 负责用户对商品的评论，回复等。

现在，商品详情页需要从这些微服务中拉取相应的信息，问题来了？

问题

由于我们使用的服务系统架构，所以没办法像传统单体应用一样依靠数据库的 join 查询来得到最终结果，那么如何才能访问各个服务呢？

按照微服务设计的指导原则，我们的微服务可能存在下面的问题：

- 服务使用了多种协议，因为不同的协议有不同的应场景用，比如可能同时使用 HTTP, AMQP, gRPC 等。
- 服务的划分可能随着时间而变化。
- 服务的实例或者Host+端口可能会动态的变化。

那么，对于前端的UI需求也可能会有以下几种：

- 粗粒度的API，而微服务通常提供的细粒度的API，对于UI来说如果要调用细粒度的api可能需要调用很多次，这是个不小的问题。
- 不同的客户端设备可能需要不同的数据。Web,H5,APP
- 不同设备的网络性能，对于多个api来说，这个访问需要转移的服务端会快得多

以上，就是我们构建微服务的过程中可能会遇到的问题。那么如何解决呢？

这种情况下，我们就需要一个今天要讲的这个东西，API 网关（API Gateway）。

API 网关

+加关注

最新随笔

- 1.CAP 2.6 版本发布通告
- 2.基于 Kong 和 Kubernetes 的 WebApi 多版本解决方案
- 3.ASP.NET Core 身份验证（一）
- 4.CAP 2.5 版本中的新特性
- 5.还在用NuGet吗？大哥FuGet了解一下
- 6.CAP 2.4版本发布，支持版本隔离特性
- 7.在 ASP.NET Core 中集成 Skywalking APM
- 8.如何在你的项目中集成 CAP【手把手视频教程】
- 9.CAP 2.3版本发布,支持 MongoDB
- 10.MongoDB 4.0 开发环境搭建集群

随笔分类

.NET Core(36)
ASP.NET Core(47)
C#(3)
EF Core(3)
Micro Service(12)
Python(3)
翻译转载(9)
前端(2)
算法 & 设计模式(4)

★ 关注我

105

0

随笔档案

下面是百度上针对于 API 网关的介绍：

API网关是一个服务器，是系统的唯一入口。从面向对象设计的角度看，它与外观模式类似。API网关封装了系统内部架构，为每个客户端提供一个定制的API。它可能还具有其它职责，如身份验证、监控、负载均衡、缓存、请求分片与管理、静态响应处理。

API网关方式的核心要点是，所有的客户端和消费端都通过统一的网关接入微服务，在网关层处理所有的非业务功能。通常，网关也是提供REST/HTTP的访问API。服务端通过API-GW注册和管理服务。

Chris Richardson 在他的博客中把 API 网关划分为以下两种：

- 单节点 API 网关
- Backends for frontends 网关

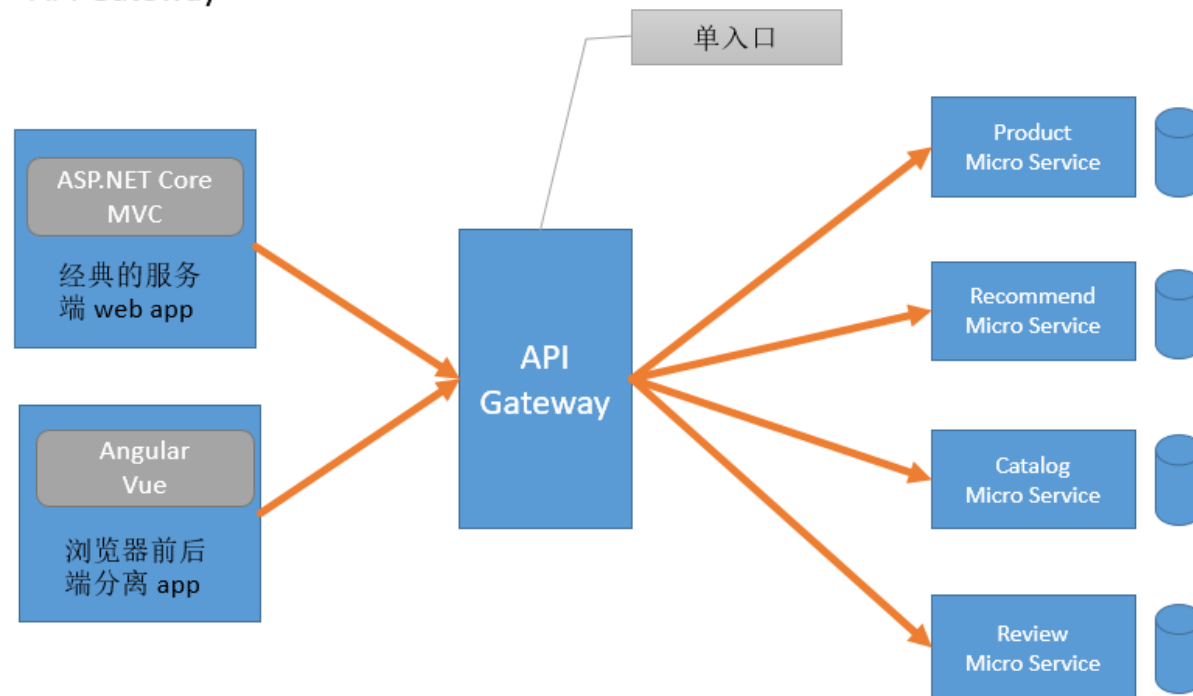
单节点网关

2019年8月(1)
2019年6月(1)
2019年4月(2)
2019年1月(2)
2018年10月(1)
2018年9月(1)
2018年8月(2)
2018年4月(1)
2018年3月(1)
2017年12月(1)
2017年10月(1)
2017年8月(4)
2017年7月(6)
2017年6月(2)
2017年5月(4)
2017年4月(4)
2017年3月(4)
2017年2月(4)
2017年1月(5)
2016年12月(5)
2016年11月(5)
2016年10月(5)
2016年9月(6)
2016年8月(5)
2016年7月(5)
2016年6月(5)
2016年5月(1)
2016年4月(2)
2014年8月(1)
2013年9月(1)
2013年5月(1)
2013年1月(2)
2012年12月(1)
2012年10月(1)

★ 关注我

105 0

API Gateway



单节点的 API网关为每个客户端提供不同的API，而不是提供一种万能风格的API。

这个网关和微软在 eShop 项目中推荐的网关是一致的。

更多详细信息我会在下文进行说明。

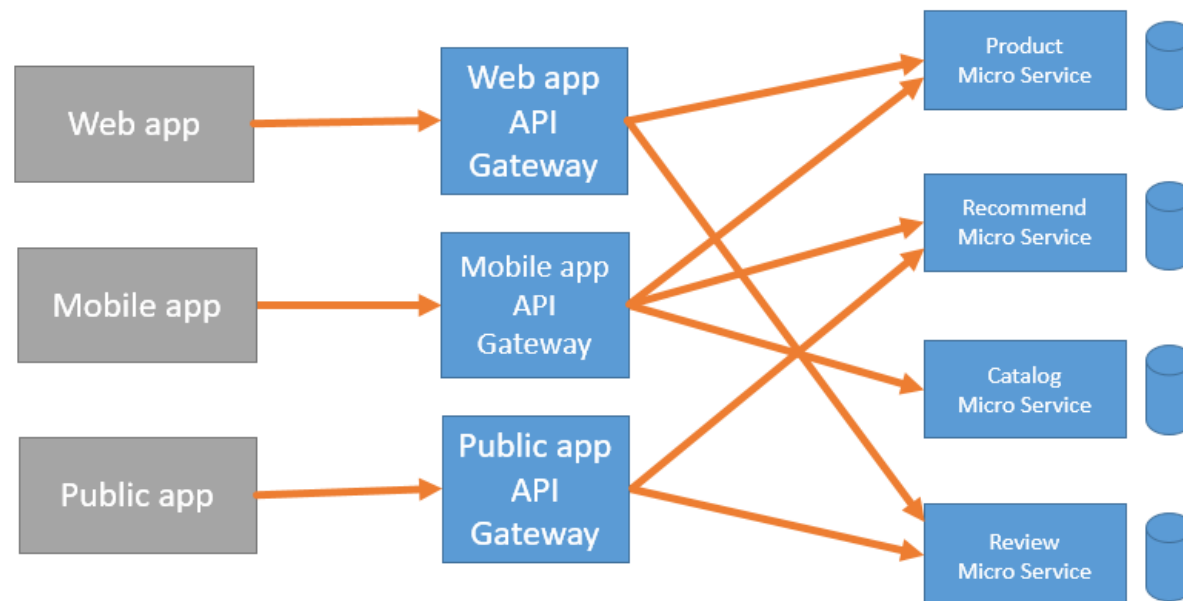
Backends for frontends 网关

★ 关注我

105

0

Backends for frontends API Gateway



这种模式是针对不同的客户端来实现一个不同的API网关。

落地方案

我一直在寻思一种最佳的 API 网关的落地方案，以上两种 API 网关有什么问题呢？

通常情况下，API 网关要做很多工作，它作为一个系统的后端总入口，承载着所有服务的组合路由转换等工作，除此之外，我们一般也会把安全，限流，缓存，日志，监控，重试，熔断等放到 API 网关来做，那么可以试想在高并发的情况下，这里可能会出现一个性能瓶颈。

另外，如果没有开源项目的支撑前提下，自己来做这样一套东西，是非常大的一个工作量，而且还要做 API 网关本身的高可用等，如果一旦做不好，有可能最先挂掉的不是你的其他服务，而就是这个API网关。

★ 关注我

105

0

这个时候，通常会去找一些开源的 API 网关项目，博主已经给你找好了，目前社区的关于 API Gateway 的项目有以下这些：

[Tyk](#)：Tyk是一个开放源码的API网关，它是快速、可扩展和现代的。Tyk提供了一个API管理平台，其中包括API网关、API分析、开发人员门户和API管理面板。Try 是一个基于Go实现的网关服务。

[Kong](#)：Kong是一个可扩展的开放源码API Layer(也称为API网关或API中间件)。Kong 在任何RESTful API的前面运行，通过插件扩展，它提供了超越核心平台的额外功能和服务。

[Orange](#)：和Kong类似也是基于OpenResty的一个API网关程序，是由国人开发的，学姐也是贡献者之一。

[Netflix zuul](#)：Zuul是一种提供动态路由、监视、弹性、安全性等功能的边缘服务。Zuul是Netflix出品的一个基于JVM路由和服务端的负载均衡器。

[apiaxle](#)：Nodejs 实现的一个 API 网关。

[api-umbrella](#)：Ruby 实现的一个 API 网关。

我们来说说上面的这些开源项目适不适合作为 API 网关来供我们使用。

拿单节点网关来说，这种网关相当于是处于 Web 层和 Service 之间，用来聚合服务的？注意，我们需要的是**聚合服务**，而以上这些开源项目都不具备这个功能，我说的聚合具体指的是开箱即用。我们要想使用这些服务需要来自自己对API网关过一些扩展或者是开发一些插件，这个时候问题就来了。扩展Tyk我需要会Go语言，扩展Kong我需要会写lua脚本，使用 zuul 还得会Java，这对于开发人员来说是不太现实的，那么这个时候怎么办？

有些同学可能会说 ASP.NET Core 可以使用 [Ocelot](#)，说得没错，我们可以通过引入Ocelot来处理API聚合服务这一块的业务，但是，这中间有一个问题，就像我在上面说的一样，这很容易造成性能问题，另外一方面，Ocelot 的功能相比上面的那些开源项目来说功能要弱很多，具体体现在哪些方面呢？

除了最重要的高性能的IO模型和集群方案外，比如会经常使用的 Dashboard 功能，这个对于运维来说是非常重要的，另外还有日志，监控，安全，服务发现，版本控制等。

但是上面的这些 API 网关缺少什么功能呢？比如超时，熔断，重试，聚合查询等。

注意：以下内容的这些想法全是我个人对于 API 网关的理解而诞生的，如有错误还请指正。

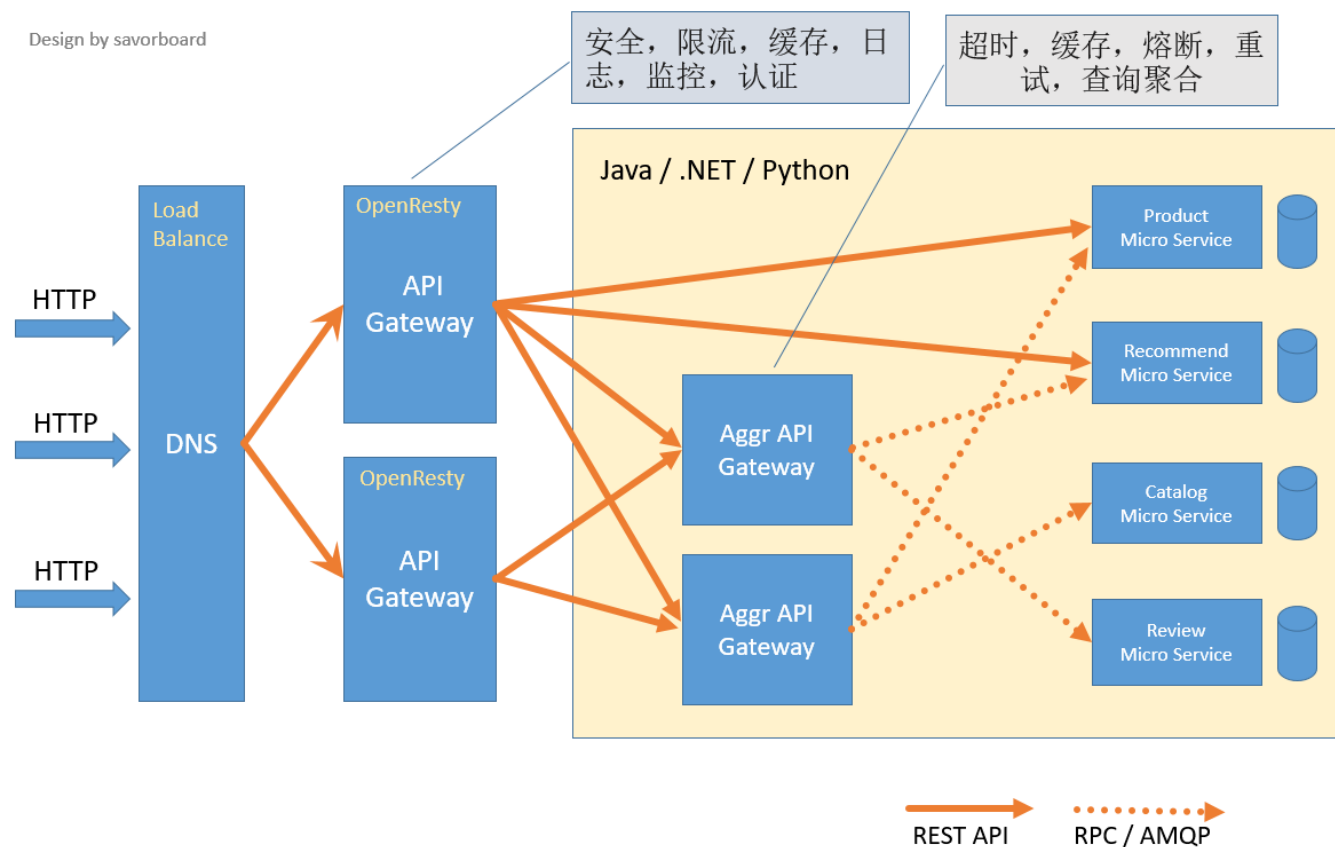
 关注我

105

0

聪明的同学可能想出来了，怎么办呢？我们可以充分来结合两者的优势来在我们的 ASP.NET Core 应用程序中实现一个“双重网关”。

下面是我画的一个 API 网关在微服务架构中的一个作用图：



应该大部分同学都可以看懂，我就简单解释一下。

- OpenResty Api Gateway

从左至右 HTTP 请求先由DNS在拿到第一手流量后负载均衡到基于 OpenResty 的 API Gataway 网关集群，在这个流程我们可以使用像 Kong,Orage,Tyk 这些开源的支持高并发高访问量 API 网关程序在做第一层流量的防护，在这一级我们可以做一些像身份认证，安全，监控，日志，流控等策略。除了这些我们还可以做一些服务的发现和注册（这个要看不同网关的支持程度），接口的版本控制，路由重写等。

★ 关注我

105

0

- Aggr Api Gateway

然后再由这些 API 网关把请求再负载到不同的 Aggr Api Gateway, 在这里我们做**聚合服务**这个操作, 具体体现也就是图中的黄色区域是需要由各个语言的开发人员来需要写代码实现的。具体流程也就是我们可以引入像 Ocelot 这种和语言相关的 API 网关开源项目, 然后通过 NuGet 包引入之后通过 Json配置+聚合代码的方式来整合后端的各个微服务提供聚合查询等操作。这期间对于有需求的接口, 我们可以应用超时, 缓存, 熔断, 重试等策略。

从 Aggr Api Gateway 到后端微服务集群这中间就属于内部的通讯了, 我们可以使用对内部友好的通讯协议比如 gRPC 或者 AMQP 等, 然后进行 RPC调用提高通讯性能。

注意: Aggr Api Gateway 这个网关对于一些接口来说的话并不是必须的, 也可以由后端微服务直接提供REST API给第一层网关使用。

以上, 就是我理解的 API 网关在整个微服务架构中的一个地位, 承上启下, 还是非常的重要。

广告

我们知道, 在 API 网关的后端是微服务的集群, 那么除了聚合查询之外有时候我们有时候需要做一些跨不同服务的操作, 比如一次电商系统的下单操作, 可以会设计到产品、价格、库存等一系列跨微服务操作, 在中间我们一般会采用集成事件(EventBus)来进行通讯这种解决方案, 在这个过程中我们不仅仅的是需要通讯, 那么这些操作还需要保证事务一致性, 而一般分布式系统中的事务我们追求的是最终一致性。

如果是在 .NET Core 中, 我们可以使用博主开源的 EventBus + 分布式事务 解决方案 [CAP](#)。

GitHub: <https://github.com/dotnetcore/CAP>

CAP 介绍:

<http://www.cnblogs.com/savorboard/p/cap.html>

有关分布式事务可以查看我的[这篇文章](#)。

感兴趣的同学可以给CAP一个 Star 以表支持, 偷偷告诉你 Ocelot的作者也Star了哦。:)

总结

★ 关注我

105

0

通过本文我们了解到了什么是 API 网关以及API网关的作用和其在微服务架构中所处的地位。然后我们了解到了 API 网关的一些开源项目以及博主介绍的落地方案，在实际的实践中还是多希望大家能够多多思考总结，这样我们才能够变得更加强大。

如果你觉得本篇文章对您有帮助的话，感谢您的【推荐】。

如果你对 .NET Core 有兴趣的话可以关注我，我会定期的在博客分享我的学习心得。

本文地址: <http://www.cnblogs.com/savorboard/p/api-gateway.html>

作者博客: [Savorboard](#)

欢迎转载，请在明显位置给出出处及链接

分类: .NET Core , Micro Service

标签: 微服务 , API Gateway , 网关

好文要顶

收藏该文



[Savorboard](#)

[关注 - 18](#)

[粉丝 - 2026](#)

推荐博客

[+加关注](#)

« 上一篇: [聊聊分布式事务，再说说解决方案](#)

» 下一篇: [招募: Wiki 文档翻译小伙伴招募](#)

posted @ 2017-12-07 12:01 Savorboard 阅读(95436) 评论(26) 编辑 收藏

评论列表

#1楼 2017-12-07 16:11 [幻天芒](#)

★ 关注我

105

0

多重复杂度也更高，出问题的几率更高。我们就是node实现的单节点（节点本身是分布式，高可用的）网关。网关自身的代码量并不大，功能可以逐步扩展。

支持(0) 反对(0)

#2楼 [楼主] 2017-12-07 17:28 Savorboard

@ 幻天芒

感觉复杂度还好，天芒兄你们的前端团队太强大，不能比啊。。

支持(0) 反对(0)

#3楼 2017-12-07 18:52 雪峰

赞

支持(0) 反对(0)

#4楼 2017-12-07 20:42 幻天芒

@ Savorboard

引用

@幻天芒

感觉复杂度还好，天芒兄你们的前端团队太强大，不能比啊。。

我们木有前端团队...node做这个很合适，request有pipe功能，可以建立管道。比client -> proxy proxy->server 效率高。

支持(0) 反对(0)

#5楼 2017-12-07 21:20 愤怒的TryCatch

目前是单网管的kong，一直想找到合适的开源网关实现，因为是一个孵化的小团队，后端就我一个人，自己去写不现实。

这种双重网关，感觉复杂度更高了，若是提交了一个新的服务，那么改动的地方或者需要运维的部分貌似多了好多。

突发的流量进入，第二层网关是不是也得和服务API同步保持实例增长啊？

支持(0) 反对(0)

★ 关注我

105

0

#6楼 [楼主] 2017-12-07 21:26 Savorboard

@ 愤怒的TryCatch

@幻天芒

不会的，只有聚合的才需要，你们用的Facade还是用node代码写的聚合呀？

支持(0) 反对(0)

#7楼 2017-12-07 21:47 愤怒的TryCatch

@ Savorboard

服务是使用的dotnet core(业务)、golang(推送服务)。

目前没有聚合，还不是根正苗红的微服务架构。

支持(0) 反对(0)

#8楼 2017-12-07 23:20 ~雨落忧伤~

感兴趣 但不会

支持(0) 反对(0)

#9楼 2017-12-07 23:21 幻天芒

@ Savorboard

引用

@愤怒的TryCatch

@幻天芒

不会的，只有聚合的才需要，你们用的Facade还是用node代码写的聚合呀？

没看懂facade和node怎么对比。我们这边是node写聚合，内层是分布式微服务（实际上就是多机部署），流量压力基本在gateway这一层，扩展也基本只需要扩展这一层即可。

支持(0) 反对(0)

#10楼 2017-12-08 07:32 张善友

★ 关注我

105

0

ocelot 还在发展中，需要大家一起参与建设

支持(1) 反对(0)

#11楼 2017-12-08 07:36 张善友

比如说dashboard， <https://github.com/dbarkwell/Ocelot.ConfigEditor>，很多想法还在探讨中，大家可以通过Issue 参与，或者直接贡献相关代码，自己撸

支持(0) 反对(0)

#12楼 [楼主] 2017-12-08 09:22 Savorboard

@ 张善友

引用

比如说dashboard， <https://github.com/dbarkwell/Ocelot.ConfigEditor>，很多想法还在探讨中，大家可以通过Issue 参与，或者直接贡献相关代码，自己撸

队长说的没错...

支持(0) 反对(0)

#13楼 2017-12-08 17:05 wdwwtzy

就是个门面模式呗？

支持(0) 反对(0)

#14楼 2017-12-08 17:42 Arthur1024

我们现在的个项目也只有聚合层网关，效果还是很好的

支持(0) 反对(0)

#15楼 2017-12-08 21:05 一百斤

★ 关注我

105

0

收藏

支持(0) 反对(0)

#16楼 2017-12-14 10:14 沈熨

谢谢分享

支持(0) 反对(0)

#17楼 2018-01-05 15:10 coolfiry

我也在做API网关，希望与你交流，微信：coolfiry

支持(0) 反对(0)

#18楼 2018-06-06 09:28 凌晨的太阳

Aggr Api Gateway，在图中的位置我觉得可以调整，他应该API GATE 与 API GATE 之间。

支持(0) 反对(0)

#19楼 [楼主] 2018-06-06 09:32 Savorboard

@ 凌晨的太阳
框架也可以实现的就是聚合服务，其实这里的聚合网关也可以理解为聚合服务，但是有个问题，就是如果Zuul到“Simple Api Gateway”再到Web Api 的话会多一层转发，担心影响性能，所以我的理解是，针对一些不需要聚合的接口而言，可以直接 Zuul到WebApi的

支持(0) 反对(0)

#20楼 2018-06-06 09:44 凌晨的太阳

@ Savorboard
楼主动作够快，Simple Api Gate的考虑基于如果存在业务层面的Service，简单的Api暴露本身意义不大。

★ 关注我

105 0

Simple Api Gate 作为一种服务的暴露方式应该是自然而然的。
Aggr Api Gateway 仅仅是其中的一种实现。

支持(0) 反对(0)

#21楼 2018-07-09 09:17 花开是心碎了

学习了，辛苦楼主，

支持(0) 反对(0)

#22楼 2018-10-16 19:12 Aiolos Wong

问下博主现在有哪些API网关对API聚合支持比较好？我调研了一圈，包括开源的和公有云厂商的API网关，发现对API聚合的支持都比较弱，无法满足实际业务场景。我期望的API聚合能支持以下功能：
API网关上定义的一个API支持映射到后台多个API（支持多种HTTP Method甚至多种协议组合）；支持抽取后台API结果的字段进行组合映射转换；支持后台API的串行调用和并行调用。串行调用就是API网关上面定义的一个API依赖于后台多个API的执行结果，执行结果之间需要有严格的执行顺序，甚至是需要后一个API的执行参数依赖前一个API的执行结果。上述功能在大部分场景下可以通过配置文件实现。

支持(0) 反对(0)

#23楼 2019-01-02 10:41 北京-阿布

mark

支持(0) 反对(0)

#24楼 2019-01-15 16:02 码行天下i

最近我们也在搞api网关,现在涉及到一个master服务器与node的双向认证的问题和协议的序列化和反序列化的问题

支持(0) 反对(0)

#25楼 2019-05-16 11:20 郝姬友

 关注我

105 0

多谢分享

支持(0) 反对(0)

#26楼 2019-09-26 14:54 WangJunZzz

你好，我们项目想用上api网关，但是有一个问题，就是项目在前后端分离的时候，那些接口实在前端调用(h5+jq或者vue),需要假如到网关里面吗？如果加入不可能在前端加密吧，（阿里云网关）

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#) 网站首页。

Copyright © 2019 Savorboard
Powered by .NET Core 3.0.0 on Linux

 [关注我](#)

105

0