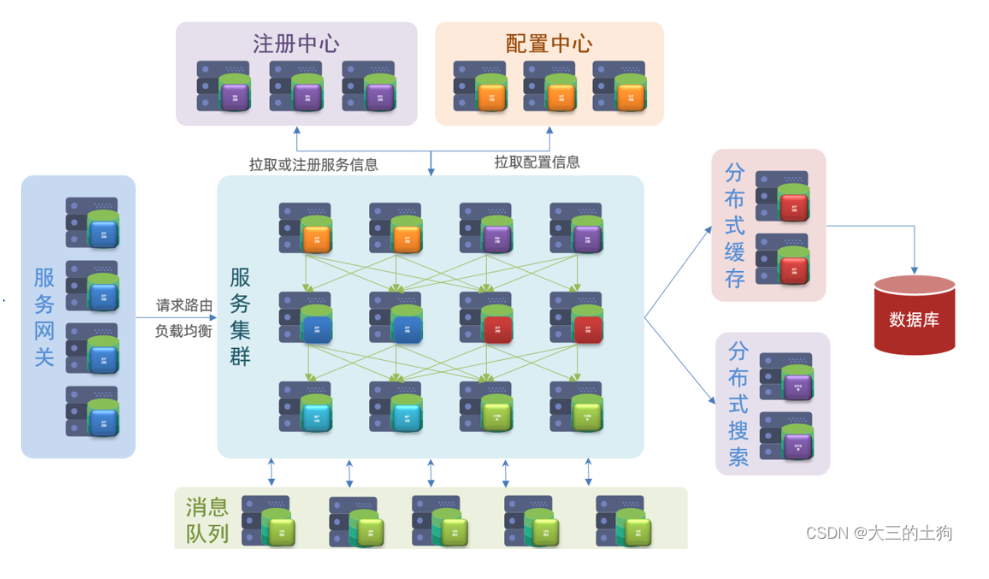
# 官网

http://projects.spring.io/spring-cloud/

# 学习路线



# 微服务架构

微服务架构就是将单体的应用程序分成多个应用程序，这多个应用程序就成为微服务，每个微服务运行在自己的进程中，并使用轻量级的机制通信。这些服务围绕业务能力来划分，并通过自动化部署机制来独立部署。这些服务可以使用不同的编程语言，不同数据库，以保证最低限度的集中式管理。

# Spring Cloud基本概念

1、Spring Cloud是一系列框架的有序集合。它利用Spring Boot的开发便利性巧妙地简化了分布式系统基础设施的开发，如服务发现注册、配置中心、智能路由、消息总线、负载均衡、断路器、数据监控等，都可以用Spring Boot的开发风格做到一键启动和部署。

2、Spring Cloud并没有重复制造轮子，它只是将各家公司开发的比较成熟、经得起实际考验的服务框架组合起来，通过Spring Boot风格进行再封装，屏蔽掉了复杂的配置和实现原理，最终给开发者留出了一套简单易懂、易部署和易维护的分布式系统开发工具包。

# SpringCloud的优缺点

## 优点

1、耦合度比较低，不会影响其他模块的开发。

2、减轻团队的成本，可以并行开发，不用关注其他人怎么开发，先关注自己的开发。

3、配置比较简单，基本注解就能实现，不用使用过多的配置文件。

4、微服务是跨平台的，可以用任何一种语言开发。

5、每个微服务可以有自己的独立数据库，也有用公共的数据库。

6、不用关注前端怎么开发，直接写自己的后端代码即可，然后暴露接口，通过组件进行服务通信。

## 缺点

1、部署比较麻烦，给运维工程师带来一定的麻烦。

2、针对数据的管理麻烦，因为每个微服务可以使用一个数据库。

3、系统集成测试比较麻烦。

4、性能的监控比较麻烦。【最好开发一个大屏监控系统】

总的来说优点大于缺点，目前看来Spring Cloud是一套非常完善的分布式框架，目前很多企业开始用微服务，Spring Cloud的优势是显而易见的。

# SpringBoot和SpringCloud的区别

1、SpringBoot专注于快速方便的开发单个个体微服务。

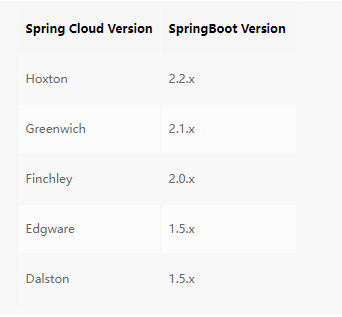
2、SpringCloud是关注全局的微服务协调整理治理框架，它将SpringBoot开发的一个个单体微服务整合并管理起来。

3、为各个微服务之间提供配置管理、服务发现、断路器、路由、微代理、事件总线、全局锁、决策竞选、分布式会话等集成服务。

4、SpringBoot可以离开SpringCloud独立使用开发项目， 但是SpringCloud离不开SpringBoot ，属于依赖的关系。

5、SpringBoot专注于快速、方便的开发单个微服务个体，SpringCloud关注全局的服务治理框架。

# Spring Cloud和SpringBoot版本对应关系



# SpringCloud组件

我讲几个开发中最重要的。

## 1、Spring Cloud Eureka

服务注册与发现

## 2、Spring Cloud Zuul

服务网关

## 3、Spring Cloud Ribbon

客户端负载均衡

## 4、Spring Cloud Feign

声明性的Web服务客户端

## 5、Spring Cloud Hystrix

断路器

## 6、Spring Cloud Config

分布式统一配置管理

## 7、Gateway

## 8、Sentinel

由阿里巴巴中间件团队开发的开源项目，是一种面向分布式微服务架构的轻量级高可用流量控制组件。

等20几个框架，开源一直在更新。

## 微服务网关Zuul和Gateway的区别

https://www.cnblogs.com/lgg20/p/12507845.html

## spring.factories文件的作用

# 使用Spring Boot开发分布式微服务时，我们面临什么问题

## 1、与分布式系统相关的复杂性

这种开销包括网络问题，延迟开销，带宽问题，安全问题。

## 2、服务发现

服务发现工具管理集群中的流程和服务如何查找和互相交谈。它涉及一个服务目录，在该目录中注册服务，然后能够查找并连接到该目录中的服务。

## 3、冗余

分布式系统中的冗余问题。

## 4、负载平衡

负载平衡改善跨多个计算资源的工作负荷，诸如计算机，计算机集群，网络链路，中央处理单元，或磁盘驱动器的分布。

## 5、性能问题

由于各种运营开销导致的性能问题。

# Eureka和ZooKeeper都可以提供服务注册与发现的功能,请说说两个的区别

1、ZooKeeper中的节点服务挂了就要选举，在选举期间注册服务瘫痪，虽然服务最终会恢复,但是选举期间不可用的。

2、Eureka各个节点是平等关系，服务器挂了没关系，只要有一台Eureka就可以保证服务可用，数据都是最新的，如果查询到的数据并不是最新的，就是因为Eureka的自我保护模式导致的。

3、Eureka本质上是一个工程，而ZooKeeper只是一个进程。

4、Eureka可以很好的应对因网络故障导致部分节点失去联系的情况,而不会像ZooKeeper 一样使得整个注册系统瘫痪。

5、ZooKeeper保证的是CP，Eureka保证的是AP。

CAP： C：一致性，A：可用性，P：分区容错性>Partition tolerance

# 网关与过滤器有什么区别

网关是对所有服务的请求进行分析过滤，过滤器是对单个服务而言。

# 常用的网关框架有那些

Nginx、Zuul、Gateway

# Zuul与Nginx有什么区别

Zuul是java语言实现的，主要为java提供网关服务，尤其在微服务架构中可以更加灵活的对网关进行操作。Nginx是使用C语言实现，性能高于Zuul，但是实现自定义操作，需要熟悉lua语言，对程序员要求较高，可以使用Nginx做Zuul集群。

# 既然Nginx可以实现网关，为什么还需要使用Zuul框架

Zuul是SpringCloud集成的网关，使用Java语言编写，可以对SpringCloud架构提供更灵活的服务。

# 如何设计一套API接口

考虑到API接口的分类可以将API接口分为开发API接口和内网API接口，内网API接口用于局域网，为内部服务器提供服务。开放API接口用于对外部合作单位提供接口调用，需要遵循Oauth2.0权限认证协议，同时还需要考虑安全性、幂等性等问题。

# 如何实现动态Zuul网关路由转发

通过path配置拦截请求，通过ServiceId到配置中心获取转发的服务列表，Zuul内部使用Ribbon实现本地负载均衡和转发。

# Zuul网关如何搭建集群

使用Nginx的upstream设置Zuul服务集群，通过location拦截请求并转发到upstream，默认使用轮询机制对Zuul集群发送请求。

# Ribbon底层实现原理

Ribbon使用discoveryClient从注册中心读取目标服务信息，对同一接口请求进行计数，使用%取余算法获取目标服务集群索引，返回获取到的目标服务信息。

# 什么是断路器

1、当一个服务调用另一个服务由于网络原因或自身原因出现问题，调用者就会等待被调用者的响应 当更多的服务请求到这些资源导致更多的请求等待，发生连锁效应（雪崩效应）

2、断路器有三种状态

打开状态：一段时间内 达到一定的次数无法调用 并且多次监测没有恢复的迹象 断路器完全打开 那么下次请求就不会请求到该服务。

半开状态：短时间内有恢复迹象 断路器会将部分请求发给该服务，正常调用时断路器关闭。

关闭状态：当服务一直处于正常状态能正常调用。

# 什么是 Hystrix

1、在分布式系统，我们一定会依赖各种服务，那么这些个服务一定会出现失败的情况，就会导致雪崩，Hystrix就是这样的一个工具，防雪崩利器，它具有服务降级，服务熔断，服务隔离，监控等一些防止雪崩的技术。

2、Hystrix有四种防雪崩方式：

服务降级：接口调用失败就调用本地的方法返回一个空

服务熔断：接口调用失败就会进入调用接口提前定义好的一个熔断的方法，返回错误信息。

服务隔离：隔离服务之间相互影响

服务监控：在服务发生调用时,会将每秒请求数、成功请求数等运行指标记录下来。

https://juejin.cn/post/6844904125717544973

# 注解

## @RefreshScope

### 基本概念