

SpringCloud配置文件最全最详细说明(properties和yml)

eureka

属性名	默认值	说明
eureka.client.allow-redirects	false	指示服务器是否可以将客户端请求重定向到备份服务器/集群。 如果设置为false，服务器将直接处理请求，如果设置为true，则可能会向客户端发送HTTP重定向，并具有新的服务器位置。
eureka.client.availability-zones		获取此实例所在区域的可用性区域列表（用于AWS数据中心）。 更改在运行时在registryFetchIntervalSeconds指定的下一个注册表提取周期中有效。
eureka.client.backup-registry-impl		获取执行BackupRegistry的实现的名称，以便仅在eureka客户端启动时首次将注册表信息作为回退选项提取。 对于需要额外的弹性的注册表信息的应用程序可能需要这一点，而无法运行它们。
eureka.client.cache-refresh-executor-exponential-back-off-bound	10	缓存刷新执行者指数退出相关属性。在发生超时序列的情况下，它是重试延迟的最大乘数值。
eureka.client.cache-refresh-executor-thread-pool-size	2	cacheRefreshExecutor初始化的线程池大小
eureka.client.client-data-accept		EurekaAccept客户端数据接受名称
eureka.client.decoder-name		这是一个瞬态配置，一旦最新的编解码器稳定，可以删除（因为只有一个）
eureka.client.disable-delta	false	指示eureka客户端是否应该禁用提取delta，而应该诉诸于获取完整的注册表信息。 请注意，增量获取可以极大地减少流量，因为尤利卡服务器的更改速率通常远低于提取速率。 更改在运行时在registryFetchIntervalSeconds指定的下一个注册表提取周期中有效
eureka.client.dollar-replacement	_-	在eureka服务器的序列化/反序列化信息期间，获取Dollar符号 <code><code> \$ </ code></code> 的替换字符串。
eureka.client.enabled	true	标记以指示启用Eureka客户端。
eureka.client.encoder-name		这是一个瞬态配置，一旦最新的编解码器稳定，可以删除（因为只有一个）
eureka.client.escape-char-replacement	__	在eureka服务器的序列化/反序列化信息期间获取下划线符号 <code><code> _ </ code></code> 的替换字符串。
eureka.client.eureka-connection-idle-timeout-seconds	30	表示到eureka服务器的HTTP连接可以在关闭之前保持空闲状态的时间（以秒为单位）。 在AWS环境中，建议值为30秒或更短，因为防火墙在几分钟内清除连接信息，将连接挂在空中
eureka.client.eureka-server-connect-timeout-seconds	5	指示在连接到eureka服务器需要超时之前等待（以秒为单位）的时间。请注意，客户端中的连接由org.apache.http.client.HttpClient汇集，此设置会影响实际的连接创建以及从池中获取连接的等待时间。
eureka.client.eureka-server-dns-name		获取要查询的DNS名称以获取eureka服务器的列表。如果合同通过实现serviceUrls返回服务URL，则不需要此信息。 当useDnsForFetchingServiceUrls设置为true时，使用DNS机制，而eureka客户端希望DNS以某种方式配置，以便可以动态获取更改的eureka服务器。 更改在运行时有效。

eureka.client.eureka-server-port		<p>获取用于构建服务url的端口， 以在eureka服务器列表来自DNS时联系eureka服务器。如果合同返回服务url eurekaServerServiceUrls（String）， 则不需要此信息。</p> <p>当useDnsForFetchingServiceUrls设置为true时，使用DNS机制， 而eureka客户端希望DNS以某种方式配置， 以便可以动态获取更改的eureka服务器。</p> <p>更改在运行时有效。</p>
eureka.client.eureka-server-read-timeout-seconds	8	指示从eureka服务器读取之前需要等待（秒）多久才能超时。
eureka.client.eureka-server-total-connections	200	获取从eureka客户端到所有eureka服务器允许的总连接数。
eureka.client.eureka-server-total-connections-per-host	50	获取从eureka客户端到eureka服务器主机允许的总连接数。
eureka.client.eureka-server-url-context		<p>获取用于构建服务网址的URL上下文， 以便在eureka服务器列表来自DNS时联系eureka服务器。如果合同从eurekaServerServiceUrls返回服务网址， 则不需要此信息。</p> <p>当useDnsForFetchingServiceUrls设置为true时，使用DNS机制， 而eureka客户端希望DNS以某种方式配置， 以便可以动态获取更改的eureka服务器。更改在运行时有效。</p>
eureka.client.eureka-service-url-poll-interval-seconds	0	表示轮询对eureka服务器信息进行更改的频率（以秒为单位）。可以添加或删除Eureka服务器， 此设置控制eureka客户端应该知道的时间。
eureka.client.fetch-registry	true	指示该客户端是否应从eureka服务器获取eureka注册表信息。
eureka.client.fetch-remote-regions-registry		逗号分隔将获取eureka注册表信息的区域列表。必须为availabilityZones返回的每个区域定义可用性区域。否则， 将导致发现客户端启动失败。
eureka.client.filter-only-up-instances	true	指示是否在仅具有InstanceStatus UP状态的实例的过滤应用程序之后获取应用程序。
eureka.client.g-zip-content	true	指示从服务器支持时， 是否必须压缩从eureka服务器提取的内容。来自eureka服务器的注册表信息被压缩以获得最佳的网络流量。
eureka.client.heartbeat-executor-exponential-back-off-bound	10	心跳执行者指数回撤相关财产。在发生超时序列的情况下， 它是重试延迟的最大乘数值。
eureka.client.heartbeat-executor-thread-pool-size	2	heartbeat执行器初始化的线程池大小
eureka.client.initial-instance-info-replication-interval-seconds	40	指示将实例信息复制到eureka服务器的开始时间（以秒为单位）
eureka.client.instance-info-replication-interval-seconds	30	指示复制要复制到eureka服务器的实例更改的频率（以秒为单位）。
eureka.client.log-delta-diff	false	<p>指示在注册表信息方面是否记录eureka服务器和eureka客户端之间的差异。</p> <p>Eureka客户端尝试仅从欧莱雅服务器检索增量更改以最小化网络流量。收到三角形后， eureka客户端将从服务器的信息进行协调， 以验证它是否已经没有漏掉一些信息。当客户端发生网络问题与服务器通信时， 可能会发生调解失败。如果对帐失败， eureka客户端将获得完整的注册表信息。</p> <p>在获取完整的注册表信息的同时， eureka客户端可以记录客户端和服务之间的差异， 并且此设置控制它。</p> <p>更改在运行时在registryFetchIntervalSecondsr指定的下一个注册表提取周期中有效</p>
eureka.client.on-demand-update-status-change	true	如果设置为true， 则通过ApplicationInfoManager进行的本地状态更新将触发对远程eureka服务器的按需（但限速）注册/更新
eureka.client.prefer-same-zone-eureka	true	<p>指示此实例是否应尝试在同一区域中使用尤里卡服务器延迟和/或其他原因</p> <p>理想情况下， eureka客户端配置为与同一区域中的服务器通信</p> <p>更改在运行时在registryFetchIntervalSeconds指定的下一个注册表提取周期中有效。</p>
eureka.client.property-resolver		
eureka.client.proxy-host		获取代理主机到eureka服务器（如果有的话）。

eureka.client.proxy-password		获取代理密码（如果有）。
eureka.client.proxy-port		获取代理端口到eureka服务器（如果有的话）。
eureka.client.proxy-user-name		获取代理用户名（如果有）。
eureka.client.region	us-east-1	获取此实例所在的区域（用于AWS数据中心）。
eureka.client.register-with-eureka	true	指示此实例是否应将其信息注册到eureka服务器以供其他人发现。 在某些情况下，您不希望实例被发现，而您只想发现其他实例。 如本地测试
eureka.client.registry-fetch-interval-seconds	30	指示从eureka服务器获取注册表信息的频率（以秒为单位）。
eureka.client.registry-refresh-single-vip-address		指示客户端是否只对单个VIP的注册表信息感兴趣。
eureka.client.service-url		可用性区域映射到与eureka服务器通信的完全限定URL的列表。每个值可以是单个URL或逗号分隔的替代位置列表。 通常，尤里卡服务器URL携带协议，主机，端口，上下文和版本信息（如果有的话）。示例：http://ec2-256-156-243-129.compute-1.amazonaws.com:7001/eureka/ 更改在运行时在eurekaServiceUrlPollIntervalSeconds指定的下一个服务网址刷新周期中有效。
eureka.client.transport		
eureka.client.use-dns-for-fetching-service-urls	false	指示eureka客户端是否应该使用DNS机制来获取要与之通信的eureka服务器列表。当DNS名称更新为具有其他服务器时，该信息将在eureka客户端轮询该eurekaServiceUrlPollIntervalSeconds中指定的信息之后立即使用。 或者，服务urls可以返回serviceUrls，但用户应该实现自己的机制来返回更新的列表，以防发生更改。 更改在运行时有效。
eureka.dashboard.enabled	true	标志以启用Eureka仪表板。默认值为true。
eureka.dashboard.path	/	到Eureka仪表板（相对于servlet路径）的路径。默认为"/"。
eureka.instance.a-s-g-name		获取与此实例关联的AWS自动缩放组名称。该信息在AWS环境中专门用于在实例启动后自动将实例停止运行，并且已将其禁用。
eureka.instance.app-group-name		获取要在eureka中注册的应用程序组的名称。
eureka.instance.appname	unknown	获取要在eureka注册的应用程序的名称。
eureka.instance.data-center-info		返回此实例部署的数据中心。如果实例部署在AWS中，则此信息用于获取一些AWS特定实例信息。
eureka.instance.default-address-resolution-order	[]	
eureka.instance.environment		
eureka.instance.health-check-url		获取此实例的绝对运行状况检查页面URL。如果运行状况检查页面驻留在与eureka通话的同一个实例中，用户可以提供healthCheckUriPath，否则在实例是其他服务器的代理的情况下，用户可以提供完整的URL。如果提供完整的URL，则优先。 它通常用于根据实例的健康状况做出有根据的决策 - 例如，它可用于确定是否继续部署到整个场，或者停止部署而不会造成进一步的损坏。完整的URL应遵循格式

feign

属性名	默认值	说明
feign.compression.request.mime-types	[text/xml, application/xml, application/json]	支持的MIME类型列表。
feign.compression.request.min-request-size	2048	最小阈值内容大小。

health

属性名	默认值	说明
health.config.enabled	false	标记以指示应安装配置服务器运行状况指示器。
health.config.time-to-live	0	生成缓存结果的时间，以毫秒为单位。默认300000（5分钟）。

hystrix

属性名	默认值	说明
hystrix.metrics.enabled	true	启用Hystrix指标轮询。默认为true。
hystrix.metrics.polling-interval-ms	2000	后续轮询度量之间的间隔。默认为2000 ms。

zuul

属性名	默认值	说明
zuul.add-host-header	false	标识以确定代理是否转发主机头。
zuul.add-proxy-headers	true	标识以确定代理是否添加X-Forwarded- *标头。
zuul.host.max-per-route-connections	20	单个路由可以使用的最大连接数。
zuul.host.max-total-connections	200	代理可以容纳到后端的总连接数。
zuul.ignore-local-service	true	
zuul.ignore-security-headers	true	标记说，如果spring security在类路径上，则将SECURITY_HEADERS添加到忽略的标头。通过将ignoreSecurityHeaders设置为false，我们可以关闭此默认行为。这应该与禁用默认的spring security标头一起使用，请参见 https://docs.spring.io/spring-security/site/docs/current/reference/html/headers.html#default-security-headers
zuul.ignored-headers		HTTP标头的名称完全忽略（即将其从下游请求中删除，并将其从下游响应中删除）。
zuul.ignored-patterns		
zuul.ignored-services		一组服务名称不考虑代理自动。默认情况下，发现客户端中的所有服务都将被代理。
zuul.prefix		所有路由的公共前缀。
zuul.remove-semicolon-content	true	标记说，可以删除超过第一个分号的路径元素。
zuul.retryable		默认情况下是否支持重试的标志（假设路由本身支持）。
zuul.ribbon-isolation-strategy		
zuul.routes		将路线名称映射到属性。
zuul.securityheaders		一般预期由Spring安全性添加的标头，因此如果代理和后端使用Spring保护，则通常会重复。默认情况下，如果存在Spring安全性，并且ignoreSecurityHeaders = true，它们将被添加到忽略的标头。
zuul.semaphore.max-semaphores	100	Hystrix的总信号量的最大数量。

zuul.sensitive-headers		不传递到下游请求的敏感标头列表。默认为通常包含用户凭据的“安全”标题集。如果下游服务是与代理相同的系统的一部分，那么从列表中删除它们是正确的，所以他们正在共享认证数据。如果在您自己的域之外使用物理URL，那么一般来说泄漏用户凭据将是一个坏主意。
zuul.servlet-path	/zuul	安装Zuul作为servlet的路径（不是SpringMVC的一部分）。对于具有大型机构的请求，例如文件上传，servlet对于更高的内存效率更高。
zuul.ssl-hostname-validation-enabled	true	标记以说明是否应验证ssl连接的主机名。默认值为true。这只应用于测试设置！
zuul.strip-prefix	true	在转发之前标记是否从路径中删除前缀。
zuul.trace-request-body	true	标记说可以跟踪请求机构。

management

属性名	默认值	说明
management.health.refresh.enabled	true	启用刷新范围的运行状况端点。
management.health.zookeeper.enabled	true	启用zookeeper的健康端点。

netflix

属性名	默认值	说明
netflix.atlas.batch-size	10000	
netflix.atlas.enabled	true	
netflix.atlas.uri		
netflix.metrics.servo.cache-warning-threshold	1000	当ServoMonitorCache达到这个大小时，会记录一个警告。 如果您在RestTemplate url中使用字符串连接，这将非常有用。
netflix.metrics.servo.registry-class	com.netflix.servo.BasicMonitorRegistry	Servo使用的监视器注册表的完全限定类名。

proxy

属性名	默认值	说明
proxy.auth.load-balanced		
proxy.auth.routes		每个路由的认证策略。

encrypt

属性名	默认值	说明
encrypt.fail-on-error	true	标记说，如果存在加密或解密错误，进程将失败。
encrypt.key		对称密钥。作为一个更强大的替代方案，考虑使用密钥库。
encrypt.key-store.alias		商店中的钥匙别名
encrypt.key-store.location		密钥存储文件的位置，例如classpath: /keystore.jks。

encrypt.key-store.password		锁定密钥库的密码。
encrypt.key-store.secret		秘密保护密钥（默认为密码相同）。
encrypt.rsa.algorithm		使用RSA算法（DEFAULT或OEAP）。一旦设置不改变它（或现有的密码将不可解密）。
encrypt.rsa.salt	deadbeef	Salt用于加密密文的随机秘密。一旦设置不改变它（或现有的密码将不可解密）。
encrypt.rsa.strong	false	标志表示应该在内部使用“强”AES加密。如果为真，则将GCM算法应用于AES加密字节。默认值为false（在这种情况下使用“标准”CBC代替）。一旦设置不改变它（或现有的密码将不可解密）。

endpoint

属性名	默认值	说明
endpoints.bus.enabled		
endpoints.bus.id		
endpoints.bus.sensitive		
endpoints.consul.enabled		
endpoints.consul.id		
endpoints.consul.sensitive		
endpoints.env.post.enabled	true	通过POST将环境更改为/ env。
endpoints.features.enabled		
endpoints.features.id		
endpoints.features.sensitive		
endpoints.pause.enabled	true	启用/暂停端点（发送Lifecycle.stop（））。
endpoints.pause.id		
endpoints.pause.sensitive		
endpoints.refresh.enabled	true	启用/ refresh端点刷新配置并重新初始化刷新作用域bean。
endpoints.refresh.id		
endpoints.refresh.sensitive		
endpoints.restart.enabled	true	启用/ restart端点重新启动应用程序上下文。
endpoints.restart.id		
endpoints.restart.pause-endpoint.enabled		
endpoints.restart.pause-endpoint.id		
endpoints.restart.pause-endpoint.sensitive		
endpoints.restart.resume-endpoint.enabled		
endpoints.restart.resume-endpoint.id		
endpoints.restart.resume-endpoint.sensitive		
endpoints.restart.sensitive		
endpoints.restart.timeout	0	

endpoints.resume.enabled	true	启用/ resume端点（发送Lifecycle.start（））。
endpoints.resume.id		
endpoints.resume.sensitive		
endpoints.zookeeper.enabled	true	启用/ zookeeper端点来检查zookeeper的状态。

spring.cloud.bus

属性名	默认值	说明
spring.cloud.bus.ack.destination-service		想要听ack的服务。默认为null（表示所有服务）。
spring.cloud.bus.ack.enabled	true	标志关闭acks（默认打开）。
spring.cloud.bus.destination	springCloudBus	名称Spring Cloud消息的流目的地。
spring.cloud.bus.enabled	true	标志表示总线已启用。
spring.cloud.bus.env.enabled	true	标志关闭环境变化事件（默认为开）。
spring.cloud.bus.refresh.enabled	true	关闭刷新事件的标志（默认为开）。
spring.cloud.bus.trace.enabled	false	打开acks跟踪的标志（默认关闭）。

spring.cloud.cloudfoundry

属性名	默认值	说明
spring.cloud.cloudfoundry.discovery.enabled	true	标记以指示启用发现。
spring.cloud.cloudfoundry.discovery.heartbeat-frequency	5000	心跳次数以毫秒为单位的频率。客户端将轮询该频率并广播服务ID列表。
spring.cloud.cloudfoundry.discovery.org		要进行身份验证的组织名称（默认为用户默认值）。
spring.cloud.cloudfoundry.discovery.password		用户验证和获取令牌的密码。
spring.cloud.cloudfoundry.discovery.space		要进行身份验证的空间名称（默认为用户默认值）。
spring.cloud.cloudfoundry.discovery.url	https://api.run.pivotal.io	Cloud Foundry API（云控制器）的URL。
spring.cloud.cloudfoundry.discovery.username		验证用户名（通常是电子邮件地址）。

spring.cloud.config

属性名	默认值	说明
spring.cloud.config.allow-override	true	标记以指示可以使用{@link #isSystemPropertiesOverride（）systemPropertiesOverride}。设置为false以防止用户意外更改默认值。默认值为true。
spring.cloud.config.authorization		客户端使用的授权令牌连接到服务器。
spring.cloud.config.discovery.enabled	false	标记以指示启用配置服务器发现（配置服务器URL将通过发现查找）。
spring.cloud.config.discovery.service-id	configserver	服务ID来定位配置服务器。
spring.cloud.config.enabled	true	标记说远程配置启用。默认为true;
spring.cloud.config.fail-fast	false	标记表示无法连接到服务器是致命的（默认为false）。

spring.cloud.config.label		用于拉取远程配置属性的标签名称。默认设置在服务器上（通常是基于git的服务器的“主”）。
spring.cloud.config.name		用于获取远程属性的应用程序名称。
spring.cloud.config.override-none	false	标志表示当{@link #setAllowOverride (boolean) allowOverride}为true时，外部属性应该采用最低优先级，并且不覆盖任何现有的属性源（包括本地配置文件）。默认为false。
spring.cloud.config.override-system-properties	true	标记以指示外部属性应覆盖系统属性。默认值为true。
spring.cloud.config.password		联系远程服务器时使用的密码（HTTP Basic）。
spring.cloud.config.profile	default	获取远程配置时使用的默认配置文件（逗号分隔）。默认为“默认”。
spring.cloud.config.retry.initial-interval	1000	初始重试间隔（以毫秒为单位）。
spring.cloud.config.retry.max-attempts	6	最大尝试次数。
spring.cloud.config.retry.max-interval	2000	退避的最大间隔
spring.cloud.config.retry.multiplier	1.1	下一个间隔的乘数。
spring.cloud.config.server.bootstrap	false	表示配置服务器应使用远程存储库中的属性初始化其自己的环境。默认情况下关闭，因为它会延迟启动，但在将服务器嵌入到另一个应用程序中时很有用。
spring.cloud.config.server.default-application-name	application	传入请求没有特定的默认应用程序名称。
spring.cloud.config.server.default-label		传入请求没有特定标签时的默认存储库标签。
spring.cloud.config.server.default-profile	default	传入请求没有特定的默认应用程序配置文件时。
spring.cloud.config.server.encrypt.enabled	true	在发送给客户端之前启用对环境属性的解密。
spring.cloud.config.server.git.basedir		库的本地工作副本的基本目录。
spring.cloud.config.server.git.clone-on-start		标记以表明应该在启动时克隆存储库（不是按需）。通常会导致启动速度较慢，但第一次查询更快。
spring.cloud.config.server.git.default-label		
spring.cloud.config.server.git.environment		
spring.cloud.config.server.git.force-pull		标记表示存储库应该强制拉。如果真的丢弃任何本地更改并从远程存储库获取。
spring.cloud.config.server.git.git-factory		
spring.cloud.config.server.git.password		使用远程存储库验证密码。
spring.cloud.config.server.git.repos		存储库标识符映射到位置和其他属性。
spring.cloud.config.server.git.search-paths		在本地工作副本中使用的搜索路径。默认情况下只搜索根。
spring.cloud.config.server.git.timeout		用于获取HTTP或SSH连接的超时（以秒为单位）（如果适用）。默认5秒。
spring.cloud.config.server.git.uri		远程存储库的URI。
spring.cloud.config.server.git.username		用于远程存储库的身份验证用户名。
spring.cloud.config.server.health.repositories		
spring.cloud.config.server.native.fail-on-error	false	标识以确定在解密期间如何处理异常（默认为false）。
spring.cloud.config.server.native.search-locations	[]	搜索配置文件的位置。默认与Spring Boot应用程序相同，因此 [classpath: /, classpath: / config /, file: ./, file: ./ config /]。

spring.cloud.config.server.native.version		为本地存储库报告的版本字符串
spring.cloud.config.server.overrides		无条件发送给所有客户的资源的额外地图。
spring.cloud.config.server.prefix		配置资源路径的前缀（默认为空）。当您不想更改上下文路径或servlet路径时嵌入其他应用程序时很有用。
spring.cloud.config.server.strip-document-from-yaml	true	标记为指示作为文本或集合（而不是映射）的YAML文档应以“本机”形式返回。
spring.cloud.config.server.svn.basedir		库的本地工作副本的基本目录。
spring.cloud.config.server.svn.default-label	trunk	用于环境属性请求的默认标签。
spring.cloud.config.server.svn.environment		
spring.cloud.config.server.svn.password		使用远程存储库验证密码。
spring.cloud.config.server.svn.search-paths		在本地工作副本中使用的搜索路径。默认情况下只搜索根。
spring.cloud.config.server.svn.uri		远程存储库的URI。
spring.cloud.config.server.svn.username		用于远程存储库的身份验证用户名。
spring.cloud.config.token		安全令牌通过到底层环境库。
spring.cloud.config.uri	http://localhost:8888	远程服务器的URI（默认http: // localhost: 8888）。
spring.cloud.config.username		联系远程服务器时使用的用户名（HTTP Basic）。
spring.cloud.consul.config.acl-token		
spring.cloud.consul.config.data-key	data	如果格式为Format.PROPERTIES或Format.YAML，则使用以下字段来查找协调配置。
spring.cloud.consul.config.default-context	application	
spring.cloud.consul.config.enabled	true	
spring.cloud.consul.config.fail-fast	true	在配置查找期间抛出异常，如果为true，否则为日志警告。
spring.cloud.consul.config.format		
spring.cloud.consul.config.prefix	config	
spring.cloud.consul.config.profile-separator	,	
spring.cloud.consul.config.watch.delay	1000	手表的固定延迟的价值在毫秒。默认为1000。
spring.cloud.consul.config.watch.enabled	true	如果手表启用 默认为true。
spring.cloud.consul.config.watch.wait-time	55	等待（或阻止）观看查询的秒数，默认为55.需要小于默认的ConsulClient（默认为60）。要增加ConsulClient超时，使用自定义的HttpClient创建一个带有自定义ConsulRawClient的ConsulClient bean。

spring.cloud.consul

属性名	默认值	说明
spring.cloud.consul.discovery.acl-token		
spring.cloud.consul.discovery.catalog-services-watch-delay	10	
spring.cloud.consul.discovery.catalog-services-watch-timeout	2	

spring.cloud.consul.discovery.default-query-tag		如果没有在serverListQueryTags中列出，请在服务列表中查询标签。
spring.cloud.consul.discovery.default-zone-metadata-name	zone	服务实例区域来自元数据。这允许更改元数据标签名称。
spring.cloud.consul.discovery.enabled	true	是否启用服务发现
spring.cloud.consul.discovery.fail-fast	true	在服务注册期间抛出异常，如果为true，否则，记录警告（默认为true）。
spring.cloud.consul.discovery.health-check-interval	10	执行健康检查的频率（例如10s）
spring.cloud.consul.discovery.health-check-path	/health	调用健康检查的备用服务器路径
spring.cloud.consul.discovery.health-check-timeout	10	健康检查超时（例如10s）
spring.cloud.consul.discovery.health-check-url		自定义健康检查网址覆盖默认值
spring.cloud.consul.discovery.heartbeat.enabled	false	
spring.cloud.consul.discovery.heartbeat.heartbeat-interval		
spring.cloud.consul.discovery.heartbeat.interval-ratio		
spring.cloud.consul.discovery.heartbeat.ttl-unit	s	
spring.cloud.consul.discovery.heartbeat.ttl-value	30	
spring.cloud.consul.discovery.host-info		
spring.cloud.consul.discovery.hostname		访问服务器时使用的主机名
spring.cloud.consul.discovery.instance-id		唯一的服务实例ID
spring.cloud.consul.discovery.instance-zone		服务实例区
spring.cloud.consul.discovery.ip-address		访问服务时使用的IP地址（还必须设置preferIpAddress才能使用）
spring.cloud.consul.discovery.lifecycle.enabled	true	
spring.cloud.consul.discovery.management-port		端口注册管理服务（默认为管理端口）
spring.cloud.consul.discovery.management-suffix	management	注册管理服务时使用后缀
spring.cloud.consul.discovery.management-tags		注册管理服务时使用的标签
spring.cloud.consul.discovery.port		端口注册服务（默认为侦听端口）
spring.cloud.consul.discovery.prefer-agent-address	false	我们将如何确定使用地址的来源
spring.cloud.consul.discovery.prefer-ip-address	false	在注册时使用ip地址而不是主机名
spring.cloud.consul.discovery.query-passing	false	将“pass”参数添加到/ v1 / health / service / serviceName。这会将健康检查推送到服务器。
spring.cloud.consul.discovery.register	true	注册为领事服务。
spring.cloud.consul.discovery.register-health-check	true	注册领事健康检查。在开发服务期间有用。
spring.cloud.consul.discovery.scheme	http	是否注册http或https服务
spring.cloud.consul.discovery.server-list-query-tags		服务器列表中要查询的serviceId的→标签的映射。这允许通过单个标签过滤服务。
spring.cloud.consul.discovery.service-name		服务名称
spring.cloud.consul.discovery.tags		注册服务时使用的标签
spring.cloud.consul.enabled	true	启用了spring cloud consul

spring.cloud.consul.host	localhost	Consul代理主机名。默认为"localhost"。
spring.cloud.consul.port	8500	Consul代理端口。默认为'8500'。
spring.cloud.consul.retry.initial-interval	1000	初始重试间隔（以毫秒为单位）。
spring.cloud.consul.retry.max-attempts	6	最大尝试次数。
spring.cloud.consul.retry.max-interval	2000	退避的最大间隔
spring.cloud.consul.retry.multiplier	1.1	下一个间隔的乘数。

spring.cloud.hypermedia

属性名	默认值	说明
spring.cloud.hypermedia.refresh.fixed-delay	5000	
spring.cloud.hypermedia.refresh.initial-delay	10000	

spring.cloud.inetutils

属性名	默认值	说明
spring.cloud.inetutils.default-hostname	localhost	默认主机名。用于发生错误的情况。
spring.cloud.inetutils.default-ip-address	127.0.0.1	默认ipaddress。用于发生错误的情况。
spring.cloud.inetutils.ignored-interfaces		将被忽略的网络接口的Java正则表达式列表。
spring.cloud.inetutils.preferred-networks		将被忽略的网络地址的Java正则表达式列表。
spring.cloud.inetutils.timeout-seconds	1	计算主机名的超时秒数。
spring.cloud.inetutils.use-only-site-local-interfaces	false	仅使用与站点本地地址的接口。有关详细信息，请参阅{@link InetAddress # isSiteLocalAddress () }。

spring.cloud.loadbalancer

属性名	默认值	说明
spring.cloud.loadbalancer.retry.enabled	false	

spring.cloud.stream

属性名	默认值	说明
spring.cloud.stream.binders		
spring.cloud.stream.bindings		
spring.cloud.stream.consul.binder.event-timeout	5	
spring.cloud.stream.consumer-defaults		
spring.cloud.stream.default-binder		
spring.cloud.stream.dynamic-destinations	[]	
spring.cloud.stream.ignore-unknown-properties	true	
spring.cloud.stream.instance-count	1	

spring.cloud.stream.instance-index	0	
spring.cloud.stream.producer-defaults		
spring.cloud.stream.rabbit.binder.admin-adresses	[]	
spring.cloud.stream.rabbit.binder.compression-level	0	
spring.cloud.stream.rabbit.binder.nodes	[]	
spring.cloud.stream.rabbit.bindings		

spring.cloud.zookeeper

属性名	默认值	说明
spring.cloud.zookeeper.base-sleep-time-ms	50	重试之间等待的初始时间
spring.cloud.zookeeper.block-until-connected-unit		与Zookeeper连接时阻止的时间单位
spring.cloud.zookeeper.block-until-connected-wait	10	等待时间阻止连接到Zookeeper
spring.cloud.zookeeper.connect-string	localhost:2181	连接字符串到Zookeeper集群
spring.cloud.zookeeper.default-health-endpoint		将检查以验证依赖关系是否存在的默认运行状况端点
spring.cloud.zookeeper.dependencies		将别名映射到ZookeeperDependency。从Ribbon的角度看，别名实际上是serviceID，因为Ribbon不能接受serviceID中的嵌套结构
spring.cloud.zookeeper.dependency-configurations		
spring.cloud.zookeeper.dependency-names		
spring.cloud.zookeeper.discovery.enabled	true	
spring.cloud.zookeeper.discovery.instance-host		预定义的主机可以在Zookeeper中注册自己的服务。对应于URI规范中的{code address}。
spring.cloud.zookeeper.discovery.instance-port		端口注册服务（默认为侦听端口）
spring.cloud.zookeeper.discovery.metadata		获取与此实例关联的元数据名称/值对。该信息被发送到动物管理员，可以被其他实例使用。
spring.cloud.zookeeper.discovery.register	true	在zookeeper中注册为服务。
spring.cloud.zookeeper.discovery.root	/services	所有实例都被注册的根Zookeeper文件夹
spring.cloud.zookeeper.discovery.uri-spec	{scheme}://{address}:{port}	在Zookeeper服务注册期间解决的URI规范
spring.cloud.zookeeper.enabled	true	启用Zookeeper
spring.cloud.zookeeper.max-retries	10	最多重试次数
spring.cloud.zookeeper.max-sleep-ms	500	每次重试最多可以以ms为单位睡眠
spring.cloud.zookeeper.prefix		将应用于所有Zookeeper依赖关系的路径的公共前缀

spring.integration.poller

属性名	默认值	说明
spring.integration.poller.fixed-delay	1000	修复默认轮询器的延迟。
spring.integration.poller.max-messages-per-poll	1	默认轮询器每轮询的最大消息。
spring.sleuth.integration.enabled	true	启用Spring Integration侦察器。
spring.sleuth.integration.patterns	*	一组简单的模式，通道名称将与之匹配。默认值为*（所有通道）。请参阅org.springframework.util.PatternMatchUtils.simpleMatch（String，String）。

spring.sleuth.keys

属性名	默认值	说明
spring.sleuth.keys.async.class-name-key	class	具有使用{@code @Async} 注释的方法的类的简单名称，从异步进程开始
spring.sleuth.keys.async.method-name-key	method	使用{@code @Async} 注释的方法的名称
spring.sleuth.keys.async.prefix		如果标题名称被添加为标签，则使用前缀。
spring.sleuth.keys.async.thread-name-key	thread	执行异步方法的线程的名称
spring.sleuth.keys.http.headers		额外的标题应该作为标签添加，如果它们存在。如果头值是多值的，则标签值将是逗号分隔的单引号列表。
spring.sleuth.keys.http.host	http.host	URL或主机头的域部分。示例：“mybucket.s3.amazonaws.com”。用于过滤主机而不是ip地址。
spring.sleuth.keys.http.method	http.method	HTTP方法或动词，如“GET”或“POST”。用于过滤http路由。
spring.sleuth.keys.http.path	http.path	绝对的http路径，没有任何查询参数。示例：“/ objects / abcd-ff”。 用于过滤http路由，可以与zipkin v1一起移植。 在zipkin v1中，只支持等于过滤器。删除查询参数使不同URI的数量减少。 例如，无论查询行中编码的签名参数如何，都可以查询相同的资源。 这不会降低HTTP单路由的基数。例如，通常将路由表示为http URI模板，如“/ resource / {resource_id}”。 在只有等量查询可用的系统中，如果实际请求是“/ resource / abcd-ff”，则搜索{@code http.uri = / resource}将不匹配。 历史记录：这通常在拉链中被表示为“http.uri”，但最常见的只是一条路。
spring.sleuth.keys.http.prefix	http.	如果标题名称被添加为标签，则使用前缀。
spring.sleuth.keys.http.request-size	http.request.size	非空HTTP请求体的大小（以字节为单位）。防爆。“16384” 大上传可能会超出限制或直接影响延迟。
spring.sleuth.keys.http.response-size	http.response.size	非空HTTP响应体的大小（以字节为单位）。防爆。“16384” 大量下载可能会超出限制或直接影响延迟。
spring.sleuth.keys.http.status-code	http.status_code	当HTTP响应代码不在2xx范围内。防爆。“503”用于过滤错误状态。 2xx范围不会被记录，因为成功代码对延迟故障排除不那么有趣。省略节省每个跨度至少20个字节。
spring.sleuth.keys.http.url	http.url	整个URL，包括方案，主机和查询参数（如果可用）。防爆。
spring.sleuth.keys.hystrix.command-group	commandGroup	命令组的名称 Hystrix使用命令组密钥将诸如报告，警报，仪表板或团队/库所有权的命令分组在一起。

spring.sleuth.keys.hystrix.command-key	commandKey	命令键的名称 描述给定命令的名称。代表用于监视，断路器，指标发布，缓存和其他此类用途的{@link com.netflix.hystrix.HystrixCommand}的关键。
spring.sleuth.keys.hystrix.prefix		如果标题名称被添加为标签，则使用前缀。
spring.sleuth.keys.hystrix.thread-pool-key	threadPoolKey	线程池密钥的名称。线程池密钥表示用于监视，指标发布，缓存和其他此类用途的
spring.sleuth.keys.message.headers		额外的标题应该作为标签添加，如果它们存在。如果值不是String，它将使用其toString () 方法转换为String。
spring.sleuth.keys.message.payload.size	message/payload-size	估计有效载荷的大小（如果有的话）。
spring.sleuth.keys.message.payload.type	message/payload-type	有效载荷的类型。
spring.sleuth.keys.message.prefix	message/	如果标题名称被添加为标签，则使用前缀。
spring.sleuth.keys.mvc.controller-class	mvc.controller.class	小写，连字符分隔处理请求的类的名称。防爆。名为“BookController”的类将导致“book-controller”标签值。
spring.sleuth.keys.mvc.controller-method	mvc.controller.method	小写，连字符分隔处理请求的类的名称。防爆。名为“listOfBooks”的方法将导致“list-of-books”标签值。

spring.sleuth.metric

属性名	默认值	说明
spring.sleuth.metric.span.accepted-name	counter.span.accepted	
spring.sleuth.metric.span.dropped-name	counter.span.dropped	
spring.sleuth.sampler.percentage	0.1	应采样的请求百分比。例如1.0 - 100%的请求应该被抽样。精度仅为全数（即不支持0.1%的痕迹）。
spring.sleuth.trace-id128	false	如果为true，则生成128位跟踪ID，而不是64位跟踪ID。

图片格式预览：

eureka

属性名	默认值	说明
eureka.client.allow-redirects	false	指示服务器是否可以将客户端请求重定向到备份服务器/集群。 如果设置为false，服务器将直接处理请求，如果设置为true，则可能会向客户端发送HTTP重定向，并具有新的服务器位置。
eureka.client.availability-zones		获取此实例所在区域的可用性区域列表（用于AWS数据中心）。 更改在运行时在registryFetchIntervalSeconds指定的下一个注册表提取周期中有效。
eureka.client.backup-registry-impl		获取执行BackupRegistry的实现名称，以便仅在eureka客户端启动时首次将注册表信息作为回退选项提取。 对于需要额外的弹性的注册表信息的应用程序可能需要这一点，而无法运行它们。
eureka.client.cache-refresh-executor-exponential-back-off-bound	10	缓存刷新执行者指数退出相关属性。在发生超时序列的情况下，它是重试延迟的最大乘数值。
eureka.client.cache-refresh-executor-thread-pool-size	2	cacheRefreshExecutor初始化的线程池大小
eureka.client.client-data-accept		EurekaAccept客户端数据接受名称
eureka.client.decoder-name		这是一个瞬态配置，一旦最新的编解码器稳定，可以删除（因为只有一个）
eureka.client.disable-delta	false	指示eureka客户端是否应该禁用提取delta，而应该诉诸于获取完整的注册表信息。 请注意，增量获取可以极大地减少流量，因为尤利卡服务器的更改速率通常远低于提取速率。 更改在运行时在registryFetchIntervalSeconds指定的下一个注册表提取周期中有效
eureka.client.dollar-replacement	“_”	在eureka服务器的序列化/反序列化信息期间，获取Dollar符号<code>\$</code>的替换字符串。
eureka.client.enabled	true	标记以指示启用Eureka客户端。
eureka.client.encoder-name		这是一个瞬态配置，一旦最新的编解码器稳定，可以删除（因为只有一个）
eureka.client.escape-char-replacement	“_”	在eureka服务器的序列化/反序列化信息期间获取下划线符号<code>_</code>的替换字符串。
eureka.client.eureka-connection-idle-timeout-seconds	30	表示到eureka服务器的HTTP连接可以在关闭之前保持空闲状态的时间（以秒为单位）。 在AWS环境中，建议值为30秒或更短，因为防火墙在几分钟内清除连接信息，将连接挂在空中
eureka.client.eureka-server-connect-timeout-seconds	5	指示在连接到eureka服务器需要超时之前等待（以秒为单位）的时间。请注意，客户端中的连接由org.apache.http.client.HttpClient 汇集，此设置会影响实际的连接创建以及从池中获取连接的等待时间。
eureka.client.eureka-server-d-n-s-name		获取要查询的DNS名称以获取eureka服务器的列表。如果合同通过实现serviceUrls返回服务URL，则不需要此信息。 当useDnsForFetchingServiceUrls设置为true时，使用DNS机制，而eureka客户端希望DNS以某种方式配置，以便可以动态获取更改的eureka服务器。 更改在运行时有效。
eureka.client.eureka-server-port		获取用于构建服务url的端口，以在eureka服务器列表来自DNS时联系eureka服务器。如果合同返回服务url eurekaServerServiceUrls（String），则不需要此信息。 当useDnsForFetchingServiceUrls设置为true时，使用DNS机制，而eureka客户端希望DNS以某种方式配置，以便可以动态获取更改的eureka服务器。 更改在运行时有效。