

Spring Cloud Bus：消息总线

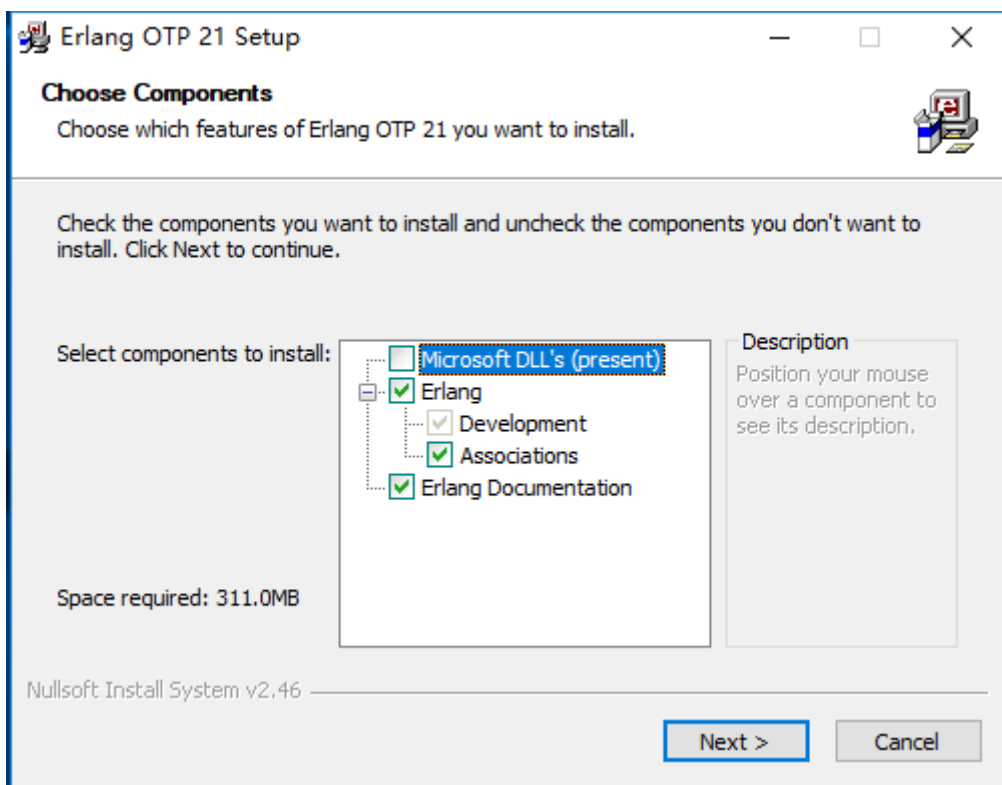
Spring Cloud Bus 使用轻量级的消息代理来连接微服务架构中的各个服务，可以将其用于广播状态更改（例如配置中心配置更改）或其他管理指令，本文将对其用法进行详细介绍。

Spring Cloud Bus 简介

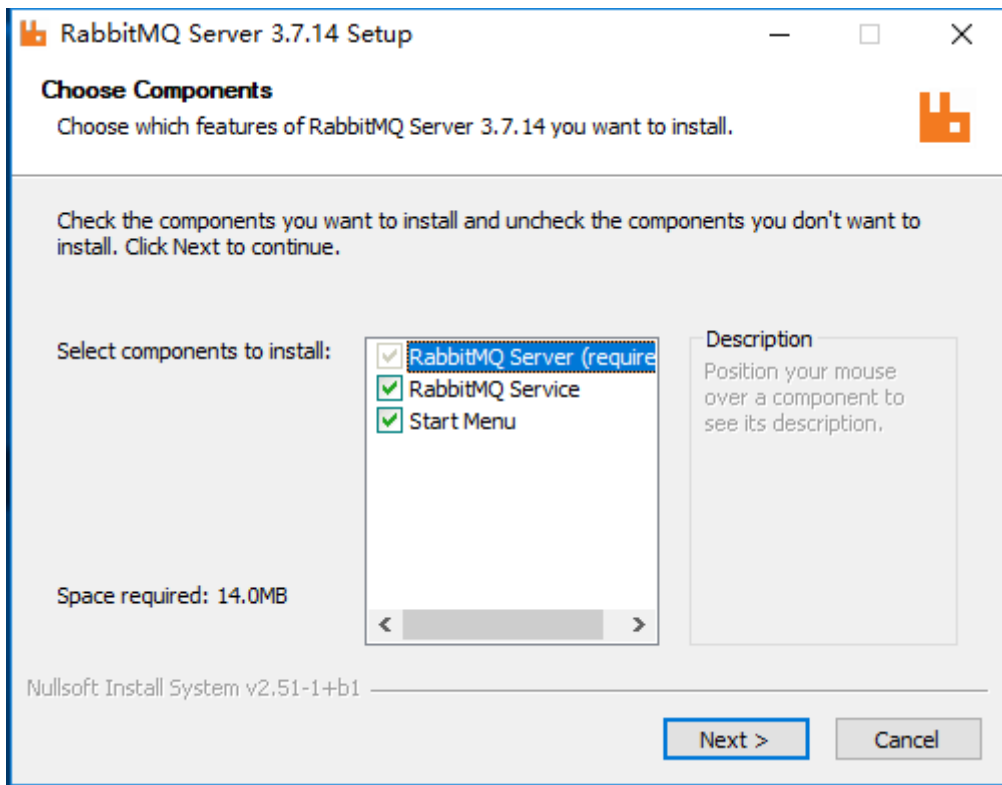
我们通常会使用消息代理来构建一个主题，然后把微服务架构中的所有服务都连接到这个主题上去，当我们向该主题发送消息时，所有订阅该主题的服务都会收到消息并进行消费。使用 Spring Cloud Bus 可以方便地构建起这套机制，所以 Spring Cloud Bus 又被称为消息总线。Spring Cloud Bus 配合 Spring Cloud Config 使用可以实现配置的动态刷新。目前 Spring Cloud Bus 支持两种消息代理：RabbitMQ 和 Kafka，下面以 RabbitMQ 为例来演示下使用 Spring Cloud Bus 动态刷新配置的功能。

RabbitMQ的安装

- 安装Erlang，下载地址：http://erlang.org/download/otp_win64_21.3.exe



- 安装RabbitMQ，下载地址：<https://dl.bintray.com/rabbitmq/all/rabbitmq-server/3.7.14/rabbitmq-server-3.7.14.exe>



- 安装完成后，进入RabbitMQ安装目录下的sbin目录：

台电脑 > 本地磁盘 (D:) > developer > env > RabbitMQ Server > rabbitmq_server-3.7.14 > sbin

名称	修改日期	类型	大小
cuttlefish	2019/3/29 5:47	文件	458 KB
rabbitmqctl.bat	2019/3/29 5:47	Windows 批处理...	3 KB
rabbitmq-defaults.bat	2019/3/29 5:47	Windows 批处理...	2 KB
rabbitmq-diagnostics.bat	2019/3/29 5:47	Windows 批处理...	3 KB
rabbitmq-echopid.bat	2019/3/29 5:47	Windows 批处理...	2 KB
rabbitmq-env.bat	2019/3/29 5:47	Windows 批处理...	17 KB
rabbitmq-plugins.bat	2019/3/29 5:47	Windows 批处理...	3 KB
rabbitmq-server.bat	2019/3/29 5:47	Windows 批处理...	11 KB
rabbitmq-service.bat	2019/3/29 5:47	Windows 批处理...	14 KB

- 在地址栏输入cmd并回车启动命令行，然后输入以下命令启动管理功能：

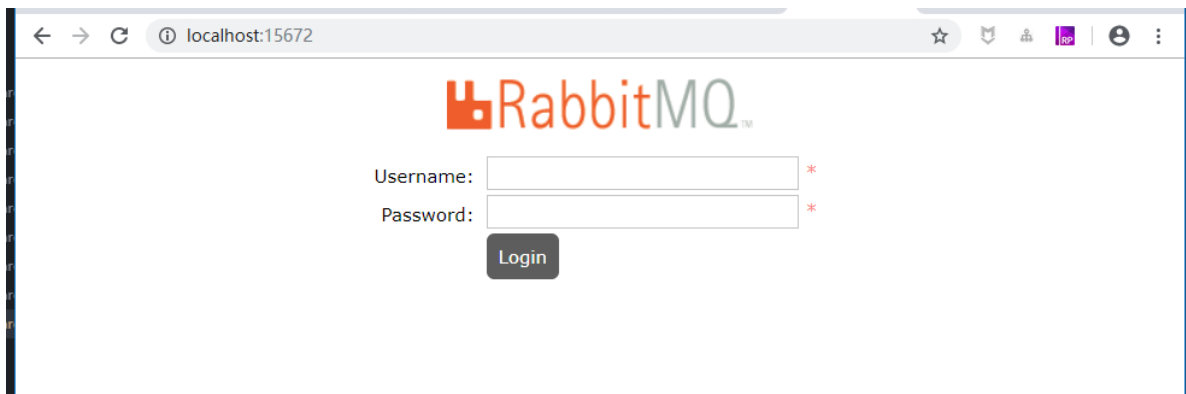
```
1 rabbitmq-plugins enable rabbitmq_managementCopy to clipboardErrorCopied
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.17134.765]
(c) 2018 Microsoft Corporation。保留所有权利。

D:\developer\env\RabbitMQ Server\rabbitmq_server-3.7.14\sbin>rabbitmq-plugins enable rabbitmq_management
Enabling plugins on node rabbit@DESKTOP-K1F707Q:
rabbitmq_management
{:plugins_not_found, [:rabbitmq_delayed_message_exchange]}

D:\developer\env\RabbitMQ Server\rabbitmq_server-3.7.14\sbin>
```

- 访问地址查看是否安装成功：<http://localhost:15672/>



- 输入账号密码并登录：guest guest

动态刷新配置

使用 Spring Cloud Bus 动态刷新配置需要配合 Spring Cloud Config 一起使用，我们使用 [上一节](#) 中的 config-server、config-client 模块来演示下该功能。

给config-server添加消息总线支持

- 在pom.xml中添加相关依赖：

```
1  <dependency>
2      <groupId>org.springframework.cloud</groupId>
3      <artifactId>spring-cloud-starter-bus-amqp</artifactId>
4  </dependency>
5  <dependency>
6      <groupId>org.springframework.boot</groupId>
7      <artifactId>spring-boot-starter-actuator</artifactId>
8  </dependency>Copy to clipboardErrorCopied
```

- 添加配置文件application-amqp.yml，主要是添加了RabbitMQ的配置及暴露了刷新配置的Actuator端点；

```
1  server:
2      port: 8904
3  spring:
4      application:
5          name: config-server
6      cloud:
7          config:
8              server:
9                  git:
10                     uri: https://gitee.com/macrozheng/springcloud-config.git
11                     username: macro
12                     password: 123456
13                     clone-on-start: true # 开启启动时直接从git获取配置
14      rabbitmq: #rabbitmq相关配置
15          host: localhost
16          port: 5672
17          username: guest
18          password: guest
19  eureka:
20      client:
```

```

21     service-url:
22         defaultZone: http://localhost:8001/eureka/
23     management:
24         endpoints: #暴露bus刷新配置的端点
25         web:
26             exposure:
27                 include: 'bus-refresh'Copy to clipboardErrorCopied

```

给config-client添加消息总线支持

- 在pom.xml中添加相关依赖：

```

1     <dependency>
2         <groupId>org.springframework.cloud</groupId>
3         <artifactId>spring-cloud-starter-bus-amqp</artifactId>
4     </dependency>Copy to clipboardErrorCopied

```

- 添加配置文件bootstrap-amqp1.yml及bootstrap-amqp2.yml用于启动两个不同的config-client，两个配置文件只有端口号不同；

```

1     server:
2         port: 9004
3     spring:
4         application:
5             name: config-client
6     cloud:
7         config:
8             profile: dev #启用环境名称
9             label: dev #分支名称
10            name: config #配置文件名称
11        discovery:
12            enabled: true
13            service-id: config-server
14        rabbitmq: #rabbitmq相关配置
15            host: localhost
16            port: 5672
17            username: guest
18            password: guest
19    eureka:
20        client:
21            service-url:
22                defaultZone: http://localhost:8001/eureka/
23    management:
24        endpoints:
25            web:
26                exposure:
27                    include: 'refresh'Copy to clipboardErrorCopied

```

动态刷新配置演示

- 我们先启动相关服务，启动eureka-server，以application-amqp.yml为配置启动config-server，以bootstrap-amqp1.yml为配置启动config-client，以bootstrap-amqp2.yml为配置再启动一个config-client，启动后注册中心显示如下：

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
CONFIG-CLIENT	n/a (2)	(2)	UP (2) - DESKTOP-K1F7O7Q:config-client:9004 , DESKTOP-K1F7O7Q:config-client:9005
CONFIG-SERVER	n/a (1)	(1)	UP (1) - DESKTOP-K1F7O7Q:config-server:8904

- 启动所有服务后，我们登录RabbitMQ的控制台可以发现Spring Cloud Bus 创建了一个叫springCloudBus的交换机及三个以springCloudBus.anonymous开头的队列：

OverviewConnectionsChannelsExchangesQueuesAdmin

Exchanges

All exchanges (19, filtered down to 1)

Pagination

Page 1 of 1 - Filter: springCloud ☐ Regex ?

Virtual host	Name	Type	Features	Message rate in	Message rate out	+/-
/	springCloudBus	topic	D	0.00/s	0.00/s	

Add a new exchange

OverviewConnectionsChannelsExchangesQueuesAdmin

Queues

All queues (7, filtered down to 3)

Pagination

Page 1 of 1 - Filter: springCloud ☐ Regex ?

Overview		Features			State			Messages			Message rates				+/-
Virtual host	Name	AD	Excl	ML		Ready	Unacked	Total	incoming	deliver / get	ack				
/	springCloudBus.anonymous.ENeR1ZzYSiysIm2Ko5r49A	AD	Excl	ML	idle	0	0	0	0.00/s	0.00/s	0.00/s				
/	springCloudBus.anonymous.apsOyA6OQAGQo1QDY7BtZg	AD	Excl	ML	idle	0	0	0	0.00/s	0.00/s	0.00/s				
/	springCloudBus.anonymous.ordmfGwJSambD3XYjLaaIg	AD	Excl	ML	idle	0	0	0	0.00/s	0.00/s	0.00/s				

- 我们先修改Git仓库中dev分支下的config-dev.yml配置文件：

```
1 # 修改前信息
2 config:
3   info: "config info for dev(dev)"
4 # 修改后信息
5 config:
6   info: "update config info for dev(dev)" Copy to clipboardErrorCopied
```

- 调用注册中心的接口刷新所有配置：<http://localhost:8904/actuator/bus-refresh>

POST http://localhost:8904/actuator/bus-refresh Send

ParamsAuthorizationHeaders (9)BodyPre-request ScriptTestsSettings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

BodyCookies (1)Headers (1)Test Results

Status: 204 No Content Time: 4.66s Size: 64 B Save

PrettyRawPreviewVisualize BETAText

1 |

- 刷新后再分别调用<http://localhost:9004/configInfo> 和 <http://localhost:9005/configInfo> 获取配置信息，发现都已经刷新了；

```
1 update config info for dev(dev)Copy to clipboardErrorCopied
```

- 如果只需要刷新指定实例的配置可以使用以下格式进行刷新：<http://localhost:8904/actuator/bus-refresh/{destination}>，我们这里以刷新运行在9004端口上的config-client为例<http://localhost:8904/actuator/bus-refresh/config-client:9004>。

配合WebHooks使用

WebHooks相当于是一个钩子函数，我们可以配置当向Git仓库push代码时触发这个钩子函数，这里以Gitee为例来介绍下其使用方式，这里当我们向配置仓库push代码时就会自动刷新服务配置了。

The screenshot shows the 'Add WebHook' page in Gitee for the repository 'macro / springcloud-config'. The interface includes a sidebar with navigation links, a main content area with a warning box, and a form for configuring the WebHook. Numbered annotations (1-4) highlight specific elements:

- 1: The '管理' (Manage) button in the top navigation bar.
- 2: The 'URL' input field containing 'http://localhost:8904/actuator/bus-refresh'.
- 3: The 'Push' checkbox under '选择事件' (Select Event).
- 4: The '添加' (Add) button at the bottom of the form.

Additional annotations include a red arrow pointing to the 'URL' field with the text: '此处需要配置公网可以访问的地址，此处的本地地址只是用作演示' (The public network address that can be accessed here needs to be configured; the local address here is only for demonstration).

使用到的模块

```
1 springcloud-learning
2 |—— eureka-server -- eureka注册中心
3 |—— config-server  -- 配置中心服务
4 |—— config-client  -- 获取配置的客户端服务
```