Cloudlink开发指南

1. **概念**

Cloudlink：万邑云链，类似邮局

appType：全局配置项，业务系统类型，示例：OMS、TMS等

appId：全局配置项，业务系统ID（不区分大写），系统类型+“.”+数据中心简码；示例：OMS.CN、TMS.CN、TMS.US。

Zone：全局配置项，数据中心唯一编码

Mq：全局配置项，消息服务器访问地址，示例guest:guest@127.0.0.1:15672

queue.block：全局配置项，消息消费时如果抛出异常，此消息是否阻塞消息队列；默认值false，移除消息，不阻塞队列。

MessageType：消息类型

MessageHandler：消息处理的监听接口

Message：消息包含消息头和消息载体，消息载体相当类似信

MessageHeaders：消息头类似信封,包含消息ID,消息发送appId,消息接收appId

Command：指令，包含指令头信息、指令载体（指令参数）。

CommandHeader：指令头信息，包含指令ID、指令名称、时间戳、接收appId。

CommandCallback：指令回调接口。

CommandExecutor：指令接收和回调消息发送的抽象类。

CommandBuilder：指令构建器，通过构建器来创建指令对象。

消息载体：消息载体是业务开发人员实际需要关心的内容。消息载体分为三类：指令执行参数，指令执行的结果，指令执行失败的异常。

指令：基于RabbitMQ的异步消息传递；包含三个动作：指令发送、指令处理、指令处理结果的回调，三者缺一不可。

1. **架构**



1. **消息投递机制**



1. **使用示例（spring集成方式）：**

示例环境：

|  |  |
| --- | --- |
| **RabbitMQ CN（中国数据中心）** | **guest:guest@172.16.3.164:15672** |
| **RabbitMQ US（美国数据中心）** | **guest:guest@172.16.3.165:15672** |
| **OMS** | appId=OMS.CN  appType=OMS  zone=IDC-CN  mq=amqp://guest:guest@172.16.3.164:5672 |
| **TMS** | appId=TMS.US  appType=TMS  zone=IDC-US  mq=amqp://guest:guest@172.16.3.165:5672 |

示例说明：

|  |  |
| --- | --- |
| **名词结束** | |
| OMS（OMS.CN） | 订单系统（appId：系统类型+“.”+中国数据中心简码） |
| TMS（TMS.US） | 运输管理系统（appId：系统类型+“.”+美国数据中心简码） |
| pickup | 指令名称（OMS向TMS发送pickup指令） |
|  |  |
| **代码类说明** | |
| PickupInfo.java | 指令执行参数类 |
| PickupReturnValue.java | 指令执行返回结果类 |
| PickupCommandExecutor.java | 指令处理类 |
| PickupCommandCallback.java | 指令回调类 |
| Cloudlink.java | 消息和指令的发送、指令处理类注册、指令回调类注册；获取消息和指令的构建器。 |
| CloudlinkFactoryBean | Spring集成工厂类，返回Cloudlink对象 |

示例代码（跨数据中心案例）：

1. pom.xml

|  |
| --- |
| <dependency>  <groupId>com.winit</groupId>  <artifactId>cloudlink</artifactId>  <version>1.0.0-SNAPSHOT</version>  </dependency> |

1. 定义指令执行参数

|  |
| --- |
| **public** **class** PickupInfo **implements** Serializable {  **public** **static** **final** String ***COMMAND\_NAME*** = "pickup";//指令名称  **private** String area;  **public** String getArea() {  **return** area;  }  **public** **void** setArea(String area) {  **this**.area = area;  }  } |

1. 定义指令返回结果

|  |
| --- |
| **public class PickupReturnValue implements Serializable {**  **private String code;**  **private String desc;**  **public String getCode() {**  **return code;**  **}**  **public void setCode(String code) {**  **this.code = code;**  **}**  **public String getDesc() {**  **return desc;**  **}**  **public void setDesc(String desc) {**  **this.desc = desc;**  **}**  **}** |

1. TMS定义指令处理类

|  |
| --- |
| @Component // 任意spring bean注解  @Lazy(**false**) // 必须配置延迟加载为false  @Block(false) // 可选注解，消息消费失败是否阻塞队列；此注解将覆盖queue.block全局配置  **public** **class** PickupCommandExecutor **extends** CommandExecutor<Command<PickupInfo>> {  @Override  **public** String getName() {  **return** PickupInfo.***COMMAND\_NAME***; // 指令名称  }  @Override  **public** void onReceive(Command<PickupInfo> command) **throws** CommandException {  System.***out***.println(">>>>>>>>>>>>>>>>>>>execute commnadExecutor");  PickupInfo pickupInfo = command.getPayload();  System.***out***.println(">>>>>>>>>>>>>>>>>>>paramter[area] : " + pickupInfo.getArea());    PickupReturnValue payload = **new** PickupReturnValue();  payload.setCode("10000001");  payload.setDesc("操作完成, from " + pickupInfo.getArea());    // 调用父类callback方法执行指令回调，此回调方法可在外部调用；开发人员根据业务需要选择性调用。  **this**.callback(command.getHeaders().getFromAppId(), payload);  }  } |

1. TMS添加集成Cloudlink的spring配置

|  |
| --- |
| <bean id=*"cloudlink"* class=*"com.winit.cloudlink.spring.CloudlinkFactoryBean"*>  <property name="configProps">  <props>  <prop key="appId">${cloudlink.appId}</prop>  <prop key="appType">${cloudlink.appType}</prop>  <prop key="zone">${cloudlink.zone}</prop>  <prop key="mq">${cloudlink.mq}</prop>  <prop key="owner">${cloudlink.owner}</prop>  <prop key="appVersion">${cloudlink.appVersion}</prop>  <prop key="organization">${cloudlink.organization}</prop>  <prop key="queue.block">${cloudlink.queue.block}</prop>  </props>  </property>  </bean> |

application.properties

|  |
| --- |
| cloudlink.appId=TMS.US  cloudlink.appType=TMS  cloudlink.zone=IDC-US  cloudlink.mq=amqp://guest:guest@172.16.3.165:5672  cloudlink.owner=winit  cloudlink.appVersion=1.0.0  cloudlink.organization=US  cloudlink.queue.block=false |

1. OMS定义指令回调类

|  |
| --- |
| @Component // 任意spring bean注解  @Lazy(**false**) // 必须配置延迟加载为false  @Block(false) // 可选注解，消息消费失败是否阻塞队列；此注解将覆盖queue.block全局配置  **public** **class** PickupCommandCallback **implements** CommandCallback<Command<PickupReturnValue>> {  @Override  **public** **void** onCallback(Command<PickupReturnValue> callbackCommand) {  CommandHeaders headers = callbackCommand.getHeaders();  PickupReturnValue value = callbackCommand.getPayload; // 指令返回结果  System.***out***.println(">>>>>>>>>>>>>>>>>>>start handler callback");  System.***out***.println(">>>>>>>>>>>>>>>>>>>code : " + value.getCode());  System.***out***.println(">>>>>>>>>>>>>>>>>>>desc : " + value.getDesc());  System.***out***.println(">>>>>>>>>>>>>>>>>>>end handler callback");  }  @Override  **public** String getCommandName() {  **return** PickupInfo.***COMMAND\_NAME***;  }  } |

1. OMS添加集成Cloudlink的spring配置

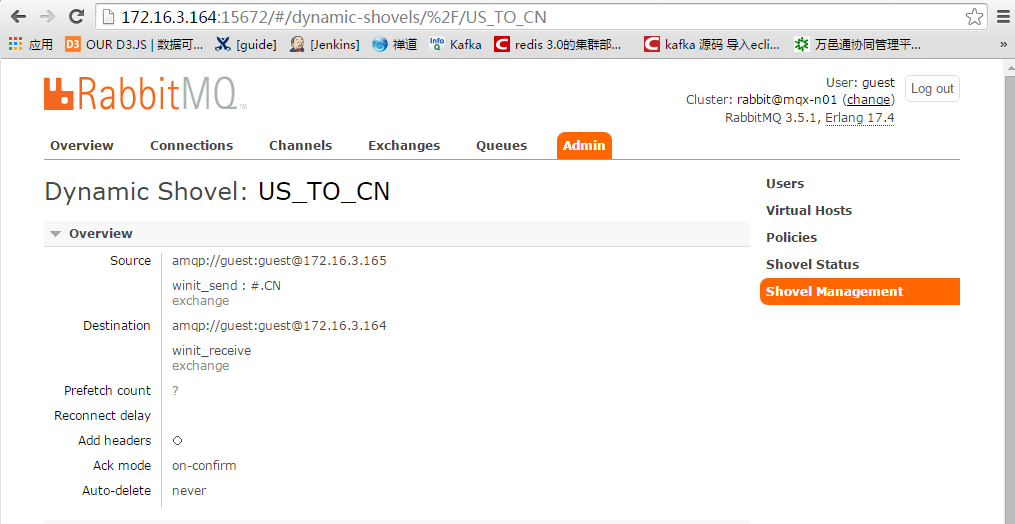
|  |
| --- |
| <bean id=*"cloudlink"* class=*"com.winit.cloudlink.spring.CloudlinkFactoryBean"*>  <property name="configProps">  <props>  <prop key="appId">${cloudlink.appId}</prop>  <prop key="appType">${cloudlink.appType}</prop>  <prop key="zone">${cloudlink.zone}</prop>  <prop key="mq">${cloudlink.mq}</prop>  <prop key="owner">${cloudlink.owner}</prop>  <prop key="appVersion">${cloudlink.appVersion}</prop>  <prop key="organization">${cloudlink.organization}</prop>  <prop key="queue.block">${cloudlink.queue.block}</prop>  </props>  </property>  </bean> |

application.properties

|  |
| --- |
| cloudlink.appId=OMS.CN  cloudlink.appType=OMS  cloudlink.zone=IDC-CN  cloudlink.mq=amqp://guest:guest@172.16.3.164:5672  cloudlink.owner=winit  cloudlink.appVersion=1.0.0  cloudlink.organization=CN  cloudlink.queue.block=false |

1. 跨数据中心配置消息复制





1. OMS发送pickup指令

|  |
| --- |
| **public** **class** CloudlinkSpringTest {  @Resource  **private** Cloudlink cloudlink;    @Test  **public** **void** testSpringDemo() {  // 创建指令参数  PickupInfo payload = **new** PickupInfo();  payload.setArea("to US");  // 构建指令  Command<PickupInfo> command = cloudlink.newCommandBuilder()  .commandName(PickupInfo.***COMMAND\_NAME***)  .toAppId("TMS.US") // 必须与目标系统的appId保持一致  .payload(payload)  .build();  // 发送指令 to CN  cloudlink.commitCommand(command);  }  } |

1. **注意事项（非常重要）**
2. **在集成Spring时，**实现以下接口（MessageHandler、CommandExecutor、CommandCallback）的bean对象必须添加“@Lazy(false)”的注解。

@Lazy(false)的作用：取消延迟加载。

1. **消息消费机制：**

有两种机制：1、消息处理失败，阻塞消息队列。2、消息处理失败，不阻塞消息队列。

默认机制是不阻塞消息队列，无论消息是否处理成功，消息都将被消费掉（也就是每条消息一次性消费完成，无论消息接收者是否处理成功）。

**消息消费机制有两种配置方式：**全局配置和私有配置；私有配置将会覆盖全局配置。

全局配置：CloudlinkFactoryBean增加queue.block配置项，取值类型：boolean。

私有配置：为不同消息接收者设定不同的消费机制，在消息处理类上添加@Block注解，取值类型：boolean；

消息接收者包含（MessageHandler、CommandExecutor、CommandCallback）。

1. **系统升级时不要修改指令（消息）参数和返回值对象的名称和包路径。**

如果修改了名称和包路径，系统升级后在消费升级前的遗留指令（消息）数据时，将导致消息内容反序列化失败。

1. **指令（消息）参数java bean需提供默认构造函数并实现序列化接口。**