

```
template:
  exchange: tacocloud.orders
  routing-key: kitchens.central
```

在本例中，所有未指明 Exchange 的消息都将会自动发送至名为 tacocloud.orders 的 Exchange。如果在调用 send()或 convertAndSend()的时候也没有指定 routing key，那么消息将会使用值为 kitchens.central 的 routing key。

通过消息转换器创建 Message 对象是非常简单的，但是使用 convertAndSend()让 RabbitTemplate 处理所有的转换操作则会更加简单：

```
public void sendOrder(Order order) {
    rabbit.convertAndSend("tacocloud.order", order);
}
```

配置消息转换器

默认情况下，消息转换是通过 SimpleMessageConverter 来实现的，它能够将简单类型（如 String）和 Serializable 对象转换成 Message 对象。但是，Spring 为 RabbitTemplate 提供了多个消息转换器，包括下面内容。

- Jackson2JsonMessageConverter：使用 Jackson 2 JSON 实现对象和 JSON 的相互转换。
- MarshallingMessageConverter：使用 Spring 的 Marshaller 和 Unmarshaller 进行转换。
- SerializerMessageConverter：使用 Spring 的 Serializer 和 Deserializer 转换 String 和任意种类的原生对象。
- SimpleMessageConverter：转换 String、字节数组和 Serializable 类型。
- ContentTypeDelegatingMessageConverter：基于 contentType 头信息，将转换功能委托给另外一个 MessageConverter。
- MessagingMessageConverter：将消息转换功能委托给另外一个 MessageConverter，并将头信息的转换委托给 AmqpHeaderConverter。

如果需要变更消息转换器，就要配置一个类型为 MessageConverter 的 bean。例如，对于基于 JSON 的转换，我们可以按照如下的方式来配置 Jackson2JsonMessageConverter：

```
@Bean
public MessageConverter messageConverter() {
    return new Jackson2JsonMessageConverter();
}
```

Spring Boot 的自动配置功能会发现这个 bean，并将它注入 RabbitTemplate 中，替换默认的消息转换器。