# Messages, events and commands

### Validation

* Commands 不支持 Pub/Sub
* Events 有多个接收者，可以被published
* ……

## Designing messages

POCO，尽量小，单一职责

# Message Identity

**var** options = **new** SendOptions();

options.SetMessageId(messageId);

**await** handlerContext.Send(**new** SomeMessage(), options) .ConfigureAwait(false);

# Messages as Interfaces

## Sending interface messages

## Publishing interface messages

## Creating interface messages with IMessageCreator

# Conventions

定义Message 的限制条件

# Unobtrusive Mode Messages

# Immutable Messages

不可变Message

# Enforcement of best practices

# routing.DoNotEnforceBestPractices();

# Evolving Message Contracts

# Sending messages

## Outside a message handler // endpointInstance.Send

## Inside the incoming message processing pipeline //context.Send

## Overriding the default routing // SetDestination, RouteToSpecificInstance

## Sending to self *//*SendLocal, RouteToThisInstance

## Influencing the reply behavior

## // RouteReplyToThisInstance, RouteReplyToAnyInstance, RouteReplyTo

## Dispatching a message immediately // RequireImmediateDispatch

# Replying to a Message

# Client-Side Callbacks

* NServiceBus.Callbacks NuGet package
* endpointConfiguration.EnableCallbacks();

### Int, Enum, Object

## Cancellation

## Message routing endpointConfiguration.MakeInstanceUniquelyAddressable("uniqueId");

# Uniform Session

# Discarding Old Messages

## Using an Attribute

## Using a custom convention

## Clock synchronization issues

## Discarding messages at startup

## Discarding messages at startup endpointConfiguration.PurgeOnStartup(true);

# Non-Durable Messaging

# Message forwarding

## Auditing vs Fowarding

* endpointConfiguration.ForwardReceivedMessagesTo("destinationQueue@machine");
* **await** context.ForwardCurrentMessageTo("destinationQueue@machine") .ConfigureAwait(false);

# Delayed Delivery \*

# Timeout Manager

# 发布-订阅

### Message-driven (persistence-based)

# 

# 

### Native

# Event

# Controlling What Is Subscribed

## 自动订阅

1. Defined as an event either using IEvent or by the .DefiningEventsAs convention.
2. At least one [message handler and/or saga](https://docs.particular.net/nservicebus/handlers/) exists for the given event.

### Exclude event types/sagas from auto-subscribe

当订阅者停止，他会继续接受event，当他再开启，会继续消费event

禁用自动订阅

手动订阅event

# Message Headers

## Timestamp format DateTimeExtensions

### NServiceBus.ContentType text/xml or text/json

### NServiceBus.EnclosedMessageTypes

### NServiceBus.MessageId message 唯一Id

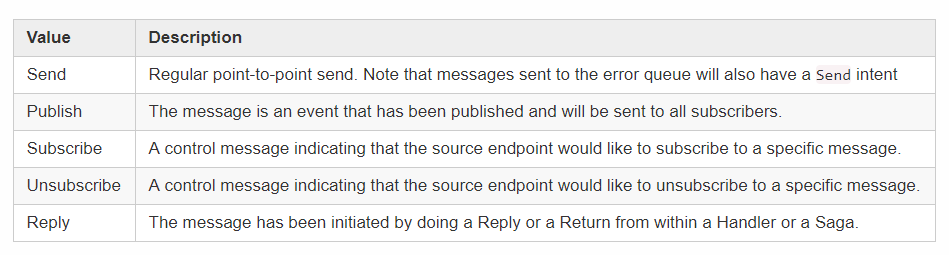
### NServiceBus.CorrelationId message 关联Id

### NServiceBus.ConversationId 一系列对话关联的Id

### NServiceBus.RelatedTo

### NServiceBus.MessageIntent

### NServiceBus.MessageIntent



## Send headers

## Reply headers

## Publish headers

## Timeout headers

## Saga-related headers

。。。。。。。。。

# Manipulating message headers

## Reading incoming headers

## Writing outgoing headers

### For all outgoing messages

endpointConfiguration.AddHeaderToAllOutgoingMessages("MyGlobalHeader", "some static value");

# Specify Endpoint Name

## Input queue

# Message routing

logical and physical 两部分组成，logical定义哪个endpoint接受什么message，physical定义哪个endpoint实例接受哪个message

## Command routing

发送到一个endpoint

### Overriding the destination

## Event routing

被多个endpoint接收， 但如果一个endpoint有多个实例，只有一个实例会接收message。Endpoint需要一个message handler

## Reply routing

基于ReplyTo header，与endpoint's routing configuration无关，只有最初的message的发布者可以影响reply routing

# Routing system extensibility points

## Routing APIs 通过API 修改routing

### Command routing

### Event routing

### Physical routing

# Centralized file-based routing

通过文件管理routing

# Scheduling

配置需要重复执行的任务

受[**delayed delivery**](https://docs.particular.net/nservicebus/messaging/delayed-delivery)影响

## When not to use it

任务有if/switch等内部逻辑，用[saga timeouts](https://docs.particular.net/nservicebus/sagas/timeouts)

某一状态发生后要做某些动作，发一个event，不用Scheduler

# Gateway

# NServiceBus Router

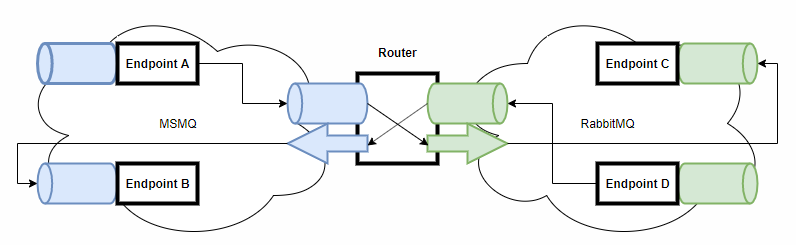
## Connecting to the router

## Router configuration

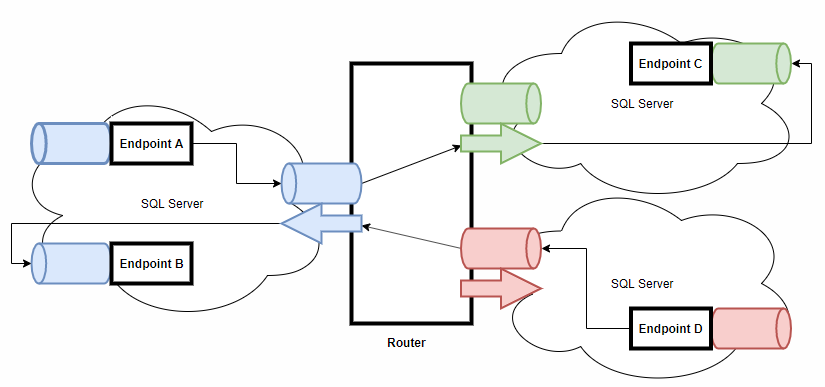
life cycle

## Error handling

# NServiceBus Router two-way bridge topology



# NServiceBus Router multi-way bridge topology



# NServiceBus Router backplane topology

# Transport Bridge

NServiceBus.Bridge is has been deprecated and replaced by NServiceBus.Router.

# Raw messaging using NServiceBus

# It is flexible in terms of message manipulation, therefore it is a good fit for integrations with 3rd party systems, building gateways and bridges.

## Configuration

## Sending

## Receiving

# Audit Filter 是否发送消息到审计队列

### Decorate messages with attributes

### Add to EndpointConfiguration

### Delegate filter fallback

# NServiceBus Mailer

## Enabling

## Usage

# Outbox

# FileShare Attachments store attachments for messages

# SQL Attachments

# Uses a SQL Server [varbinary](https://docs.microsoft.com/en-us/sql/t-sql/data-types/binary-and-varbinary-transact-sql) to store attachments for messages

# DataBus 发送大数据

# FluentValidation message validation

# DataAnnotations message validation

# 