

RabbitMQ for .Net Developers – Part 2

Dealing with Errors

Michael Stephenson
@michael_stephen
Michael_stephensonuk@yahoo.co.uk



pluralsight
hardcore developer training

Agenda

- **Errors & Retries**
- **Dead Letter Queue**
- **Routing Failures**

Errors & Retries - Basic

Background

- **Error Scenarios**

- Exception processing a message
- Poison Message
- A message we don't understand

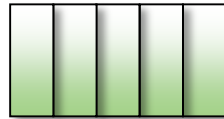
- **Basic Error Handling**

- Reject and Retry
- Reject and Discard

Retry Sample



Reject Sample



How Can We Do It

//To Reject Message

```
model.BasicReject(deliveryArgs.DeliveryTag, false);
```

//To Retry Message

```
model.BasicReject(deliveryArgs.DeliveryTag, true);
```

Demo Scenarios

Scenario	What happens
1	This message will reject and be discarded
2	This message will reject and retry

Demo

Errors & Retries

Errors & Retries - Advanced

Background

- We can't retry forever!
- We should retry n times and then reject
- We can decide if to reject sooner or later depending on the type of error

Retry Sample



Reject Sample



Scenarios

Scenario	What happens
1	This message will work every time
2	If retry count is < 3 = Send for Retry If retry count ≥ 3 reject message
3	If retry count is < 3 = Send for Retry If retry count $= 3$ accept & process message
4+	Reject the message

How Can We Do It

- **Reject = BasicReject (no re-queue)**
- **Retry = Custom Solution**
 - Custom Property to Track no. Retries
 - No retries incremented with each attempt
 - Instead of rejecting we re-publish a new message with updated properties
 - Application decides how many times it will allow a message to retry

Demo

Errors & Retries - Advanced

Dead Letter Queue

Background

- When we reject messages we don't just want to throw them away
- Store errors for review by Ops
- Advanced error handling?
- Repair & Resubmit

Dead Letter Sample



Repair & Resubmit
Message



Dead Letter Queue

How Can We Do It

- Declare an Exchange for receiving messages
- On queue set the x-dead-letter-exchange property

```
$args = @{"x-dead-letter-exchange"="Module3.Sample8.DeadLetterExchange"};
```

```
$model.QueueDeclare("Module3.Sample8.Normal", $true, $false, $false, $args)
```

- **Routing Rules**
 - Normal exchange routing rules can apply
 - You can setup a dead letter routing key

Demo

Dead Letter Queue

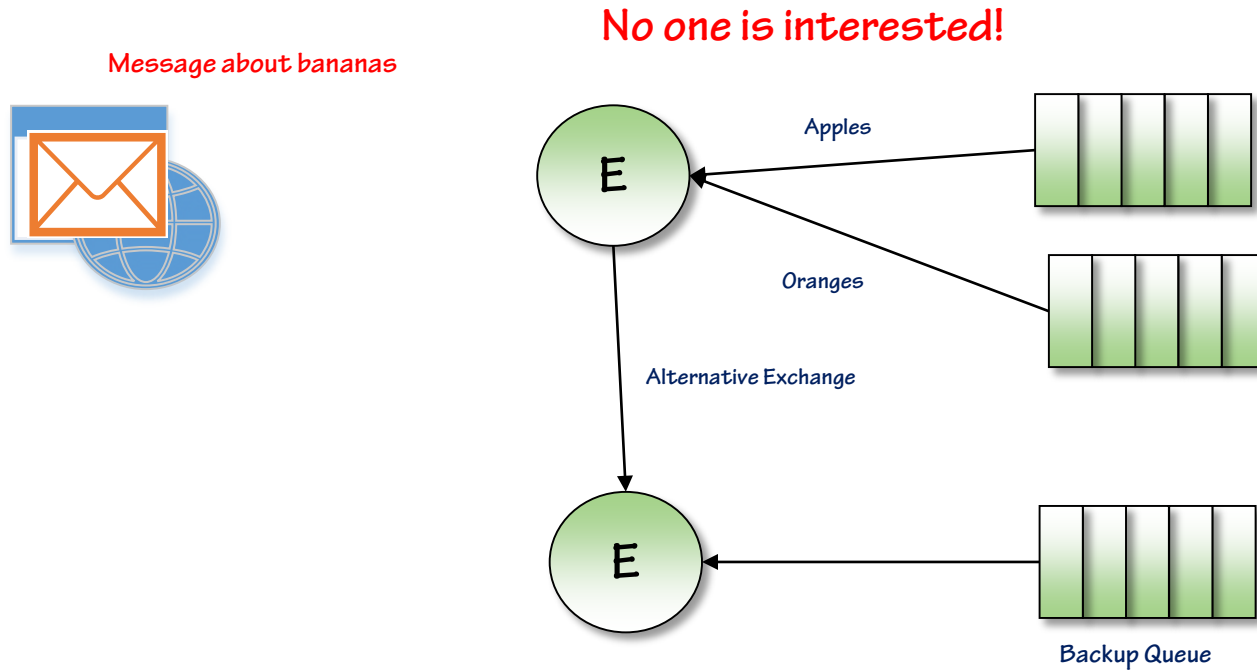
Routing Failures

Background

- When processing messages sometimes a message will not be subscribed to
- Subscription failure
 - Could be an error
 - Could be a malicious message
 - Could be fixable

Alternative Exchange!!

Routing Failures



How Can We Do It

```
Write-Host "Creating Exchange" -foregroundcolor green

$exchangeType = [RabbitMQ.Client.ExchangeType]::Topic

$args = @{"alternate-exchange"="Module3.Sample10.FailuresExchange"};

$model.ExchangeDeclare("Module3.Sample10.Exchange", $exchangeType,
    $true, $false, $args)
```

Demo

Routing Failures

Summary

- **Challenges of real world messaging solutions**
 - Basic Error Handling
 - Advanced Error Handling
 - Dead Letter Queue
 - Routing Failures