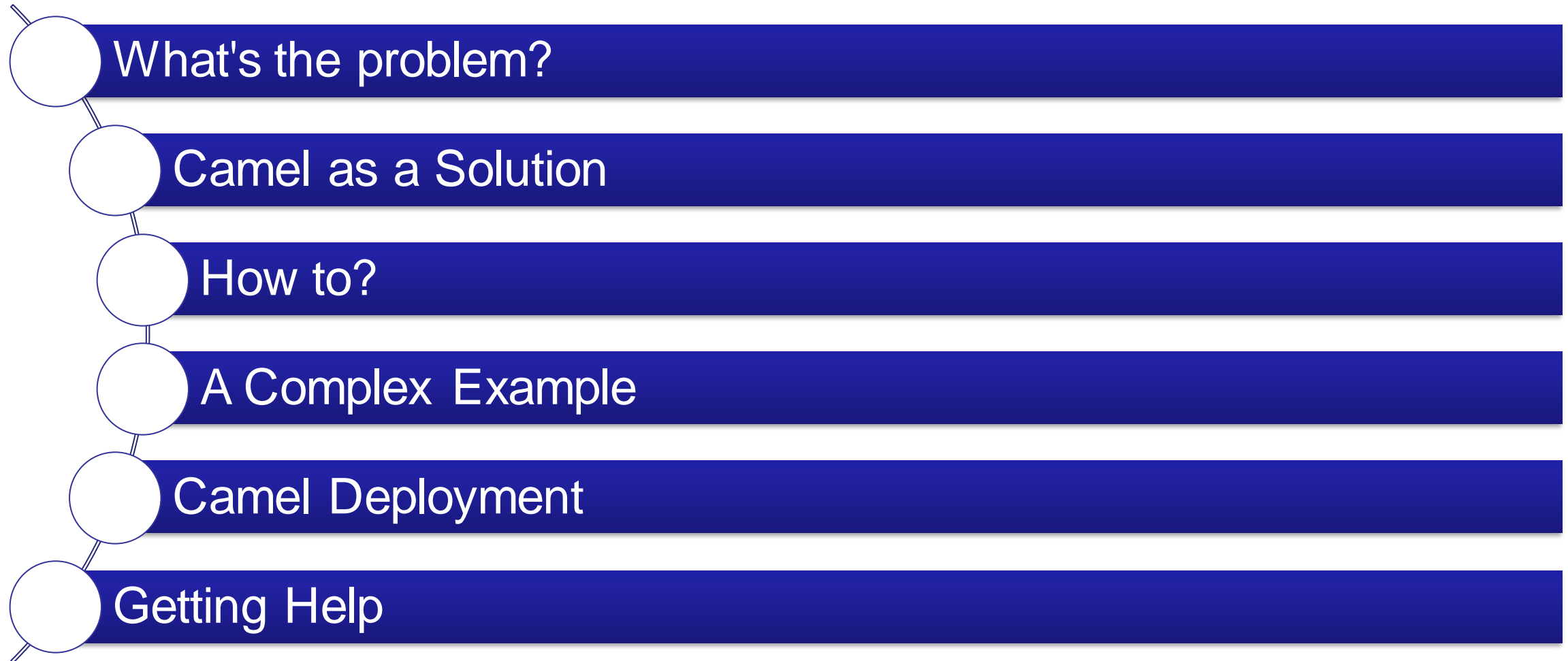


Integrate IBM i with Anything using Apache Camel

Jesse Gorzinski
Open Source Madman
jgorzins@us.ibm.com
Twitter: @IBMJesseG



Agenda



The Problem

Quick Exercise...

I'm going to show you **three slides**

They all have something in **common...**

See if you can **figure out what it is!**



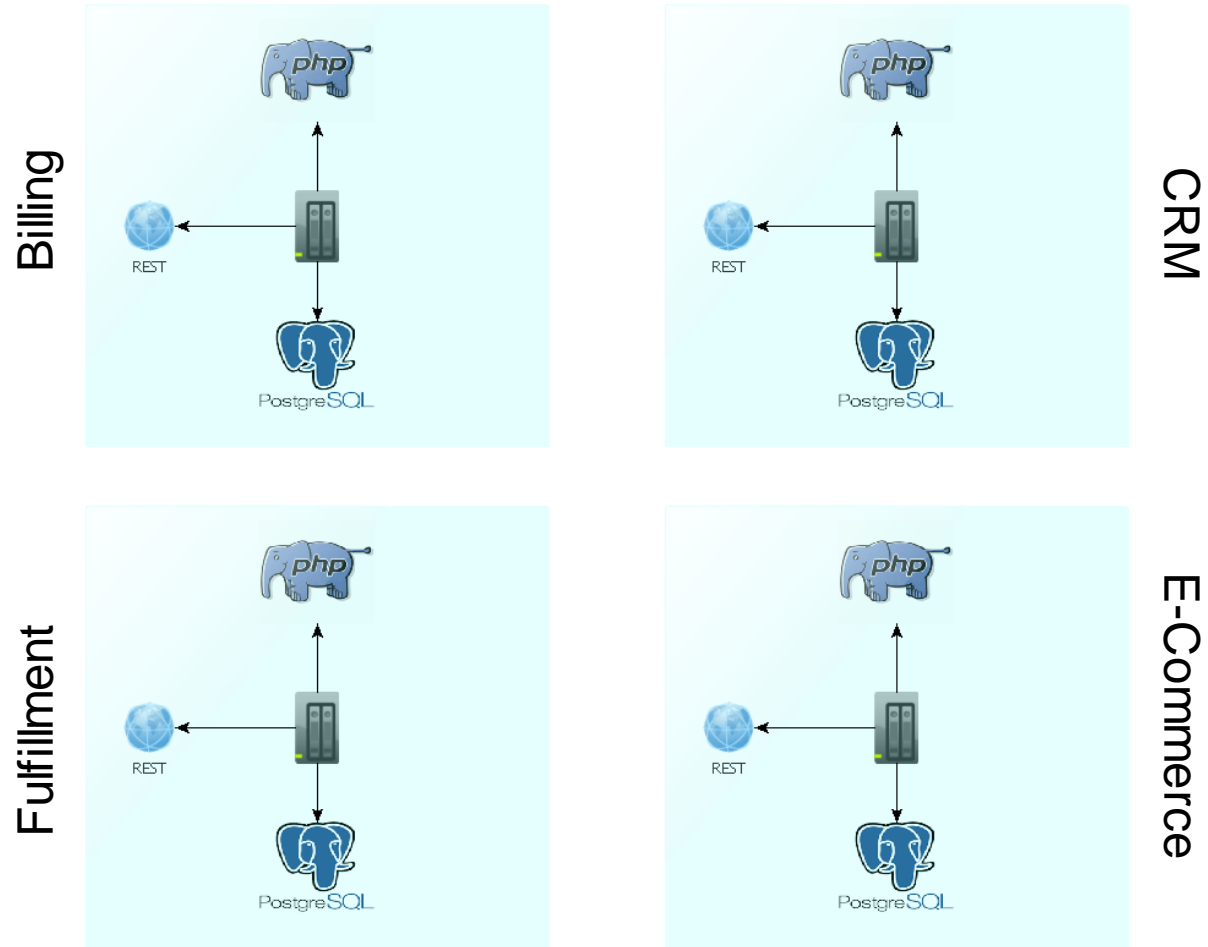
The Chupacabra



A Magical Unicorn...



A Fully Homogeneous Enterprise Landscape...



✓ **None of these things exist in real life!**

ESB Capabilities

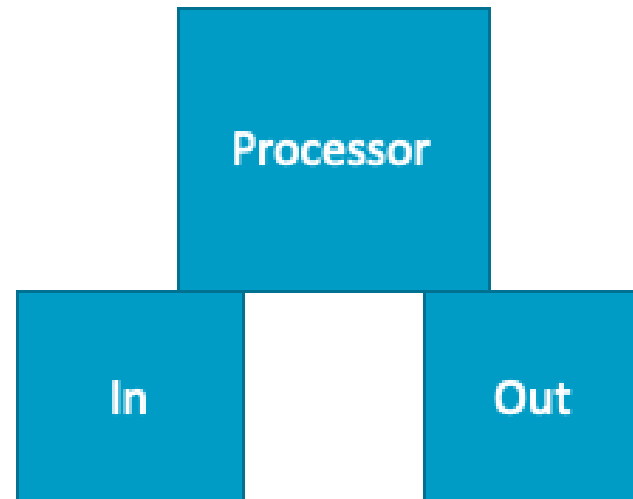
- Perhaps the most recognizable solution for this problem of heterogeneous systems integration is the **pattern of an Enterprise Service Bus** or ESB
- In general, an **ESB should provide** the following functionality:
 - **Transport Invocation** – *protocols and data binding*
 - **Data routing and transformation** – *Message routing patterns*
 - **Platform mediation** – *Language-specific adapters and mapping*
 - **Messaging** – *Message oriented middleware patterns*
 - **Orchestration** – *Business process coordination*
 - **QoS** – *Security, guaranteed delivery, transactions*
 - **Administration** – *Monitoring, operational administration*
 - **Platform agnosticism** – *Loose coupling, support for disparity*
 - **Data validation** – *Schema and/or canonical data validation*



Camel as a Solution

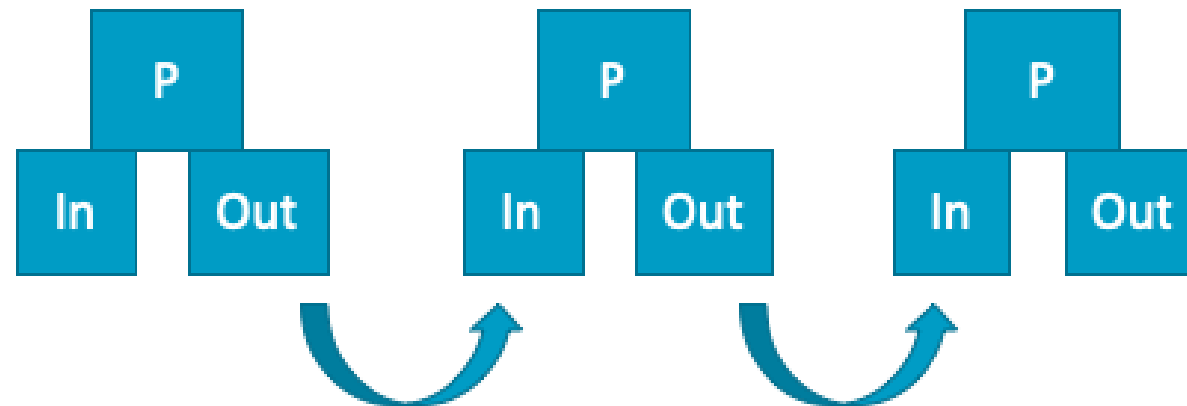
How Does It Work?

- Camel uses a repeatable, normalized concatenation of "Processor" and "Message" objects in a group called an Exchange
- There is an "in" message, a "Processor", and an "Out" message



How Does It Work?

- Exchanges can be chained together – like piping commands through *NIX – to form a Camel Route
- The "Out" message of a previous Exchange becomes the "in" message of a new Exchange
- This defines the route



- camel-ahc-ws
- camel-ahc
- camel-amqp
- camel-apns
- camel-asterisk
- camel-atmos
- camel-atmosphere-websocket
- camel-atom
- camel-avro
- camel-aws
- camel-azure
- camel-bam
- camel-barcode
- camel-beanio
- camel-beanstalk
- camel-bindy
- camel-blueprint
- camel-bonita
- camel-boon
- camel-box
- camel-braintree
- camel-cache
- camel-cassandraql
- camel-castor
- camel-cdi
- camel-chronicle
- camel-chunk
- camel-cm-sms
- camel-cmis
- camel-coap
- camel-cometd
- camel-consul
- camel-context
- camel-core-osgi
- camel-core-xml
- camel-couchbase
- camel-couchdb
- camel-crypto
- camel-csv
- camel-cxf-transport
- camel-cxf
- camel-dns
- camel-docker
- camel-dozer
- camel-drill
- camel-dropbox
- camel-eclipse
- camel-ehcache
- camel-ejb
- camel-elasticsearch
- camel-elasticsearch5
- camel-elsql
- camel-etcd
- camel-eventadmin
- camel-exec
- camel-flatpack
- camel-flink
- camel-fop
- camel-freemarker
- camel-ftp
- camel-ganglia
- camel-geocoder
- camel-git
- camel-github
- camel-google-calendar
- camel-google-drive
- camel-google-mail
- camel-google-pubsub
- camel-gora
- camel-grape
- camel-grpc
- camel-gson
- camel-guava-eventbus
- camel-guice
- camel-hawtdb
- camel-hazelcast
- camel-hbase
- camel-hdfs
- camel-hdfs2
- camel-hessian
- camel-hipchat
- camel-hl7
- camel-http-common
- camel-http
- camel-hytrix
- camel-ibatis
- camel-ical
- camel-ignite
- camel-infinispan
- camel-influxdb
- camel-irc
- camel-ironmq
- camel-jackson
- camel-jacksonxml
- camel-jasypt
- camel-javaspaces
- camel-jaxb
- camel-jbpm
- camel-jcache
- camel-jdbc
- camel-jetty-common
- camel-jetty
- camel-jetty9
- camel-jgroups
- camel-jibx
- camel-jing
- camel-jira
- camel-jms
- camel-jmx
- camel-johnzon
- camel-jolt
- camel-josql
- camel-jpa
- camel-jsonpath
- camel-jt400
- camel-juel
- camel-jxpath
- camel-kafka
- camel-kestrel
- camel-krati
- camel-kubernetes
- camel-kura
- camel-ldap
- camel-leveldb
- camel-linkedin
- camel-lucene
- camel-lumberjack
- camel-lzf
- camel-milo
- camel-mina
- camel-mina2
- camel-mllp
- camel-mongodb-gridfs
- camel-mongodb
- camel-mongodb3
- camel-mqtt
- camel-msv
- camel-mustache
- camel-mvel
- camel-mybatis
- camel-nagios
- camel-nats

There are a lot of components!!

Synergy with the Operating System via Apache Camel

To send or receive data from a data queue

```
jt400://user:password@system/QSYS.LIB/LIBRARY.LIB/QUEUE.DTAQ[?options]
```

To send or receive messages from a message queue

A blue starburst badge with the word "New!" in white text.

```
jt400://user:password@system/QSYS.LIB/LIBRARY.LIB/QUEUE.MSGQ[?options]
```

To call remote program

```
jt400://user:password@system/QSYS.LIB/LIBRARY.LIB/program.PGM[?options]
```

You can append query options to the URI in the following format, `?option=value&option=value&...`

Db2 Enhancements for Apache Camel

- JSON Publishing Functions provide data in a manner understood by Kafka/ActiveMQ consumers

```
SELECT JSON_OBJECT(  
  KEY 'Department' VALUE  
  JSON_ARRAYAGG(JSON_OBJECT(  
    KEY 'Id' VALUE X.DEPTNO,  
    KEY 'Name' VALUE X.DEPTNAME)))  
  AS DEPT_JSON  
FROM TOYSTORE.DEPT X;
```

- Data Queue Functions allow integration with queues (and therefore Apache Camel) directly from the database

```
call qsys2.send_data_queue_utf8(  
  message_data      => scottf.dq_json,  
  data_queue        => 'HANDOFF_DQ',  
  data_queue_library => 'BANKONOSS');
```

Camel Source

- Camel routes can be coded in Java DSL

```
from("activemq:foo").filter().
  xquery("//foo").
  to("activemq:bar")
```

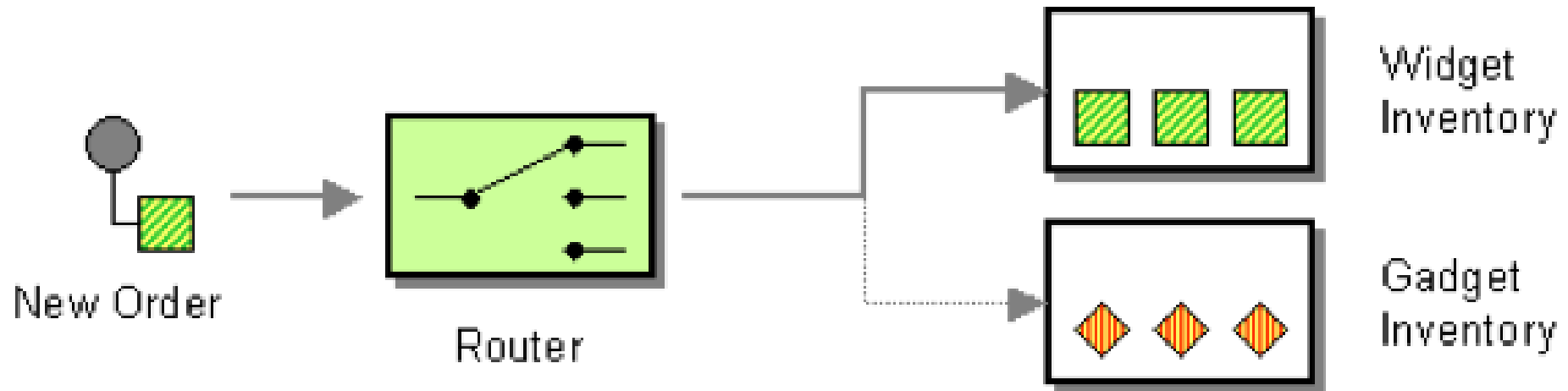
- Or by using Camel Spring DSL XML files

```
<route xmlns="http://camel.apache.org/schema/spring"
id="TimerClient">
  <from uri="activemq:topic:broadcastData"/>
  <setBody>
    <constant>This is a test JMS message.</constant>
  </setBody>
  <to uri="activemq:Test_ActiveMQ_Route.inQueue"/>
</route>
```



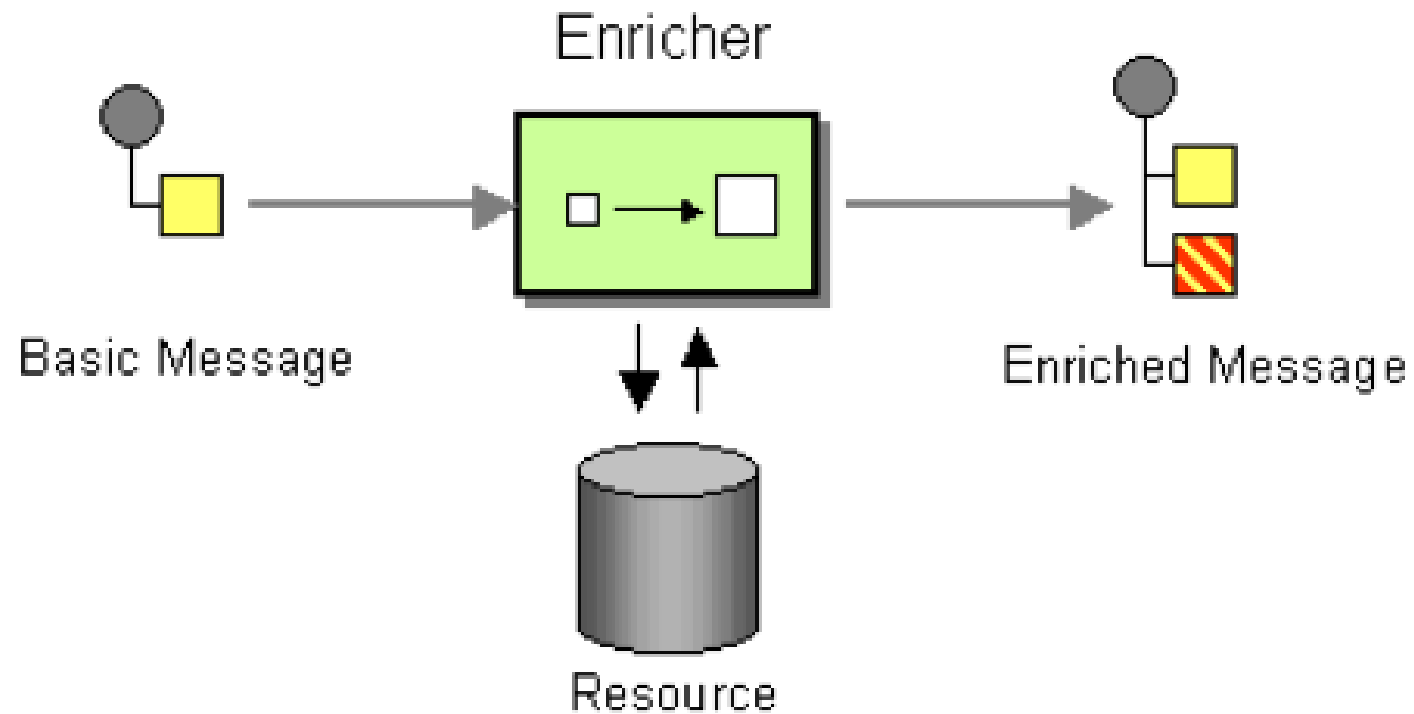
Integration Patterns – Content Based Router

- Message routing based on message criteria



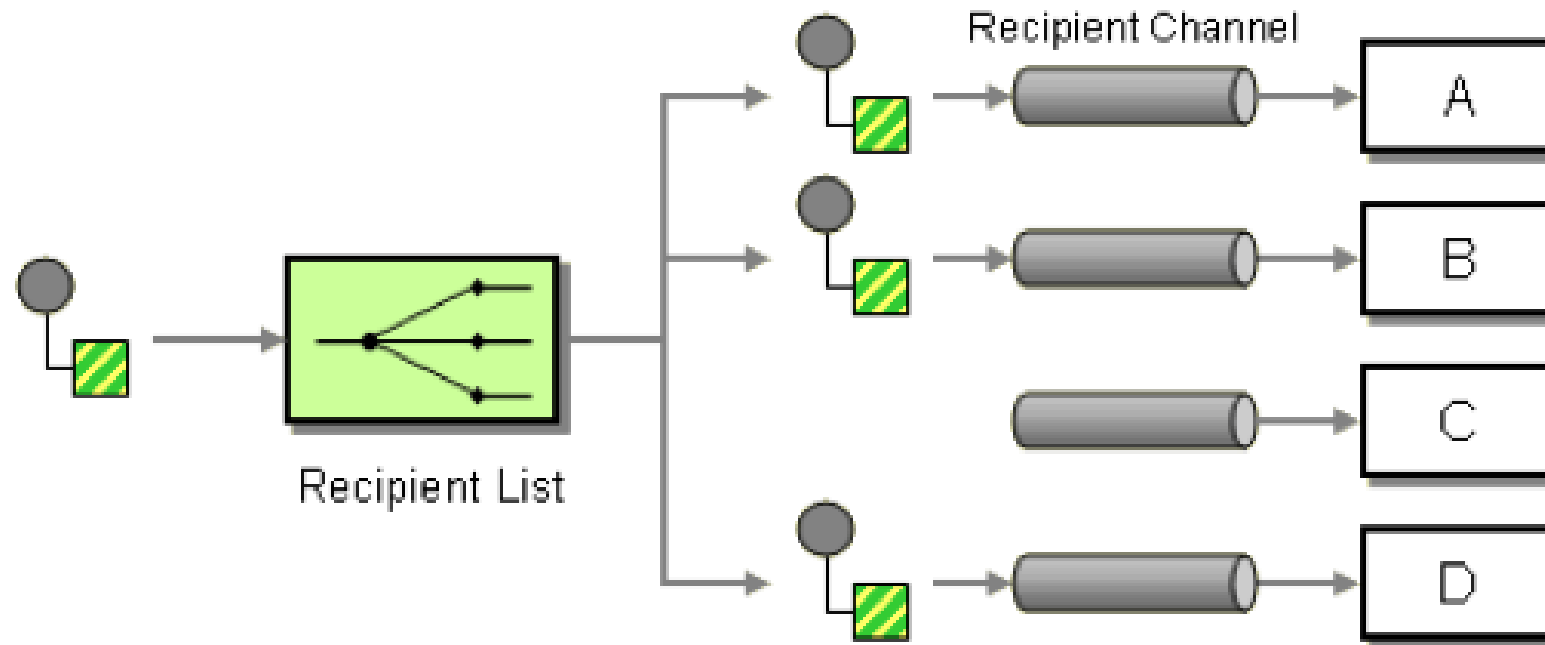
Integration Patterns – Content Enricher

- Pattern adds data to message from another source



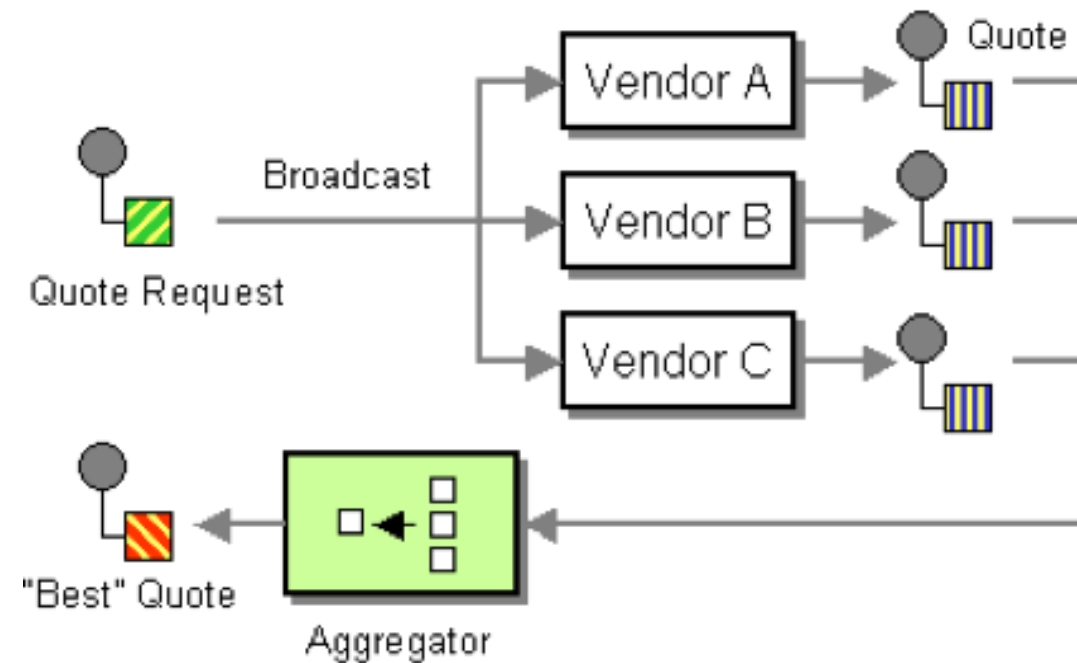
Integration Patterns – Recipient List

- Message broadcast to n channels based on message criteria
- Broadcast to all channels == Multicast Pattern



Integration Patterns – Scatter-Gather

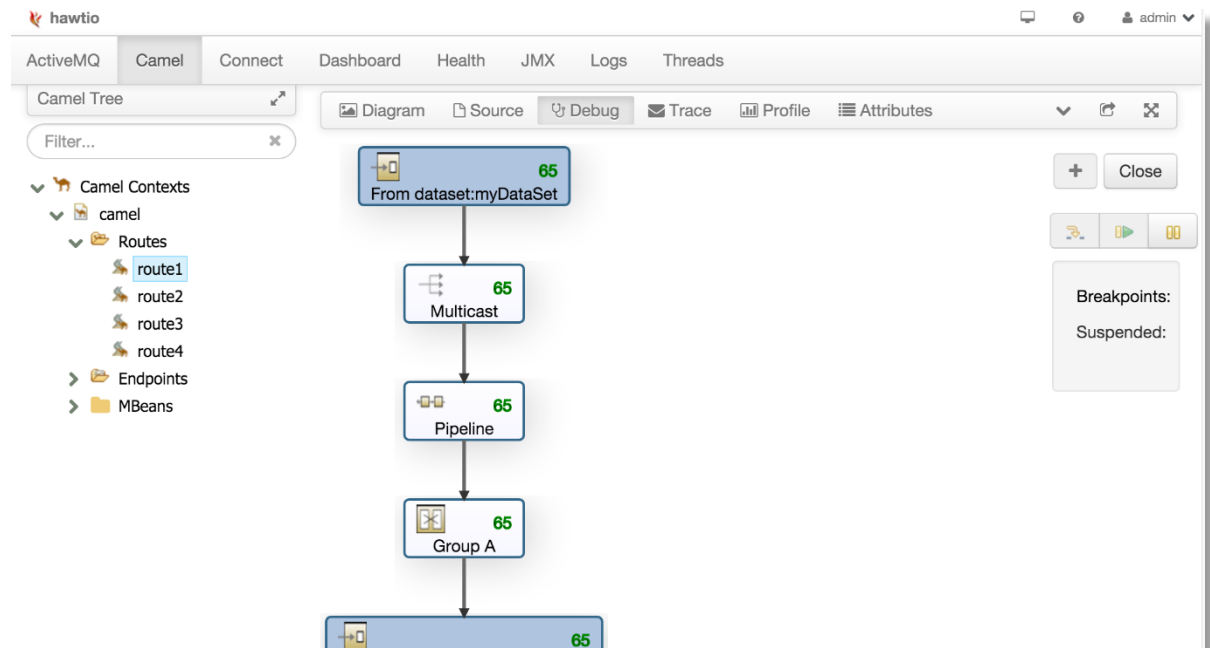
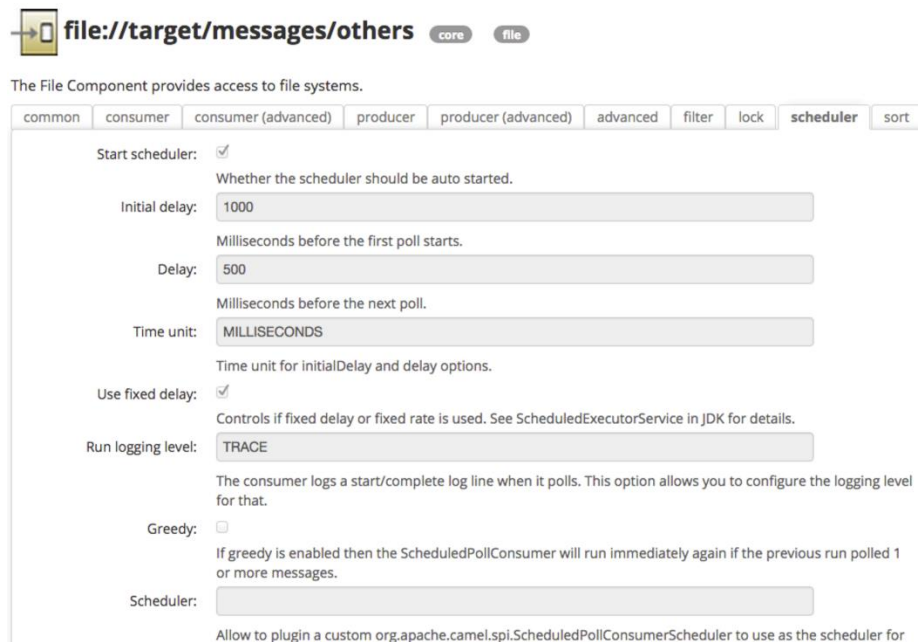
- Messages broadcast to disparate endpoints and results aggregated into a single message



How to? (Java DSL emphasis)

Camel Development

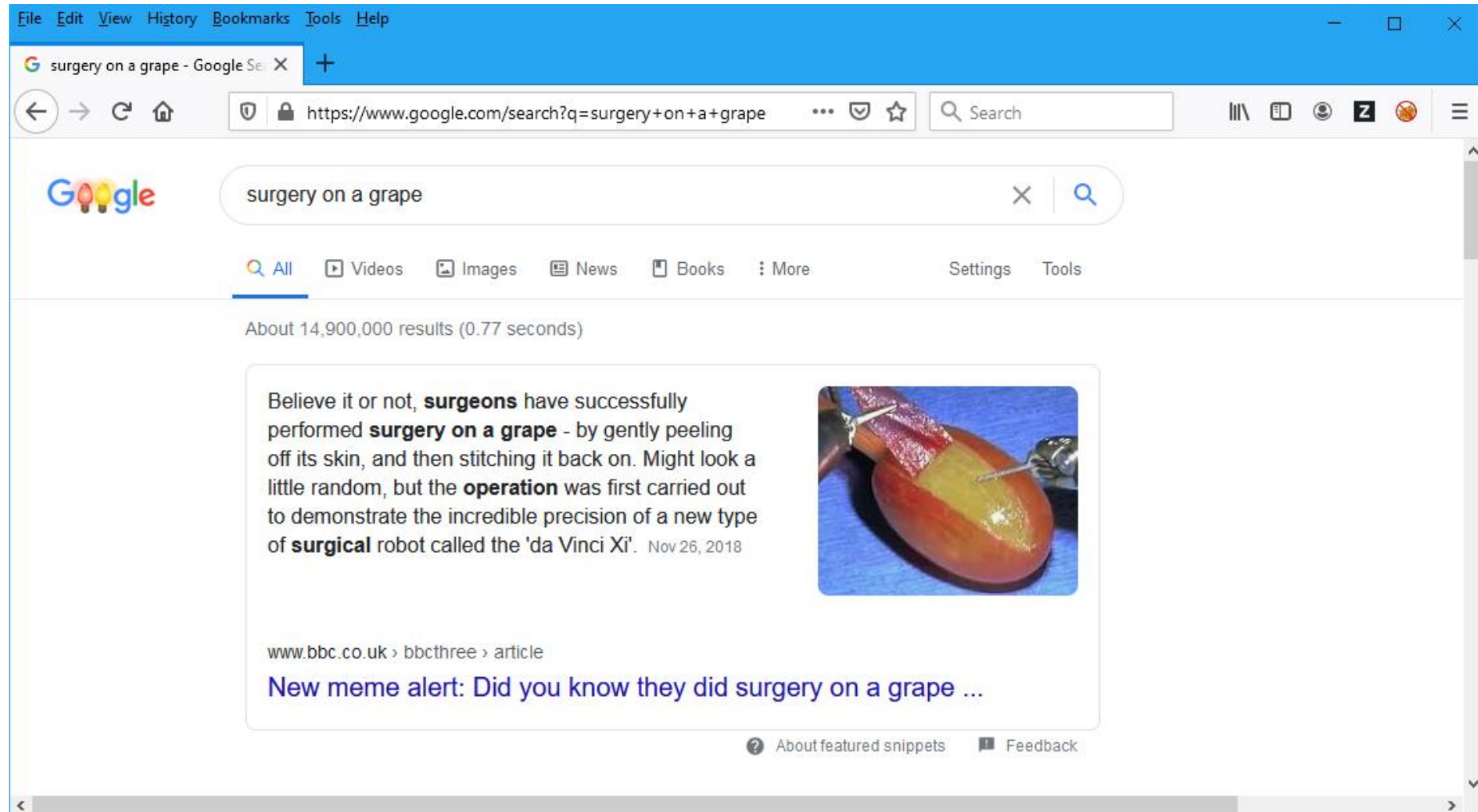
- **Major IDEs** like Eclipse and IntelliJ can **ease Camel development** by allowing routes to execute within the IDE
- As of Camel 2.16, **comprehensive in-line tools are available** which allow auto-completion and even list every parameter of an available component
- Hawtio allows for **visual debugging of routes**, tracing of message lifecycle, diagram driven development, as well as monitoring



Creating/Configuring Endpoints

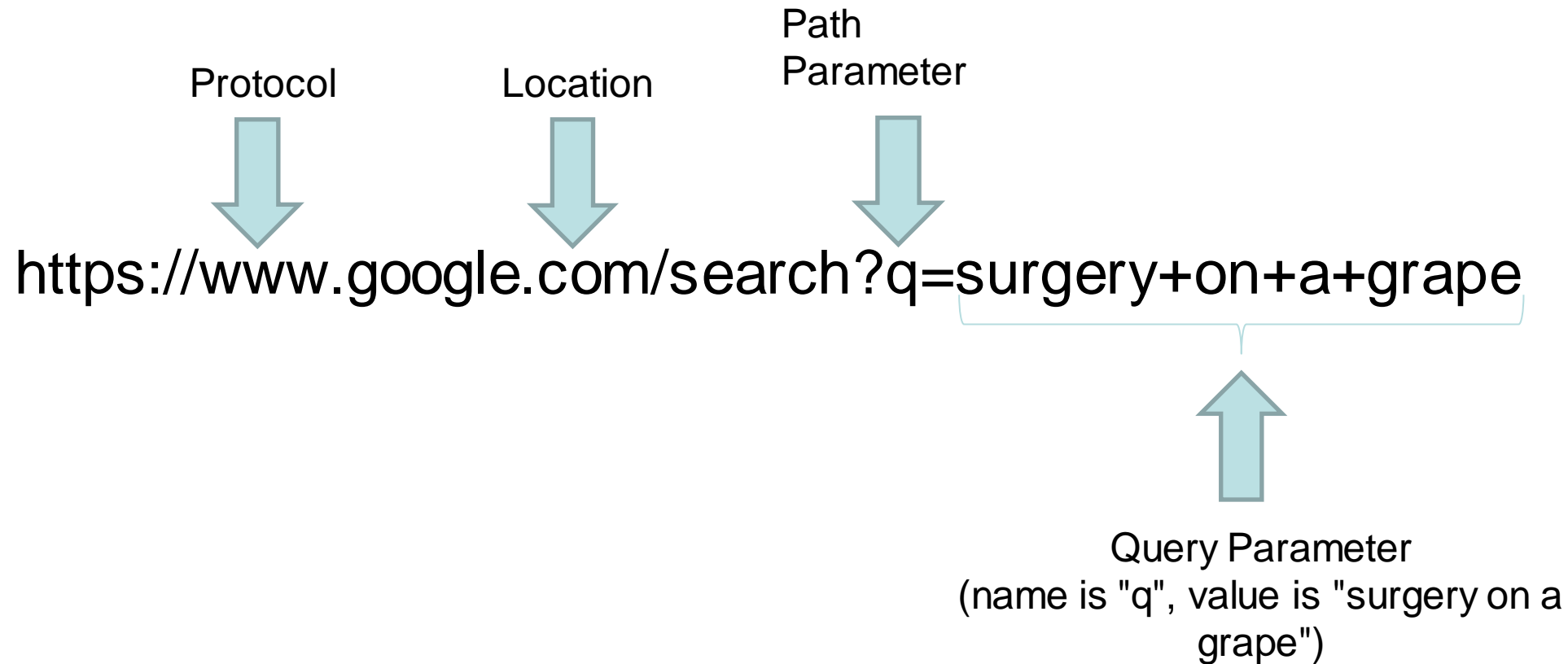
- Camel documentation is at : <https://camel.apache.org/manual/latest/faq/how-do-i-configure-endpoints.html>
- Several techniques
 - Explicit Java code (uses URIs)
 - Spring XML (uses URIs)
 - Java DSL (uses URIs)

URI Example (URL is a type of URI)



The screenshot shows a web browser window with a Google search for "surgery on a grape". The search results page displays a featured snippet with the following text: "Believe it or not, **surgeons** have successfully performed **surgery on a grape** - by gently peeling off its skin, and then stitching it back on. Might look a little random, but the **operation** was first carried out to demonstrate the incredible precision of a new type of **surgical** robot called the 'da Vinci Xi'. Nov 26, 2018". To the right of the text is a photograph of a grape being operated on with surgical instruments. Below the snippet, the source is cited as "www.bbc.co.uk > bbcthree > article" and a link is provided: "New meme alert: Did you know they did surgery on a grape ...". The browser's address bar shows the URL "https://www.google.com/search?q=surgery+on+a+grape".

URI Example



Apache Camel Component Documentation

- <https://camel.apache.org/components/latest/jt400-component.html>

URI FORMAT

To send or receive data from a data queue

```
jt400://user:password@system/QSYS.LIB/LIBRARY.LIB/QUEUE.DTAQ[?options]
```

To send or receive messages from a message queue

```
jt400://user:password@system/QSYS.LIB/LIBRARY.LIB/QUEUE.DTAQ[?options]
```

To call remote program

```
jt400://user:password@system/QSYS.LIB/LIBRARY.LIB/QUEUE.DTAQ[?options]
```

PATH PARAMETERS (5 PARAMETERS):

Name	Description
userID	Required Returns the ID of the IBM i user.
password	Required Returns the password of the IBM i user.
systemName	Required Returns the name of the IBM i system.
objectPath	Required Returns the fully qualified integrated file system path.
type	Required Whether to work with data queues or remote message queues (MSGQ).

QUERY PARAMETERS (33 PARAMETERS):

Name	Description
ccsid (common)	Sets the CCSID to use for the connection with the IBM i system.
format (common)	Sets the data format for sending messages. There are 2 enums: text and binary.
guiAvailable (common)	Sets whether IBM i prompting is enabled in the environment running the component.
keyed (common)	Whether to use keyed or non-keyed data queues.
searchKey (common)	Search key for keyed data queues.

Simplest Camel Route

- Specify a simple route with a source URI and a target URI
- Camel will implicitly create the endpoint objects and tie them together
- Camel doc on Routes is very useful for starters: <https://camel.apache.org/manual/latest/routes.html>

```
final String sourceUri = // some URI of an endpoint that produces data
final String targetUri = // some URI of an endpoint to receive data
context.addRoutes(new RouteBuilder() {
    @Override
    public void configure() {
        from(sourceUri)
        .to(targetUri);
    }
});
```

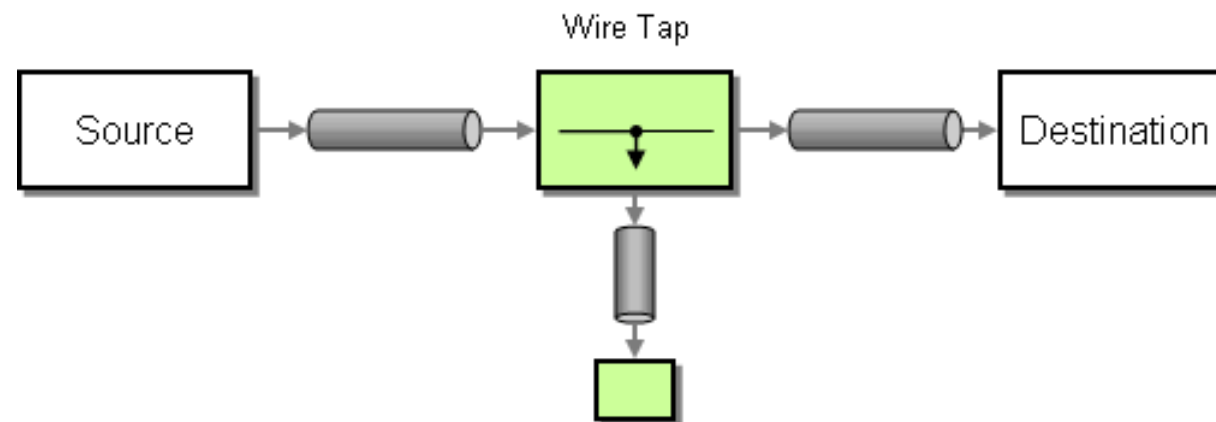
Message Queue to Email Bridge

- This sample code uses a helper object to generate the URI (parsed from a config file)
- Introduced the "Wire Tap" EIP

```
final String msgqUri = conf.getMsgQUri(); //something like -> jt400://username:password@localhost/qsys.lib/mylib.lib/myq.MSGQ?format=binary&guiAvailable=false
final String smtpUri = conf.getSmtpUri(); //something like -> smtp://my.smtp.server.com?from=jgorzins@us.ibm.com&to=jgorzins@us.ibm.com&subject=Camel is Really Amazing!
context.addRoutes(new RouteBuilder() {
    @Override
    public void configure() {
        from(msgqUri)
        .wireTap("log:msgq_to_email?showAll=true&level=INFO") // This is just for debug data flowing through the route
        .to(smtpUri);
    }
});
```

Wire Tap EIP

- <https://camel.apache.org/components/latest/eips/wireTap-eip.html>
- Used here because:
 - Demo/starter program, visual feedback is nice :)



Data Queue to Kafka Bridge

- Same concept, different URIs
- Applied case: stream Db2 transactions to Apache Kafka

```
final String dtaqUri = conf.getDtaqUri(); //something like -> jt400://username:password@localhost/qsys.lib/mylib.lib/myq.DTAQ?keyed=false&format=binary&guiAvailable=false
final String kafkaUri = conf.getKafkaUri(); //something like -> kafka:mytopic?brokers=mybroker:9092
context.addRoutes(new RouteBuilder() {
    @Override
    public void configure() {
        from(dtaqUri)
        .wireTap("log:msgq_to_email?showAll=true&level=INFO") // This is just for debugging data flowing through the route
        .to(kafkaUri);
    }
});
```

Consume IoT Data? No Problem!

```

A - COMMON1.IINTHECLOUD.COM
File Edit View Communication Actions Window Help
A - COMMON1.IINTHECLOUD.C... B - IDEVPHP.IDEV.CLOUD.COM
Display Messages
Queue . . . . . : DRIVEWAY          System: COMMON1
Library . . . . : QUSRSYS           Program . . . . : *DSPMSG
Severity . . . . : 00              Delivery . . . . : *NOTIFY

Type reply (if required), press Enter.
- 0, 0 - light: 84
  From . . . . : DRIVEWAY          03/02/21  15:18:36
  0, 0 - temp: 25.25
  From . . . . : DRIVEWAY          03/02/21  15:18:36
  0, 0 - light: 84
  From . . . . : DRIVEWAY          03/02/21  15:19:07
  0, 0 - temp: 25.25
  From . . . . : DRIVEWAY          03/02/21  15:19:07
  0, 0 - light: 84
  From . . . . : DRIVEWAY          03/02/21  15:19:37
  0, 0 - temp: 25.31
  From . . . . : DRIVEWAY          03/02/21  15:19:37
  0, 0 - light: 82

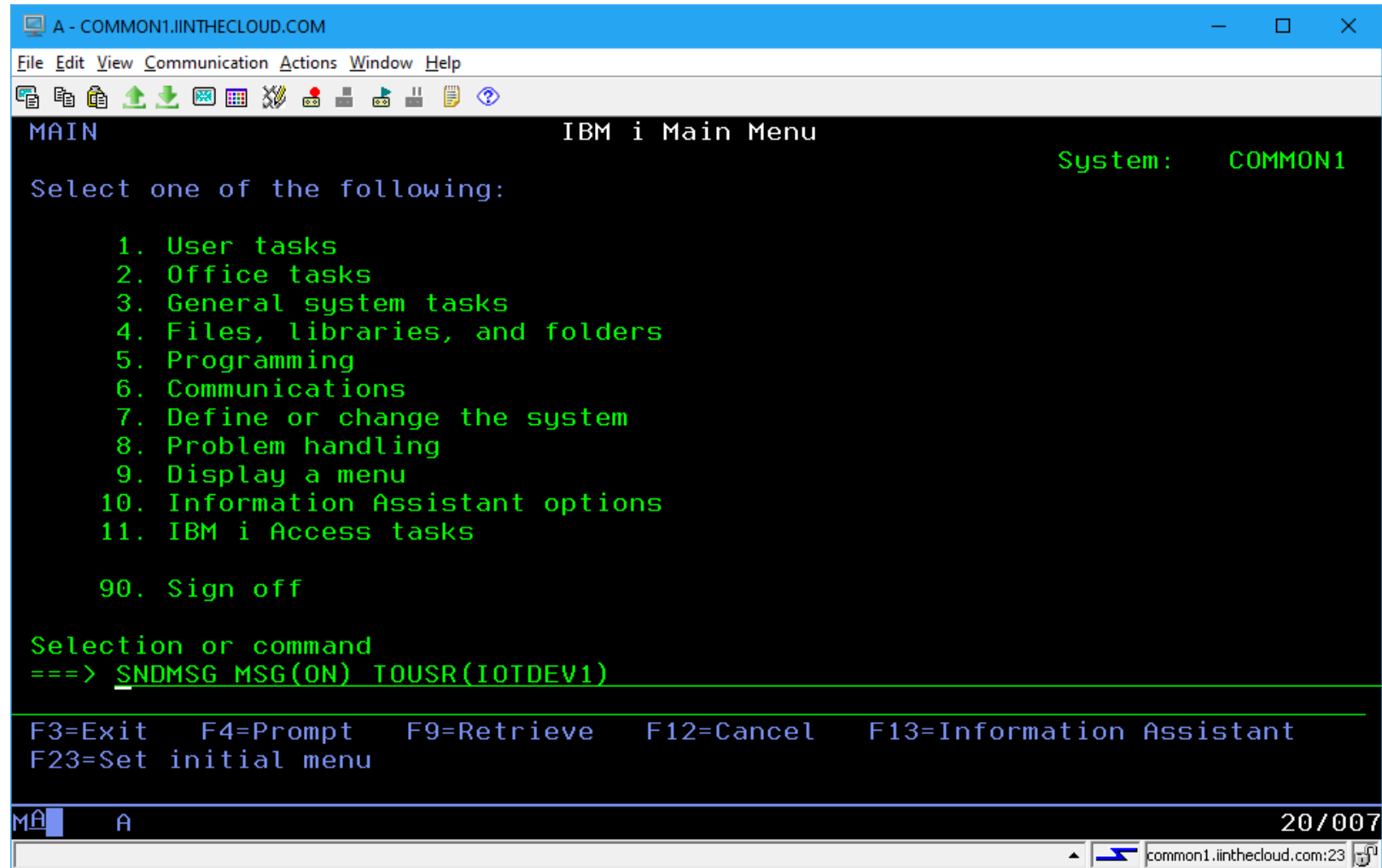
Bottom
F3=Exit          F11=Remove a message    F12=Cancel
F13=Remove all   F16=Remove all except  F24=More keys

MA A
08/001
common1.iinthecloud.com:23
  
```

Consume IoT Data? No Problem!

```
context.addRoutes(new RouteBuilder() {
    @Override
    public void configure() {
        from("paho:factory/1/light?brokerUrl=ssl://localhost")
            .to("jt400://driveway:xxxxxx@localhost/qsys.lib/QUSRSYS.lib/DRIVEWAY.MSGQ?guiAvailable=false");
    }
});
context.addRoutes(new RouteBuilder() {
    @Override
    public void configure() {
        from("paho:factory/1/temp?brokerUrl=ssl://localhost")
            .to("jt400://driveway:xxxxxx@localhost/qsys.lib/QUSRSYS.lib/DRIVEWAY.MSGQ?&guiAvailable=false");
    }
});
```

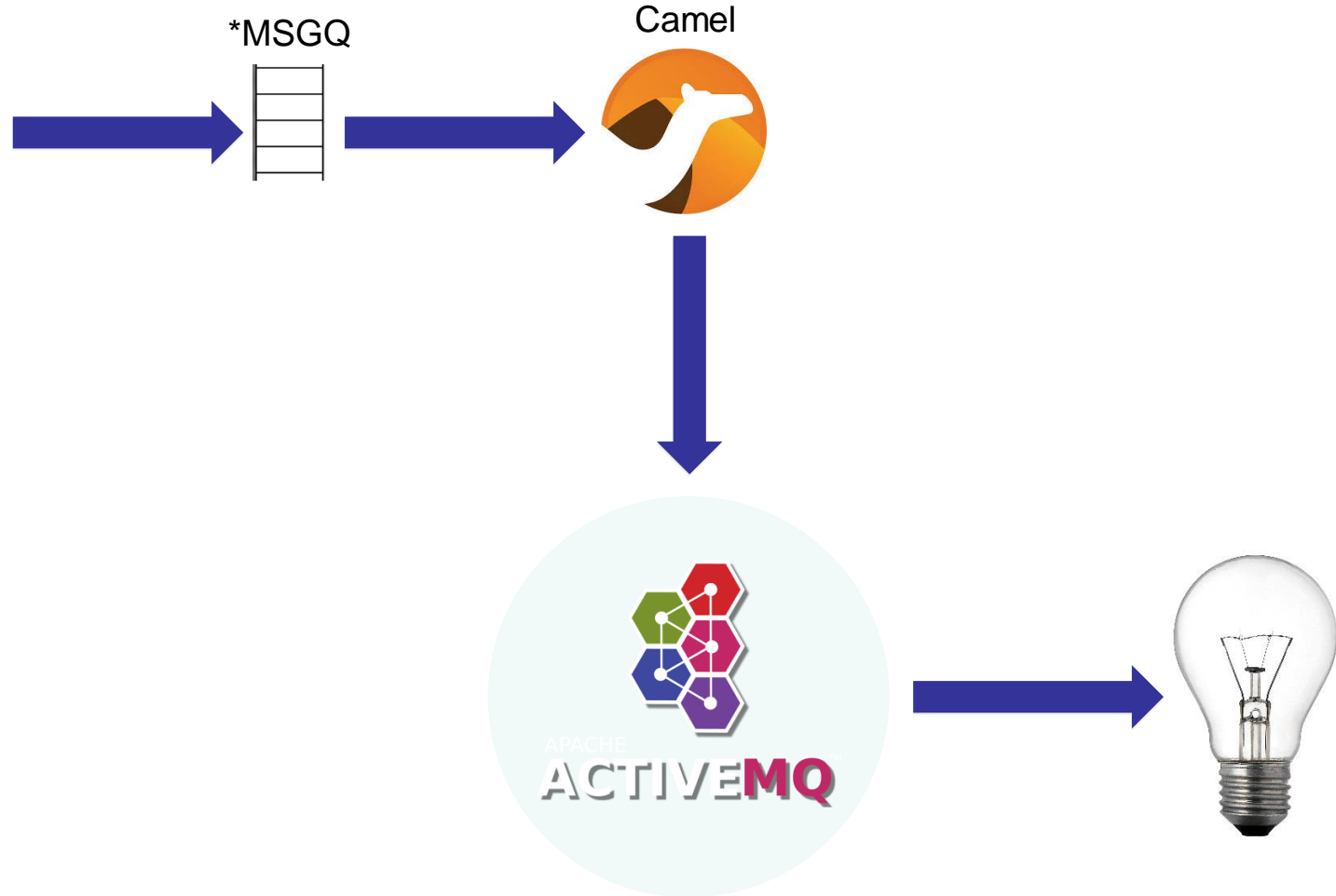
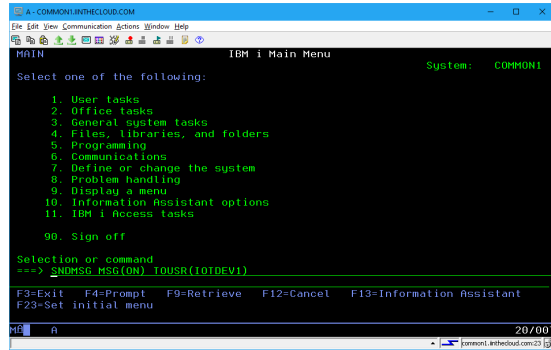

Control IoT Devices? No Problem!



```
A - COMMON1.IINTHECLOUD.COM
File Edit View Communication Actions Window Help
MAIN                                IBM i Main Menu                                System:  COMMON1
Select one of the following:
1. User tasks
2. Office tasks
3. General system tasks
4. Files, libraries, and folders
5. Programming
6. Communications
7. Define or change the system
8. Problem handling
9. Display a menu
10. Information Assistant options
11. IBM i Access tasks
90. Sign off
Selection or command
==> SNDMSG MSG(ON) TOUSR(IOTDEV1)
F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel  F13=Information Assistant
F23=Set initial menu
Mâ A 20/007
common1.iinthecloud.com:23
```

Control IoT Devices? No Problem!

IBM i SNDMSG Command

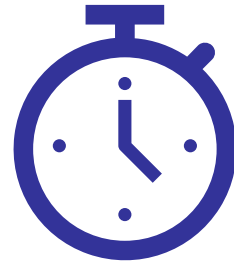


A Complex Example

A More Complex Example...

- This example is a simple disk usage monitor
 - Continually checks average disk usage
 - If disk usage is >90%, send an email
 - Send an email every hour until disk usage is freed back up
- Will use several techniques and EIPs

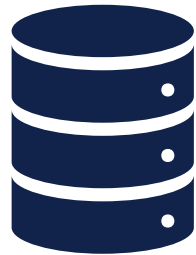
Timer Endpoint



Often used to kick off tasks on some form of periodic basis (in this case, polling disk usage)

<https://camel.apache.org/components/latest/timer-component.html>

JDBC Endpoint



**Used to query the database
(using QSYS2.SYSDISKSTAT)**

<https://camel.apache.org/components/latest/jdbc-component.html>



**Supports both queries (SELECT) and
operations (INSERT, UPDATE, etc.).**

SQL is sent in the message body

JDBC DataSource Config

URI FORMAT

```
jdbc:dataSourceName[?options]
```

OPTIONS

The JDBC component supports 3 options, which are listed below.

Name	Description	Default	Type
dataSource (producer)	To use the DataSource instance instead of looking up the data source by name from the registry.		DataSource

Using Java Objects for Endpoint Configuration

- <https://camel.apache.org/manual/latest/faq/how-do-i-configure-endpoints.html>
- One technique: Use the "#class:classname" syntax
 - Uses zeroable constructor of the given class
- Another technique: store the object into the Camel registry, then refer to it by name in the URI

```
context.getRegistry().bind("jt400", new AS400JDBCDataSource("localhost", "*CURRENT", ""));
```


- In this example, we're using IBM Java Toolbox classes. If using maven, define your dependency in pom.xml

```
<dependency>  
  <groupId>net.sf.jt400</groupId>  
  <artifactId>jt400</artifactId>  
  <version>10.5</version>  
</dependency>
```


Disk Monitor Example

```
context.getRegistry().bind("jt400", new AS400JDBCDataSource("localhost", "*CURRENT", ""));
context.addRoutes(new RouteBuilder() {
    @Override
    public void configure() {
        from("timer://foo?period=5000").routeId("diskmon")
            .setBody(constant("SELECT AVG(PERCENT_USED)FROM QSYS2.SYSDISKSTAT A"))
            .to("jdbc:jt400")
            .process((exchange) -> {
                Object diskUsageValue = ((ArrayList<HashMap<String, Object>>)exchange.getIn().getBody()).remove(0).values().iterator().next();
                exchange.getIn().setBody(diskUsageValue);
            })
            .choice()
                .when(body().isGreaterThan(90))
                    .process((exchange) -> {exchange.getIn().setBody("Average utilization of your disks is "+ exchange.getIn().getBody() + "%");})
                    .to("log:disk_space_mon?showAll=true&level=ERROR")
                    .to("smtp://mysmtpserver:1025?from=me@gmail.com&to=me@gmail.com&subject=Disk getting full!!")
                    .process((exchange) -> { pauseContext(exchange.getContext(), 1000*60*60); })
                .otherwise()
                    .wireTap("log:disk_space_mon?showAll=true&level=INFO");
    }
});
```

Disk Monitor Example

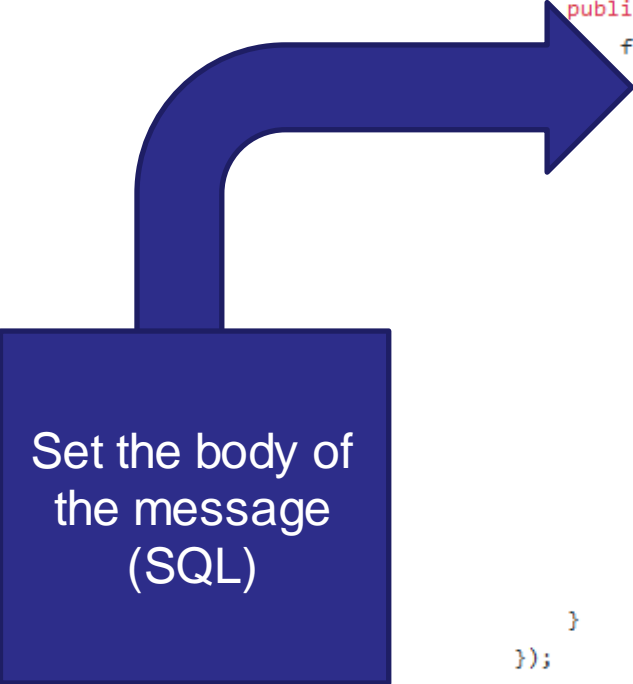


Instantiate
Timer to kick off
the route

```
context.addRoutes(new RouteBuilder() {
    @Override
    public void configure() {
        from("timer://foo?period=5000").routeId("diskmon")
            .setBody(constant("SELECT AVG(PERCENT_USED)FROM QSYS2.SYSDISKSTAT A"))
            .to("jdbc:jt400")
            .process((exchange) -> {
                Object diskUsageValue = ((ArrayList<HashMap<String, Object>>)exchange.getIn().getBody()).remove(0).values().iterator().next();
                exchange.getIn().setBody(diskUsageValue);
            })
            .choice()
                .when(body().isGreaterThan(90))
                    .process((exchange) -> {exchange.getIn().setBody("Average utilization of your disks is "+ exchange.getIn().getBody() + "%");})
                    .to("log:disk_space_mon?showAll=true&level=ERROR")
                    .to("smtp://mysmtpserver:1025?from=me@gmail.com&to=me@gmail.com&subject=Disk getting full!!")
                    .process((exchange) -> { pauseContext(exchange.getContext(), 1000*60*60); })
                .otherwise()
                    .wireTap("log:disk_space_mon?showAll=true&level=INFO");
    }
});
```

```
from("timer://foo?period=5000").routeId("diskmon")
```

Disk Monitor Example



Set the body of
the message
(SQL)

```
context.addRoutes(new RouteBuilder() {
    @Override
    public void configure() {
        from("timer://foo?period=5000").routeId("diskmon")
            .setBody(constant("SELECT AVG(PERCENT_USED)FROM QSYS2.SYSDISKSTAT A"))
            .to("jdbc:jt400")
            .process((exchange) -> {
                Object diskUsageValue = ((ArrayList<HashMap<String, Object>>)exchange.getIn().getBody()).remove(0).values().iterator().next();
                exchange.getIn().setBody(diskUsageValue);
            })
            .choice()
                .when(body().isGreaterThan(90))
                    .process((exchange) -> {exchange.getIn().setBody("Average utilization of your disks is "+ exchange.getIn().getBody() + "%");})
                    .to("log:disk_space_mon?showAll=true&level=ERROR")
                    .to("smtp://mysmtpserver:1025?from=me@gmail.com&to=me@gmail.com&subject=Disk getting full!!")
                    .process((exchange) -> { pauseContext(exchange.getContext(), 1000*60*60); })
                .otherwise()
                    .wireTap("log:disk_space_mon?showAll=true&level=INFO");
    }
});
```

```
.setBody(constant("SELECT AVG(PERCENT_USED)FROM QSYS2.SYSDISKSTAT A"))
```

Disk Monitor Example

Route the message to the JDBC component (note the "jt400" datasource)

```
context.addRoutes(new RouteBuilder() {
    @Override
    public void configure() {
        from("timer://foo?period=5000").routeId("diskmon")
            .setBody(constant("SELECT AVG(PERCENT_USED)FROM QSYS2.SYSDISKSTAT A"))
            .to("jdbc:jt400")
            .process((exchange) -> {
                Object diskUsageValue = ((ArrayList<HashMap<String, Object>>)exchange.getIn().getBody()).remove(0).values().iterator().next();
                exchange.getIn().setBody(diskUsageValue);
            })
            .choice()
                .when(body().isGreaterThan(90))
                    .process((exchange) -> {exchange.getIn().setBody("Average utilization of your disks is "+ exchange.getIn().getBody() + "%");})
                    .to("log:disk_space_mon?showAll=true&level=ERROR")
                    .to("smtp://mysmtpserver:1025?from=me@gmail.com&to=me@gmail.com&subject=Disk getting full!!")
                    .process((exchange) -> { pauseContext(exchange.getContext(), 1000*60*60); })
                .otherwise()
                    .wireTap("log:disk_space_mon?showAll=true&level=INFO");
    }
});
```

```
.to("jdbc:jt400")
```

Disk Monitor Example

```

context.addRoutes(new RouteBuilder() {
    @Override
    public void configure() {
        from("timer://foo?period=5000").routeId("diskmon")
            .setBody(constant("SELECT AVG(PERCENT_USED)FROM QSYS2.SYSDISKSTAT A"))
            .to("jdbc:jt400")
            .process((exchange) -> {
                Object diskUsageValue = ((ArrayList<HashMap<String, Object>>)exchange.getIn().getBody()).remove(0).values().iterator().next();
                exchange.getIn().setBody(diskUsageValue);
            })
            .choice()
                .when(body().isGreaterThan(90))
                    .process((exchange) -> {exchange.getIn().setBody("Average utilization of your disks is "+ exchange.getIn().getBody() + "%");})
                    .to("log:disk_space_mon?showAll=true&level=ERROR")
                    .to("smtp://mysmtpserver:1025?from=me@gmail.com&to=me@gmail.com&subject=Disk getting full!!")
                    .process((exchange) -> { pauseContext(exchange.getContext(), 1000*60*60); })
                .otherwise()
                    .wireTap("log:disk_space_mon?showAll=true&level=INFO");
    }
});

```

Translate the results to a simple number (Message Translator EIP)


```

process((exchange) -> {
    Object diskUsageValue = ((ArrayList<HashMap<String, Object>>)exchange.getIn().getBody()).remove(0).values().iterator().next();
    exchange.getIn().setBody(diskUsageValue);
});

```

Disk Monitor Example

Make a decision
(Content Based
Router EIP)

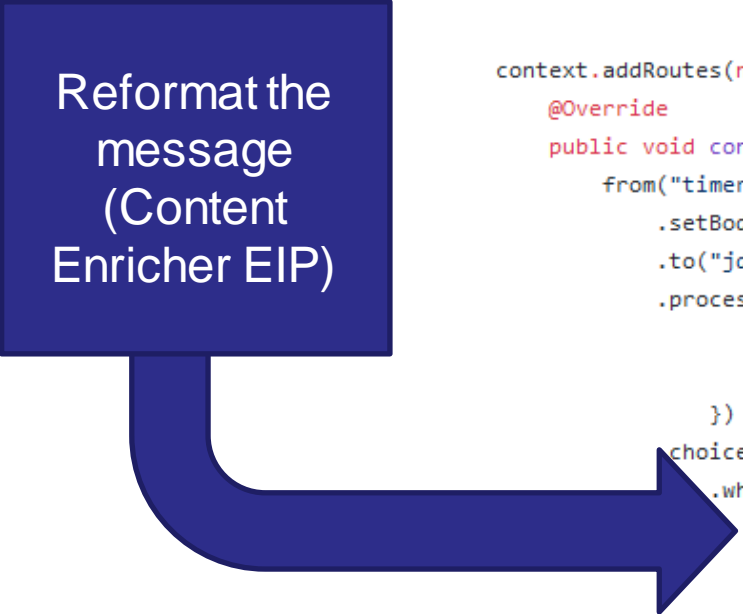


```
context.addRoutes(new RouteBuilder() {
    @Override
    public void configure() {
        from("timer://foo?period=5000").routeId("diskmon")
            .setBody(constant("SELECT AVG(PERCENT_USED)FROM QSYS2.SYSDISKSTAT A"))
            .to("jdbc:jt400")
            .process((exchange) -> {
                Object diskUsageValue = ((ArrayList<HashMap<String, Object>>)exchange.getIn().getBody()).remove(0).values().iterator().next();
                exchange.getIn().setBody(diskUsageValue);
            })
            .choice()
                .when(body().isGreaterThan(90))
                    .process((exchange) -> {exchange.getIn().setBody("Average utilization of your disks is "+ exchange.getIn().getBody() + "%");})
                    .to("log:disk_space_mon?showAll=true&level=ERROR")
                    .to("smtp://mysmtpserver:1025?from=me@gmail.com&to=me@gmail.com&subject=Disk getting full!!")
                    .process((exchange) -> { pauseContext(exchange.getContext(), 1000*60*60); })
                .otherwise()
                    .wireTap("log:disk_space_mon?showAll=true&level=INFO");
    }
});
```

```
choice()
    .when(body().isGreaterThan(90))
    .otherwise()
```

Disk Monitor Example

Reformat the message
(Content
Enricher EIP)

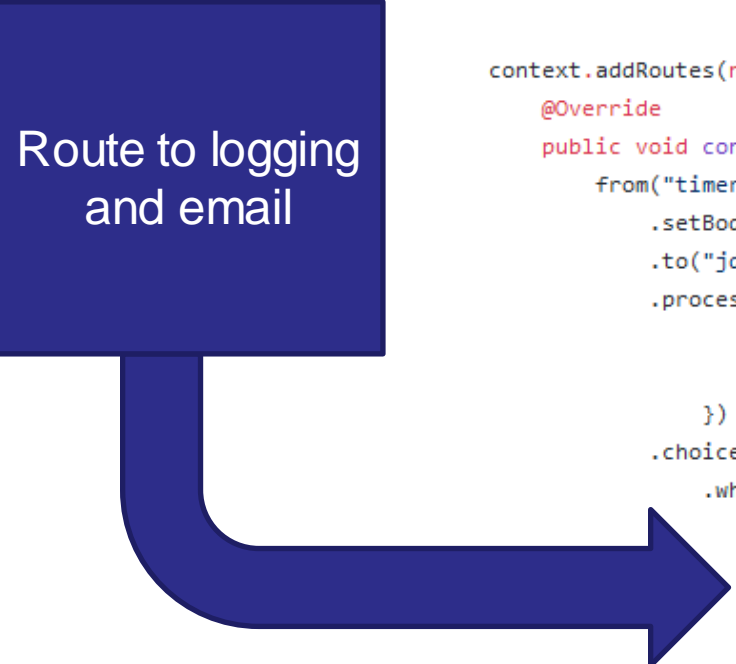


```
context.addRoutes(new RouteBuilder() {
    @Override
    public void configure() {
        from("timer://foo?period=5000").routeId("diskmon")
            .setBody(constant("SELECT AVG(PERCENT_USED)FROM QSYS2.SYSDISKSTAT A"))
            .to("jdbc:jt400")
            .process((exchange) -> {
                Object diskUsageValue = ((ArrayList<HashMap<String, Object>>)exchange.getIn().getBody()).remove(0).values().iterator().next();
                exchange.getIn().setBody(diskUsageValue);
            })
            .choice()
                .when(body().isGreaterThan(90))
                    .process((exchange) -> {exchange.getIn().setBody("Average utilization of your disks is "+ exchange.getIn().getBody() + "%");})
                    .to("log:disk_space_mon?showAll=true&level=ERROR")
                    .to("smtp://mysmtpserver:1025?from=me@gmail.com&to=me@gmail.com&subject=Disk getting full!!")
                    .process((exchange) -> { pauseContext(exchange.getContext(), 1000*60*60); })
                .otherwise()
                    .wireTap("log:disk_space_mon?showAll=true&level=INFO");
    }
});
```

```
.process((exchange) -> {exchange.getIn().setBody("Average utilization of your disks is "+ exchange.getIn().g
```

Disk Monitor Example

Route to logging
and email



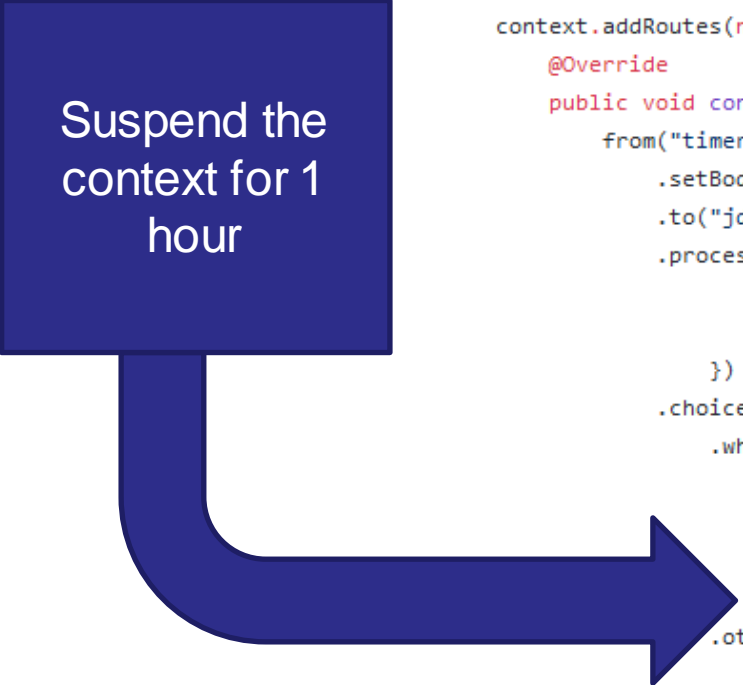
```
context.addRoutes(new RouteBuilder() {
    @Override
    public void configure() {
        from("timer://foo?period=5000").routeId("diskmon")
            .setBody(constant("SELECT AVG(PERCENT_USED)FROM QSYS2.SYSDISKSTAT A"))
            .to("jdbc:jt400")
            .process((exchange) -> {
                Object diskUsageValue = ((ArrayList<HashMap<String, Object>>)exchange.getIn().getBody()).remove(0).values().iterator().next();
                exchange.getIn().setBody(diskUsageValue);
            })
            .choice()
                .when(body().isGreaterThan(90))
                    .process((exchange) -> {exchange.getIn().setBody("Average utilization of your disks is "+ exchange.getIn().getBody() + "%");})
                    .to("log:disk_space_mon?showAll=true&level=ERROR")
                    .to("smtp://mysmtpserver:1025?from=me@gmail.com&to=me@gmail.com&subject=Disk getting full!!")
                    .process((exchange) -> { pauseContext(exchange.getContext(), 1000*60*60); })
                .otherwise()
                    .wireTap("log:disk_space_mon?showAll=true&level=INFO");
    }
});
```

```
.to("log:disk_space_mon?showAll=true&level=ERROR")
```

```
.to("smtp://mysmtpserver:1025?from=me@gmail.com&to=me@gmail.com&subject=Disk getting full!!")
```


Disk Monitor Example

Suspend the
context for 1
hour



```
context.addRoutes(new RouteBuilder() {
    @Override
    public void configure() {
        from("timer://foo?period=5000").routeId("diskmon")
            .setBody(constant("SELECT AVG(PERCENT_USED)FROM QSYS2.SYSDISKSTAT A"))
            .to("jdbc:jt400")
            .process((exchange) -> {
                Object diskUsageValue = ((ArrayList<HashMap<String, Object>>)exchange.getIn().getBody()).remove(0).values().iterator().next();
                exchange.getIn().setBody(diskUsageValue);
            })
            .choice()
                .when(body().isGreaterThan(90))
                    .process((exchange) -> {exchange.getIn().setBody("Average utilization of your disks is "+ exchange.getIn().getBody() + "%");})
                    .to("log:disk_space_mon?showAll=true&level=ERROR")
                    .to("smtp://mysmtpserver:1025?from=me@gmail.com&to=me@gmail.com&subject=Disk getting full!!")
                    .process((exchange) -> { pauseContext(exchange.getContext(), 1000*60*60); })
                .otherwise()
                    .wireTap("log:disk_space_mon?showAll=true&level=INFO");
    }
});
```

```
.process((exchange) -> { pauseContext(exchange.getContext(), 1000*60*60); })
```

EIPs in Use

<https://camel.apache.org/components/latest/eips/setBody-eip.html>

[https://camel.apache.org/components/3.4.x/eips/message-
translator.html](https://camel.apache.org/components/3.4.x/eips/message-
translator.html)

[https://camel.apache.org/components/3.4.x/eips/content-based-
router-eip.html](https://camel.apache.org/components/3.4.x/eips/content-based-
router-eip.html)

[https://camel.apache.org/components/3.4.x/eips/content-
enricher.html](https://camel.apache.org/components/3.4.x/eips/content-
enricher.html)

<https://camel.apache.org/components/3.4.x/eips/wireTap-eip.html>

Deploying Your First Simple Routes

- Can visit the "IBM i OSS Examples" repo at <https://github.com/IBM/ibmi-oss-examples/>
 - Note the "camel" subdirectory: <https://github.com/IBM/ibmi-oss-examples/tree/master/camel>
- Examples currently include
 - A data queue to Kafka bridge (with information about using this as a Db2->Kafka bridge)
 - Requires you to have Kafka somewhere
 - A message queue to email bridge
 - Requires you to have an SMTP server
 - Disk monitor email example
 - Requires you to have an SMTP server
 - Disk monitor message queue example (send messages to *SYSOPR message queue)
 - Should work "out of the box"

Deploying Your First Simple Routes

- Examples use Java and Maven
 - Maven has a central repository of packages
 - Configuration of project in 'pom.xml' file
 - Feel free to dig in the examples to learn more

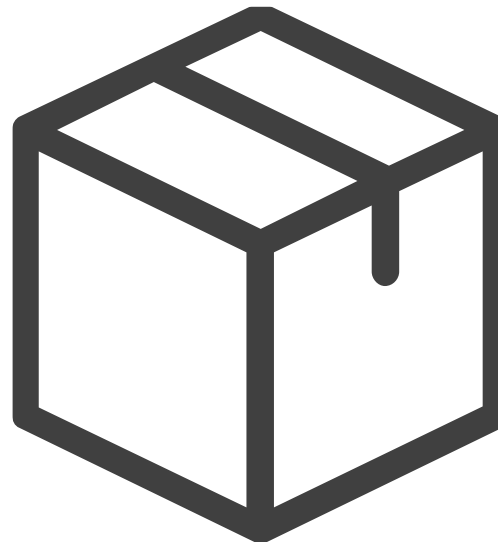
MavenTM

The logo for Maven, featuring the word "Maven" in a bold, black, sans-serif font. The letter "v" is replaced by a stylized feather with a gradient of colors from purple at the base to orange at the tip. A small "TM" trademark symbol is positioned to the upper right of the word.

Deployment

Camel Deployment

- It can be deployed as a **Java Web App** using Spring or POJO
- **Standalone Camel routes** can also be deployed using Maven and Camel (best starting point)
- Spring Boot can launch Camel routes
- Camel can be **embedded** within another Java application
- Or **within a Java-based container** such as Jetty, Tomcat, JBoss/Wildfly, and others
- **ActiveMQ contains the Camel core** libraries and can launch and deploy Camel routes
- And **OSGI containers** like Karaf or its Servicemix counterpart can launch Camel routes
- Standalone Camel routes **scale very nicely in microservice containers** as well



Running a Standalone Java Camel Application/Route as a Service

CL

Submit to batch

PASE/SSH

Install coreutils-gnu and use "nohup"

PASE/SSH

Use ServiceCommander

- <https://github.com/ThePrez/ServiceCommander-IBMi>
- Define the program in a .yaml file (or use "scinit" utility)
- TIP: For POJO camel applications, submit to batch with a custom job name

Getting Help

IBM OSS Support! Example “Supportables” for IBM i

Git
 Jenkins
 Rsync
 Ansible
 Node.js
 PHP
 Apache Tomcat
 WordPress
 Python
 R
 Apache ActiveMQ
 Apache Camel
 Apache Kafka
 Apache Zookeeper



APACHE
 Camel



For more resources, see:

<http://ibm.biz/ibmi-oss-support>

Community Portals



IBM i OSS Ryver chat

<http://ibm.biz/ibmiooss-chat>
(join at <http://ibm.biz/ibmiooss-chat-join>)



ZulipChat

<https://camel.zulipchat.com/>



Mailing Lists

<https://camel.apache.org/manual/atest/mailling-lists.html>