Integrate IBM i with Anything using Apache Camel

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

Agenda

- What's the problem?
- Camel as a Solution
- How to?
- A Complex Example
- Camel Deployment
- Getting Help

The Problem

Quick Exercise…

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

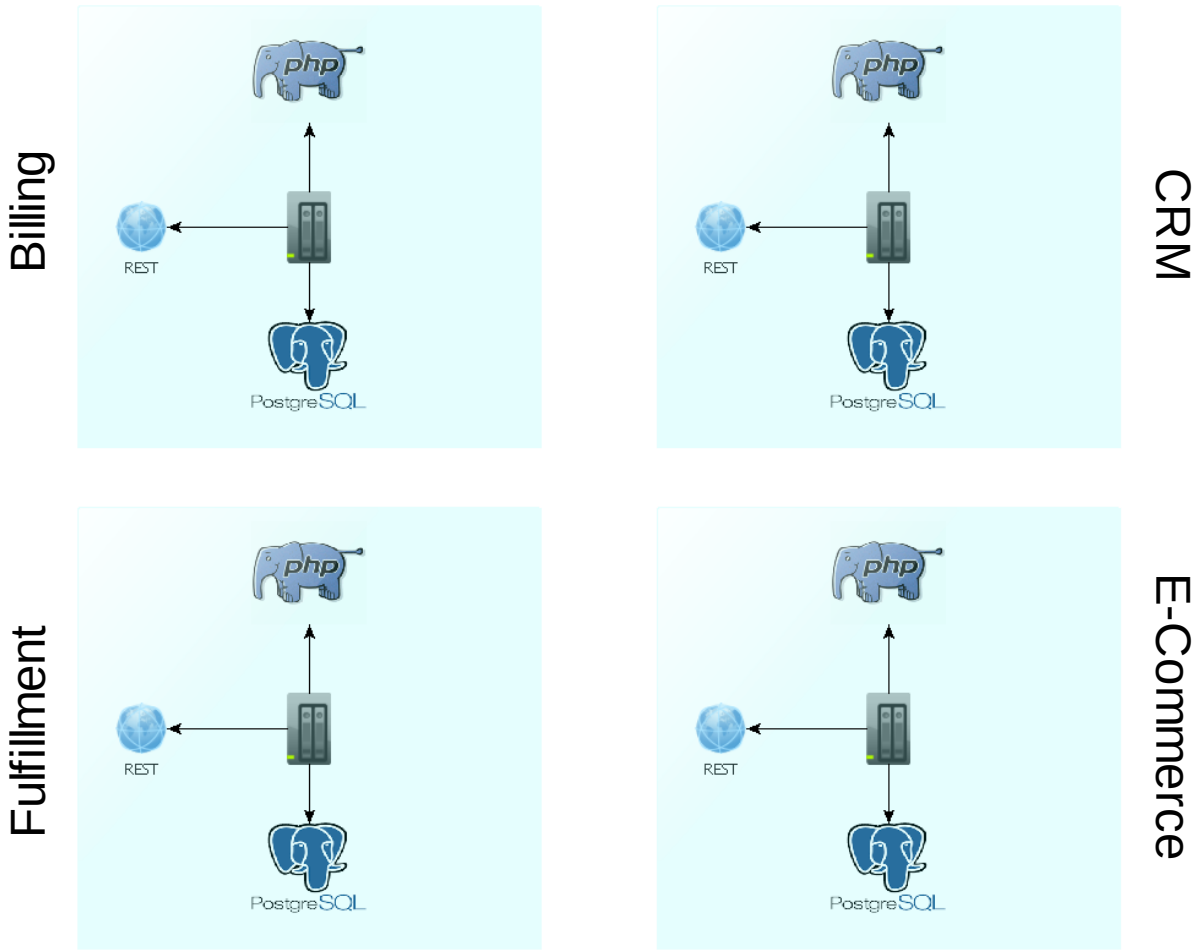
They all have something in **common**…

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!**

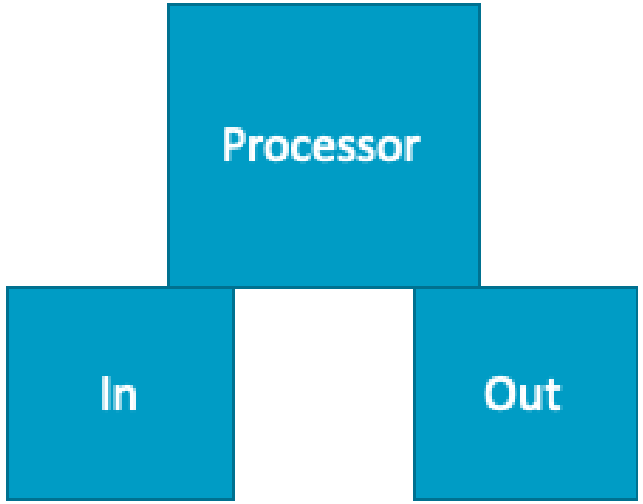
ESB Capabilities

- 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

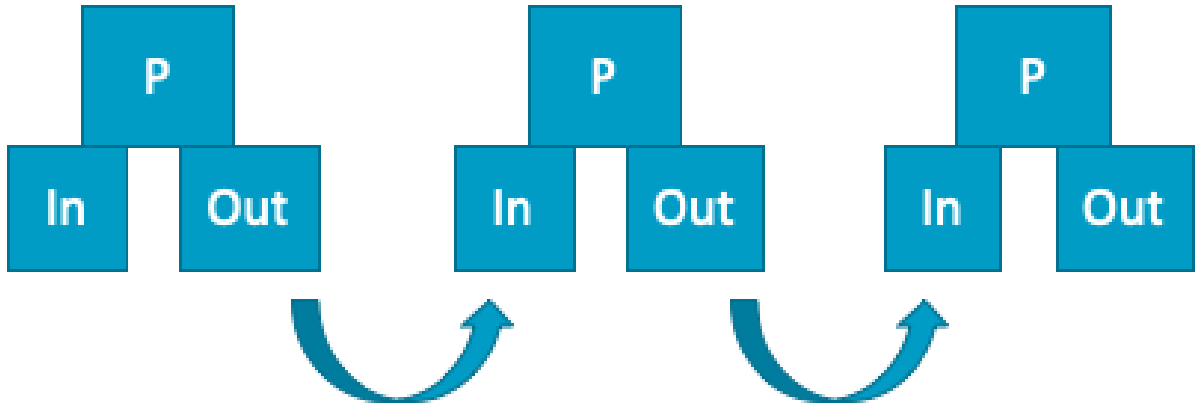
How Does It Work?

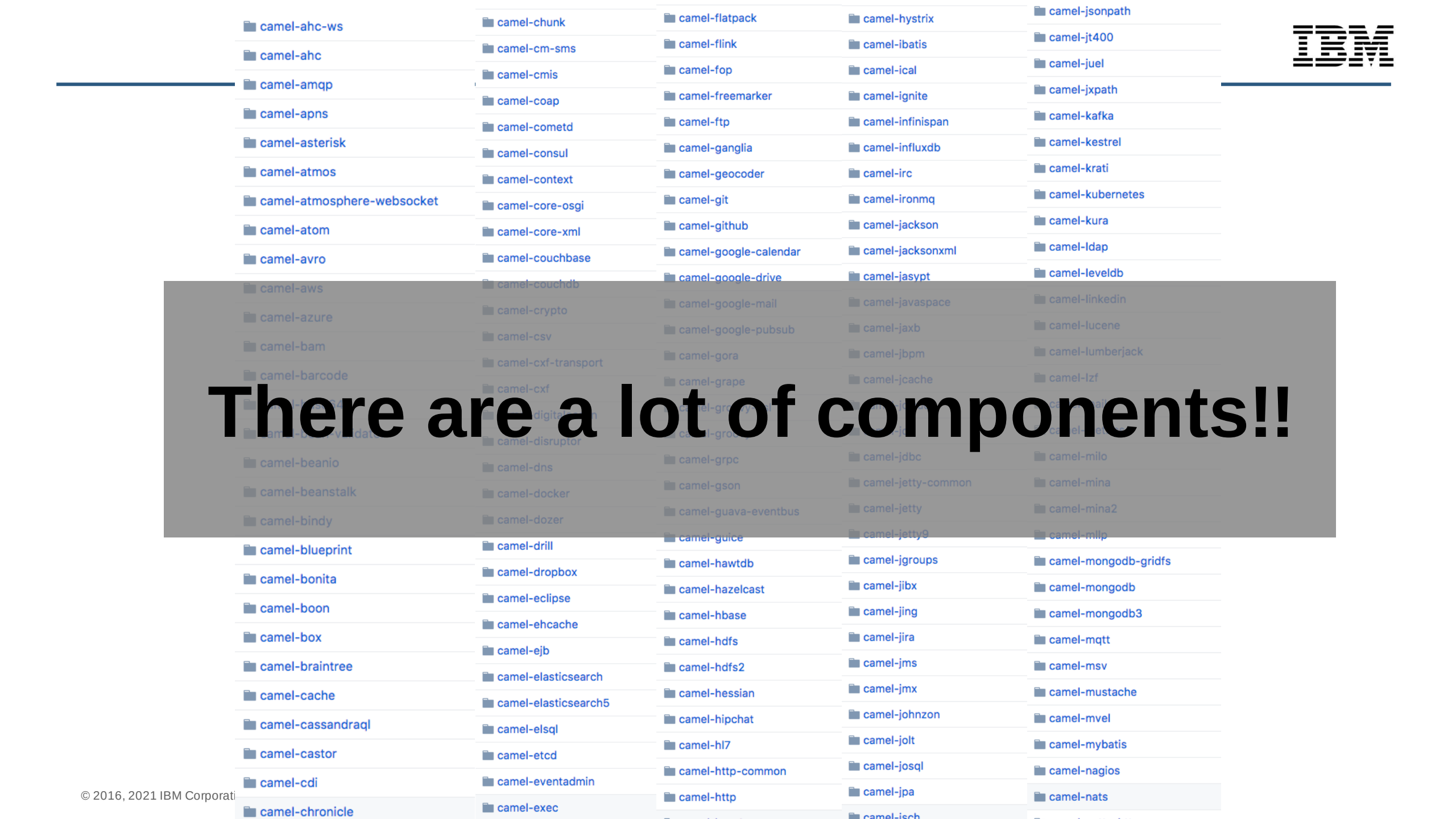
- Camel uses a repeatable, normalized concatenation of "Processor" and "Message" objects in a group called an Exchange
- There is an "in" message, a "Processor", and an "Out" message



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





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

New!

```
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

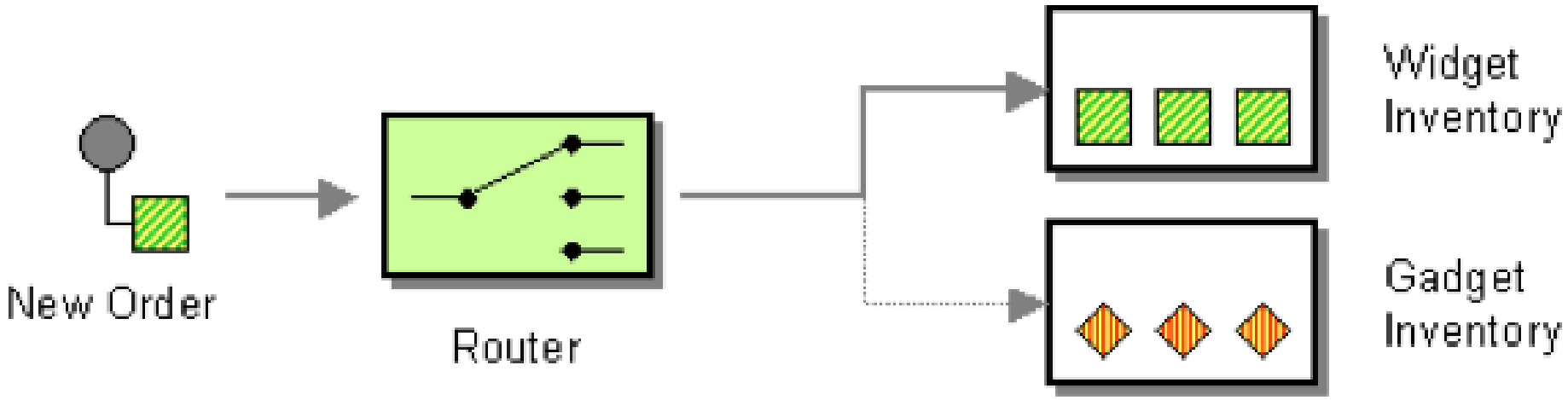
```
from("activemq:foo").filter().
  xquery("//foo").
  to("activemq:bar")
```

- 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>
```

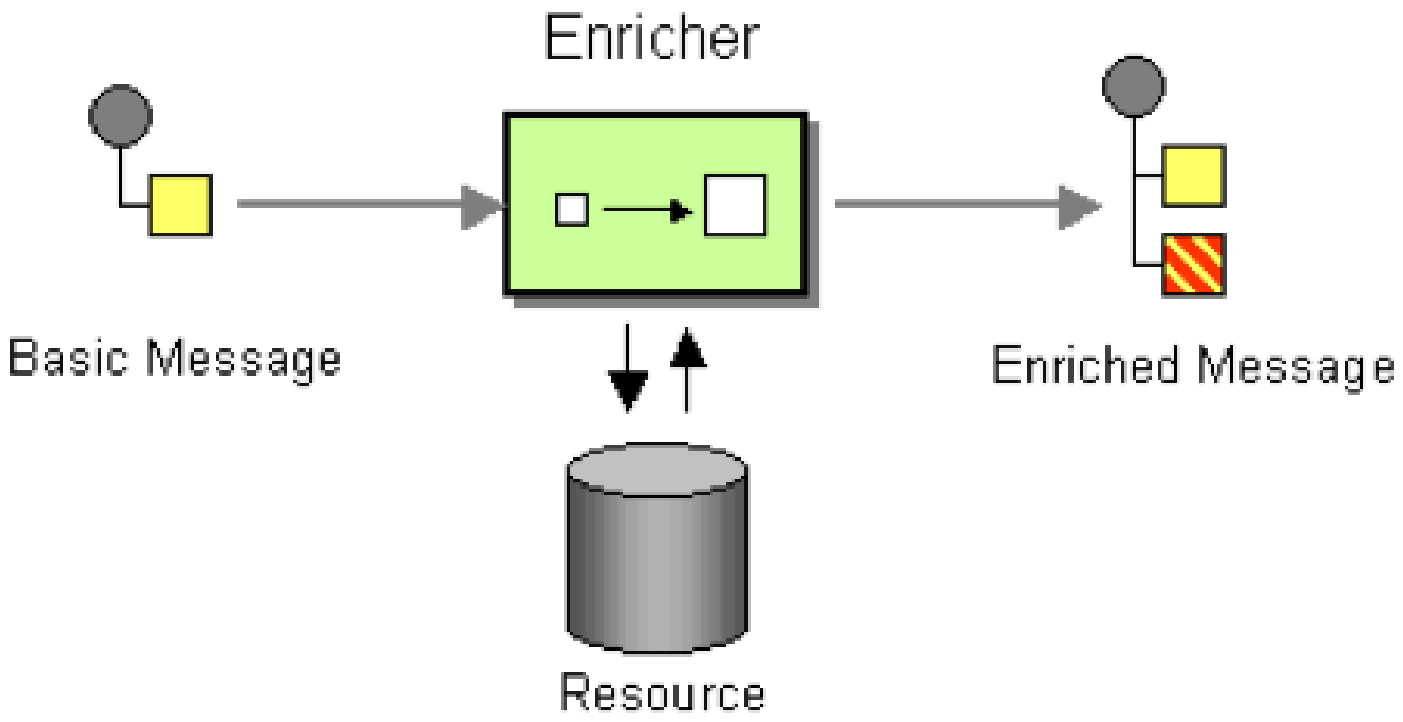
Integration Patterns – Content Based Router

- Message routing based on message criteria



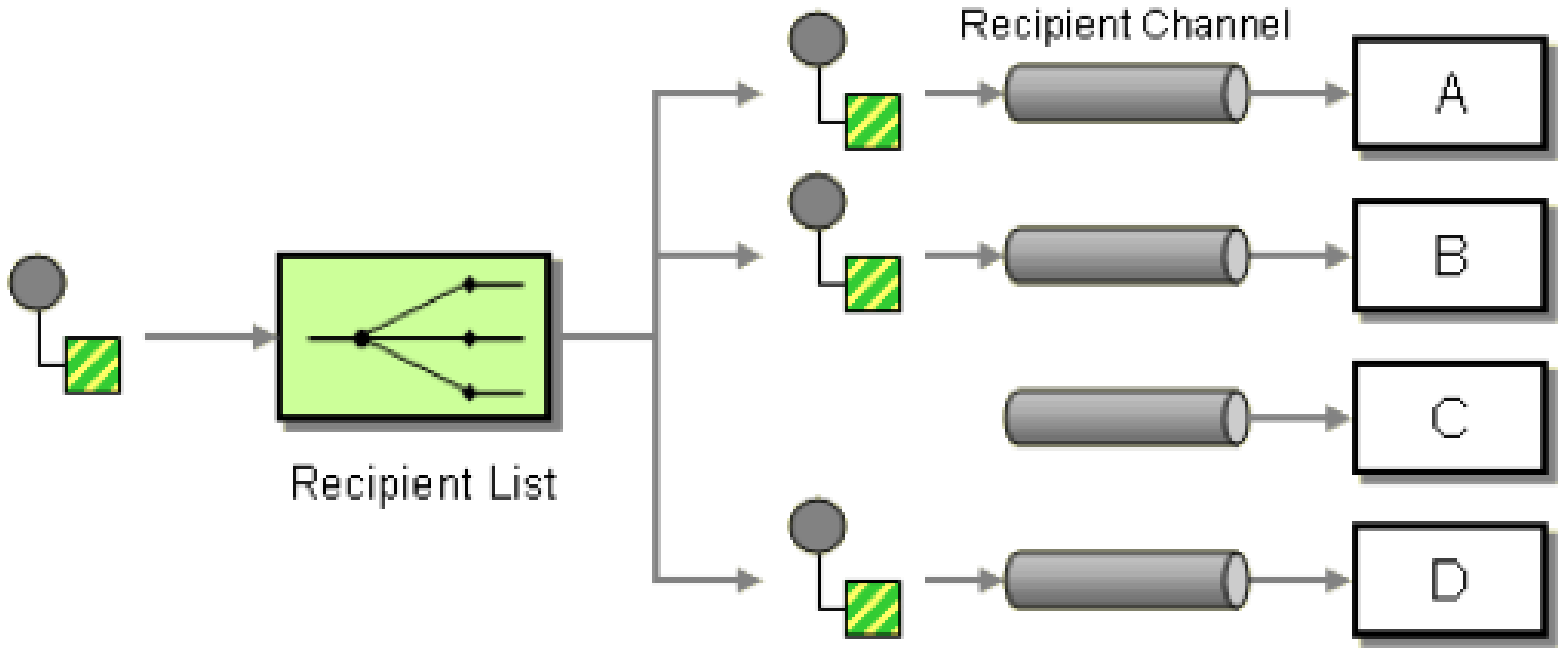
Integration Patterns – Content Enricher

- Pattern adds data to message from another source



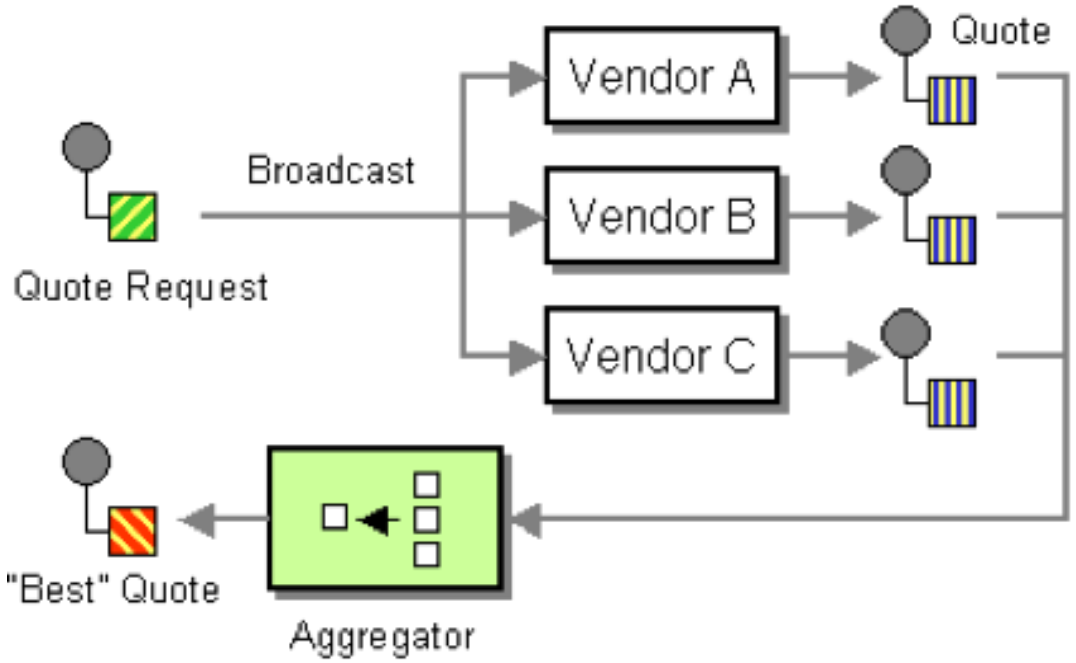
Integration Patterns – Recipient List

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



Integration Patterns – Scatter-Gather

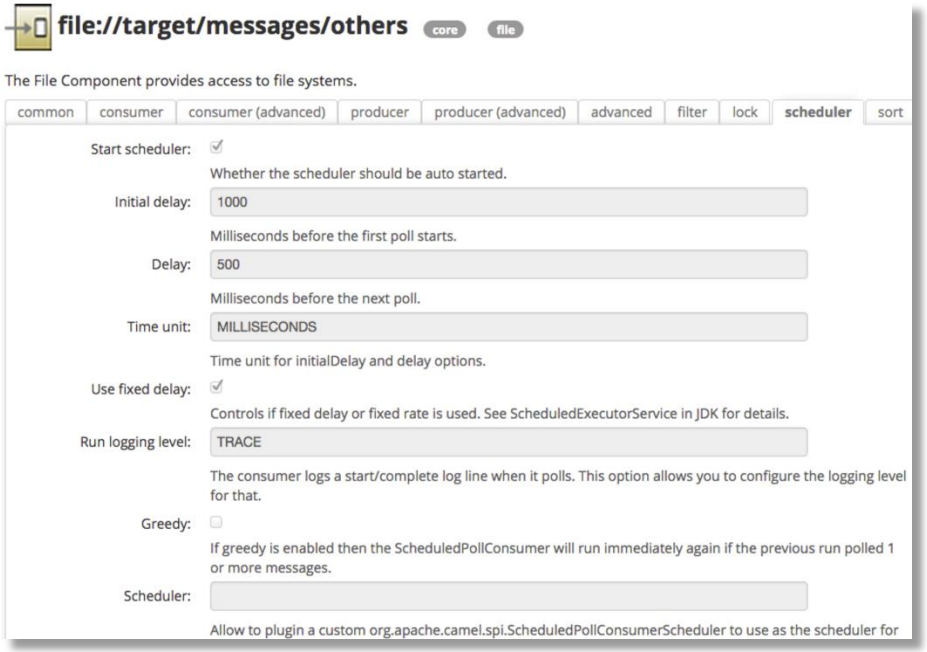
- Messages broadcast to disparate endpoints and results aggregated into a single message

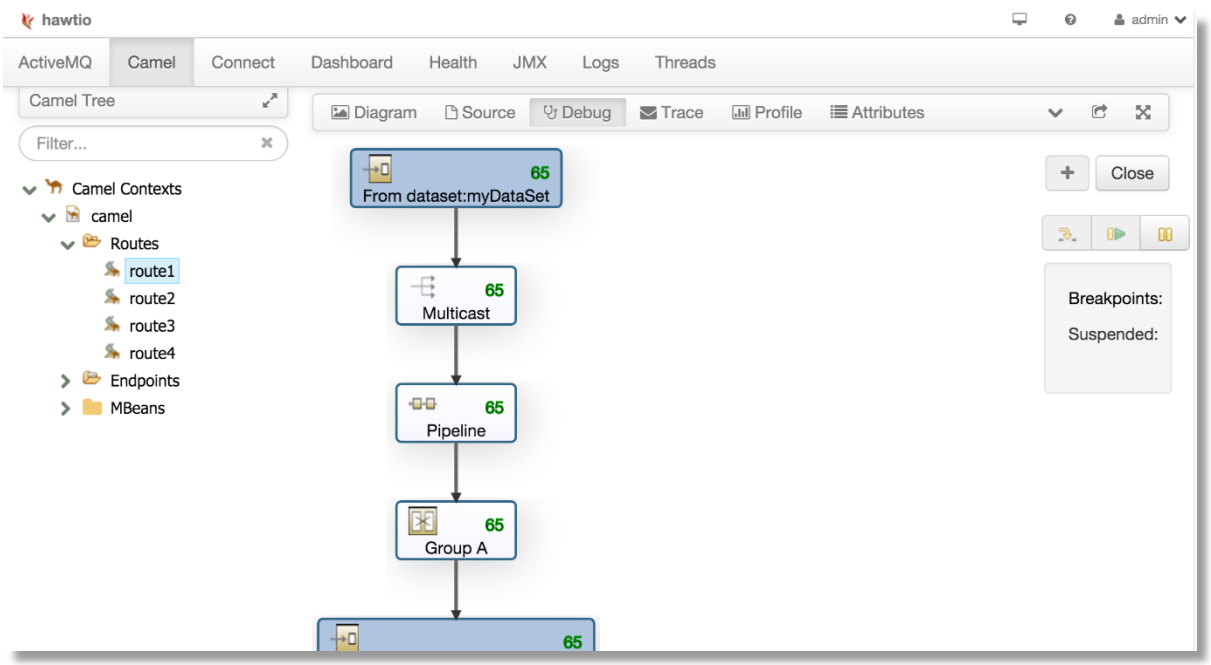


How to? (Java DSL emphasis)

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

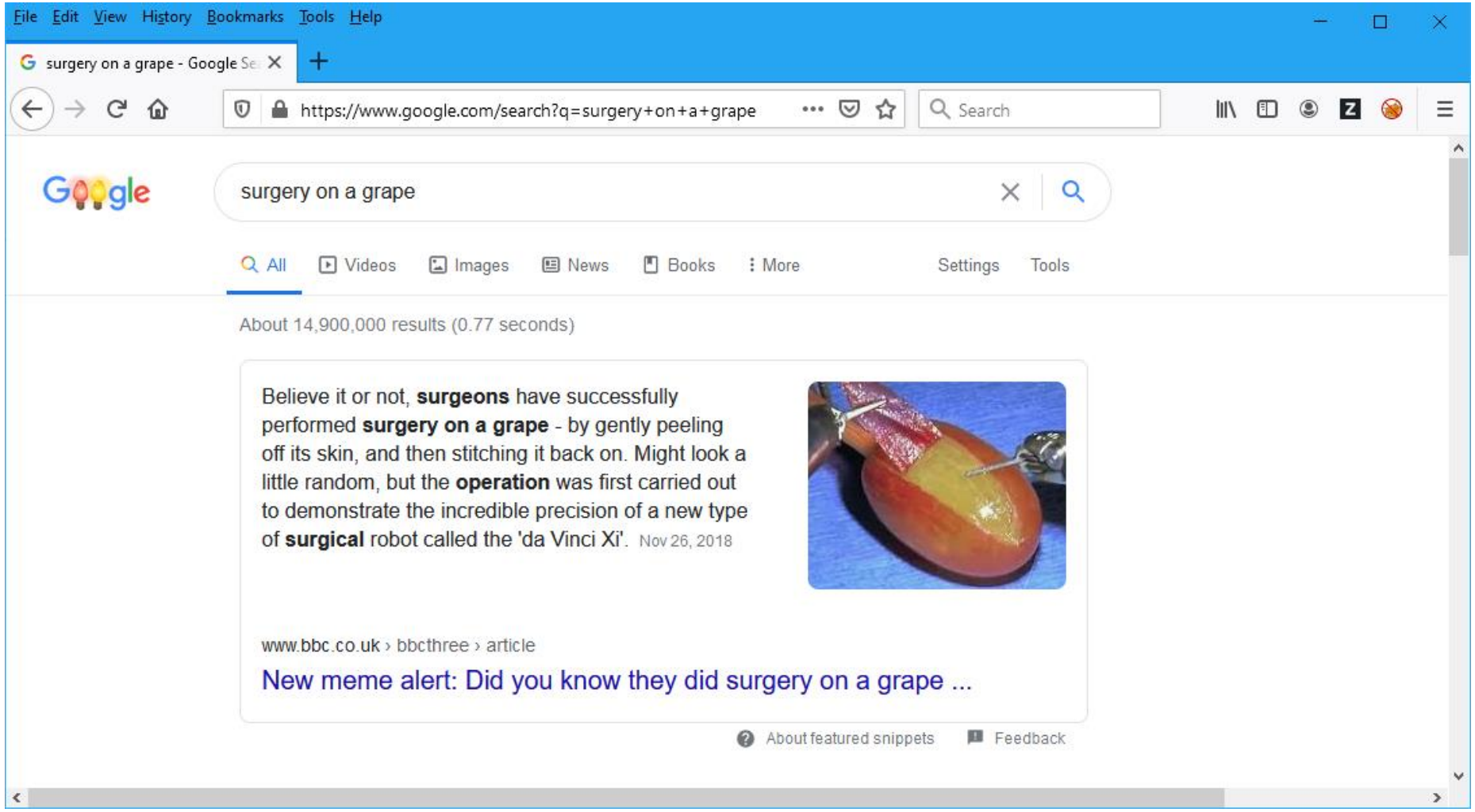




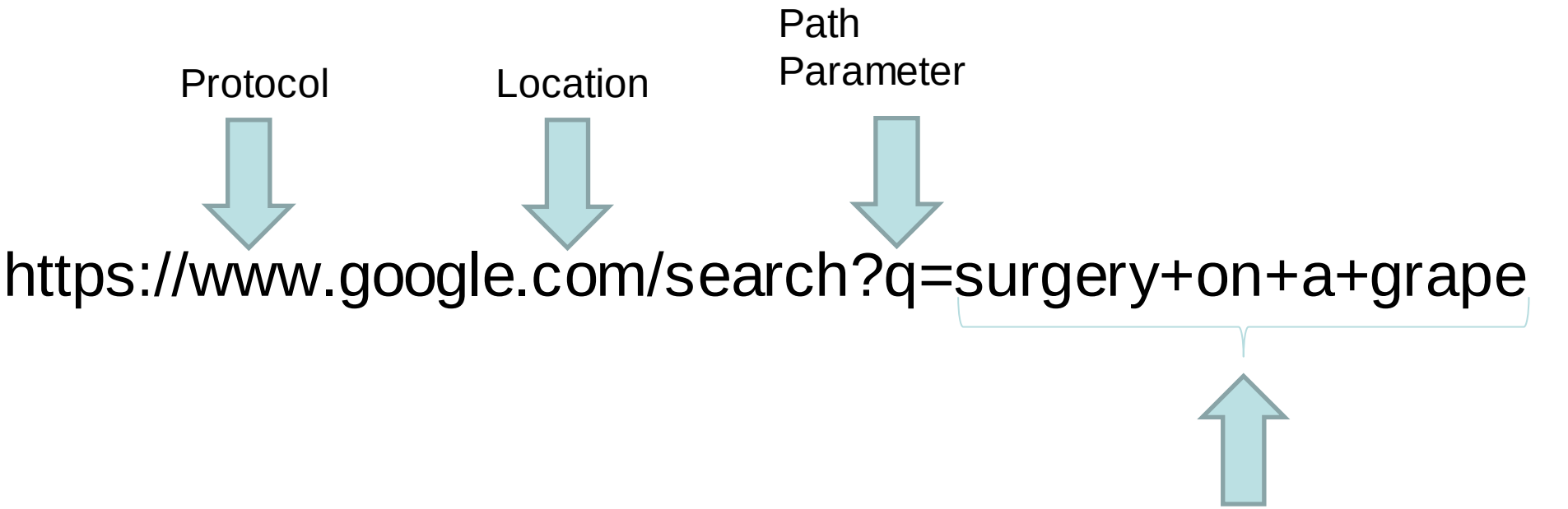
Creating/Configuring Endpoints

- 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)



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

```
jt400://user:password@system/QSYS.LIB/LI
```

To call remote program

```
jt400://user:password@system/QSYS.LIB/LI
```

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 a binary
guiAvailable (common)	Sets whether IBM i prompting is enabled in the environment run
keyed (common)	Whether to use keyed or non-keyed data queues.
searchKey (common)	Search key for keyed data queues.

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 syste
type	**Required** Whether to work with data queues or remote MSGQ

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);
    }
});
```

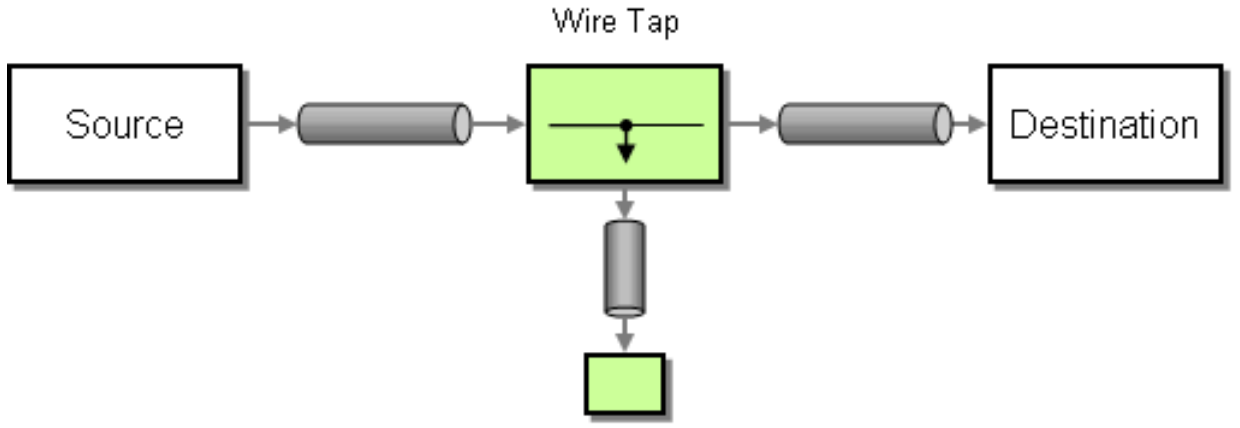
Message Queue to Email Bridge

- 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 :)

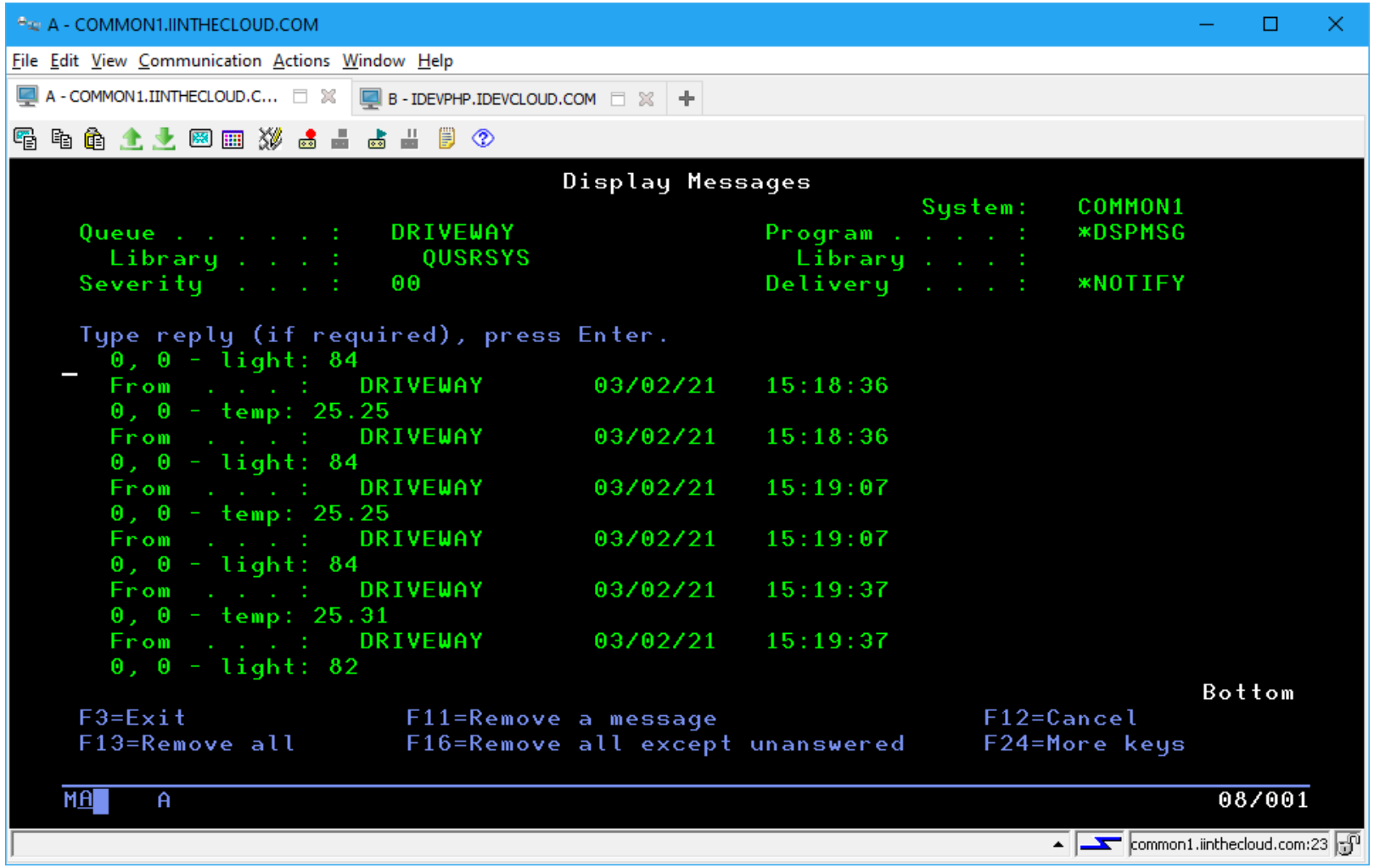


Data Queue to Kafka Bridge

- 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);
    }
});
```

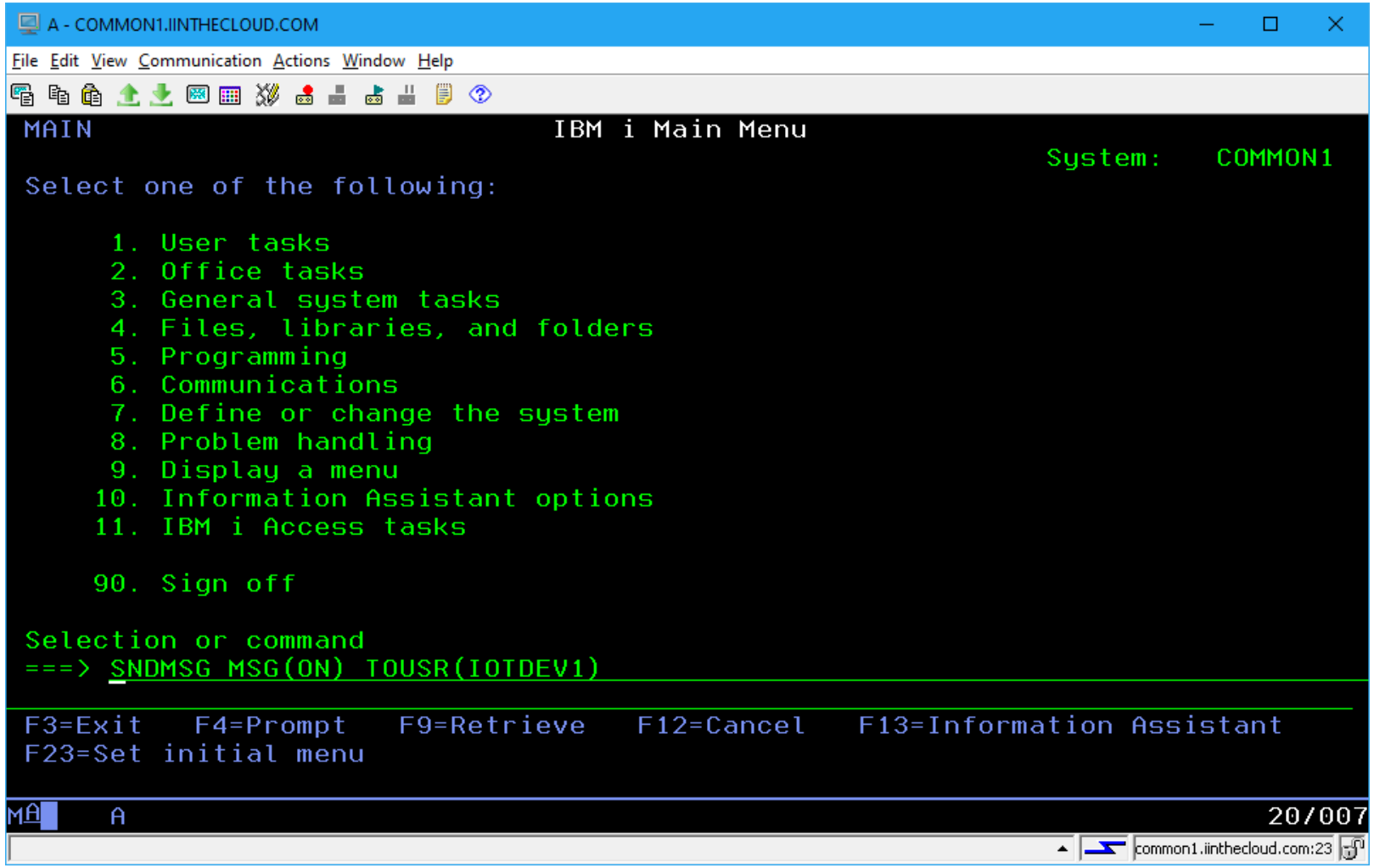
Consume IoT Data? No Problem!



Consume IoT Data? No Problem!

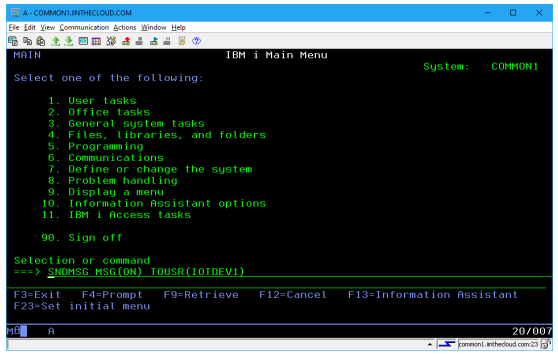
```
context.addRoutes(new RouteBuilder() {
    @Override
    public void configure() {
        from("paho:factory/1/light?brokerUrl=ssl://localhost")
        .to("jt400://driveway:xxxxxx@localhost/qsys.lib/QUSRSYS.lib/DRIVEWAY.MSGQ?guiAvailable=false");
    }
});
context.addRoutes(new RouteBuilder()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

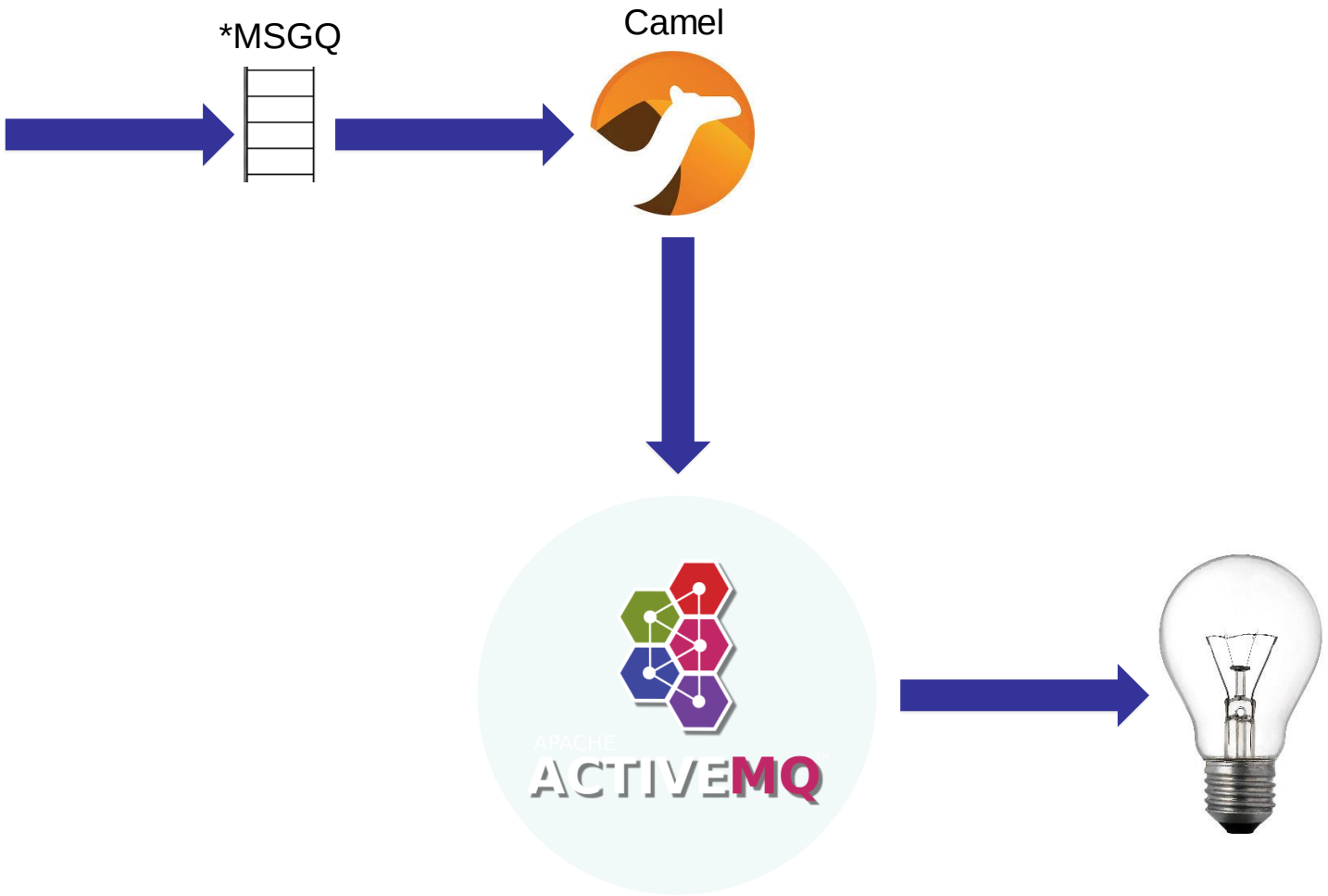
Control IoT Devices? No Problem!



Control IoT Devices? No Problem!

IBM i SNDMSG Command





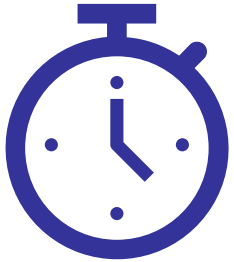
A Complex Example

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

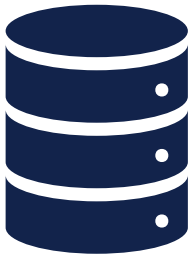
Timer Endpoint



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

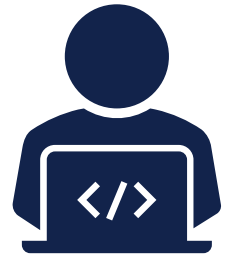
https://camel.apache.org/components/latest/timer-component.html

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

```
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");
    }
});
```

Set the body of the message (SQL)

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

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");
    }
});
```

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

```
.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().getE
        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/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/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"

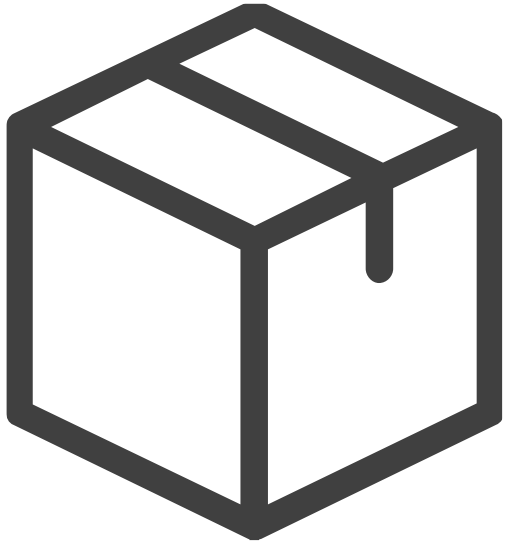
Deploying Your First Simple Routes

- 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

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

For more resources, see: http://ibm.biz/ibmi-oss-support

Community Portals

IBM i OSS Ryver chat

http://ibm.biz/ibmioss-chat
(join at http://ibm.biz/ibmioss-chat-join)

ZulipChat

https://camel.zulipchat.com/

Mailing Lists

https://camel.apache.org/manual/latest/mailing-lists.html