Kafka, Camel, and other IBM i integration techniques

Presented by:
Jesse Gorzinski
jgorzins@us.ibm.com
Twitter: @IBMJesseG

IBM i

IBM

1

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**!

2

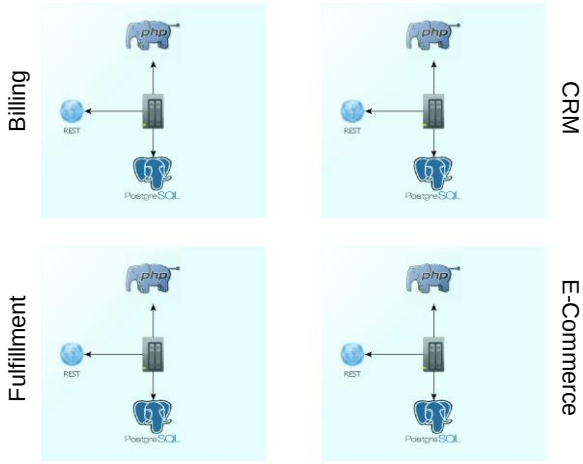
The Chupacabra

3

A Magical Unicorn…

4

A Fully Homogeneous Enterprise Landscape…



5

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

6

IBM

Agenda

7

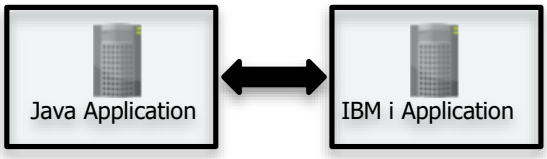
IBM

Apache Kafka

8

Traditional Enterprise Messaging

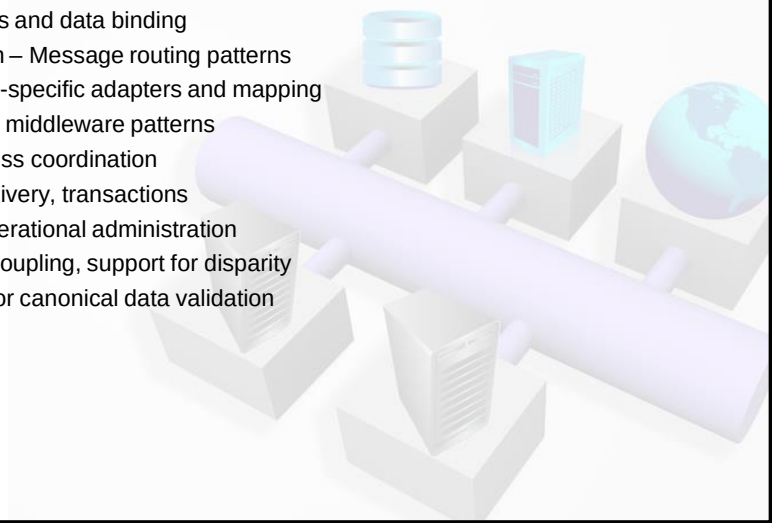
- Applications often have a need to send information back and forth to one another in a normalized fashion
- Before the advent of messaging systems, it was often difficult if not impossible to “federate” applications written in disparate languages or residing on heterogeneous platforms
- For instance, JMS, the Java Messaging Service, arose out of a growing need to federate very different systems with each other



9

ESB Capabilities

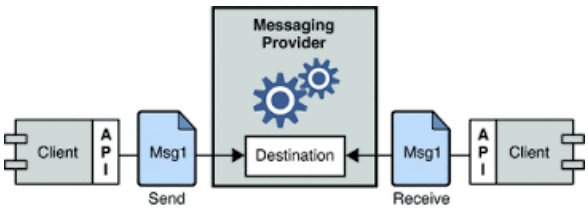
- In general, an ESB should provide the following functionality:
  - Transport Invocation – protocols and data binding
  - Data routing and transformation – Message routing patterns
  - Platform mediation – Language-specific adapters and mapping
  - Messaging – Message oriented middleware patterns
  - 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



10

Message oriented middleware

- If you don't need a full ESB, you may still be making use of (and paying for) commercial messaging
- Technically, MOM is any platform that sends and receives messages between distributed applications
- Commercial versions include IBM MQ, Amazon SQS, and Oracle AQ, many others
- This pattern allows for asynchronous processing, and normalization of data exchanges
- Clients connect to a messaging provider, and send and receive messages via that provider



11

Meet: Some Really Good, "Free" Solutions!

- Apache ActiveMQ is a JMS implementation, and can help with both federation of systems and asynchronous processing
- Apache Camel is a sophisticated message oriented middleware language, with hundreds of integration components
- Apache Kafka is a high-throughput streaming event engine suitable for very large datasets
- So…

Apache ACTIVEMQ

APACHE Camel

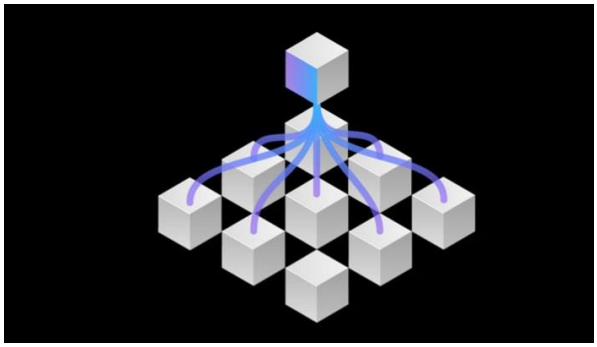
kafka

12

IBM

Hybrid Multicloud

- What needs drove Zookeeper, ActiveMQ, Kafka?
  - Data federation
  - Edge computing
  - Distributed computing
  - Large data workloads



13

Hybrid cloud is the new normal

98%

of organizations will use multiple hybrid cloud environments by 2021

Source: IBM Institute of Business Value A field guide to multicloud management

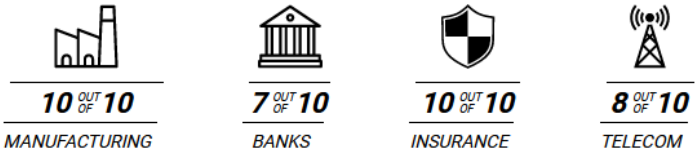
Movement between clouds	73%	priority concern
Connectivity between clouds	82%	priority concern
Consistency of management	67%	priority concern

Sources: IDC Cloud Forecast; BCG & McKinsey

4

14

Industry interest in Kafka



10/10 Largest insurance companies
10/10 Largest manufacturing companies
10/10 Largest information technology and services companies
8/10 Largest telecommunications companies
8/10 Largest transportation companies
7/10 Largest retail companies
7/10 Largest banks and finance companies
6/10 Largest energy and utilities organizations

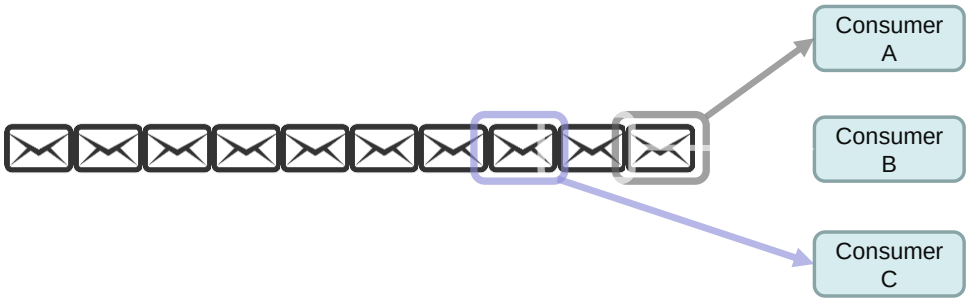
Analysis of key industries using Kafka. (Source: https://kafka.apache.org)

Analysis is based on the 10 largest companies in each sector.

15

Streaming vs. Traditional Message Queuing

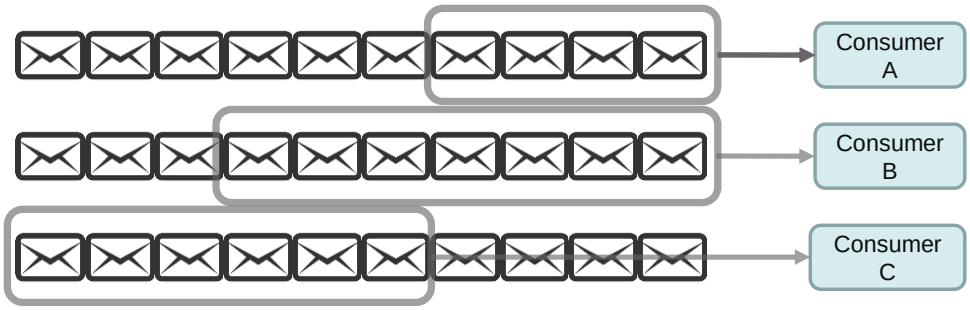
- Primarily, the difference comes down to how messages are processed by receiving systems
- In traditional message queueing (not topics), a single message is processed at a time, even if that payload contains a lot of data:



16

Streaming or Stream Processing

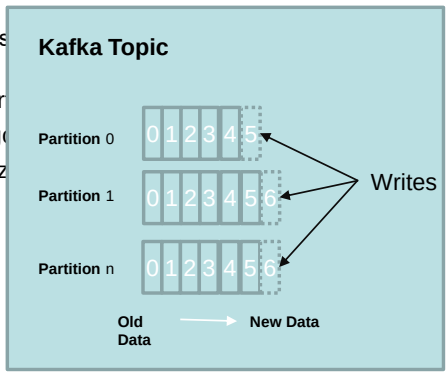
- Streaming differs in that chunks of data, or series of messages tend to be processed or at least referenced at a time
- Historical data is usually an option as well, and consumers can time slice the data they want to receive



17

Partitions

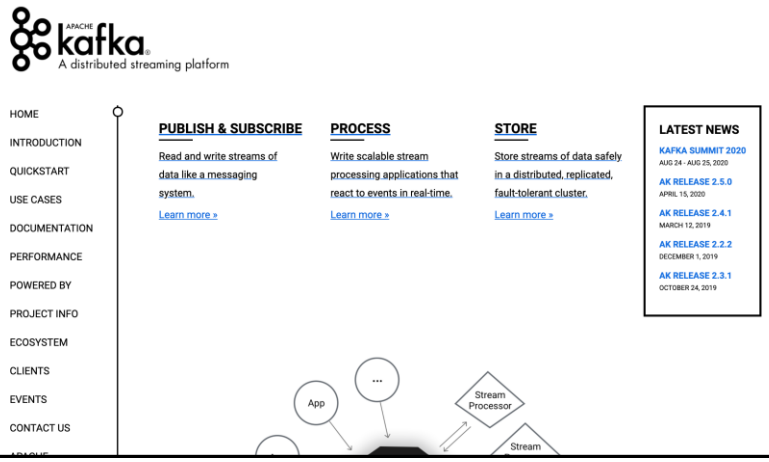
- Kafka stores streamed data in Partitions, which are on-disk logical groupings of writes from Producing applications
- Notice that “new” writes are written to the end of these partitions and other traditional FIFO messaging solutions
- Consuming applications will be able to subscribe to a single par
- This allows for excellent redundancy in retaining the data and g
- It also provides a straightforward mechanism for achieving horiz



18

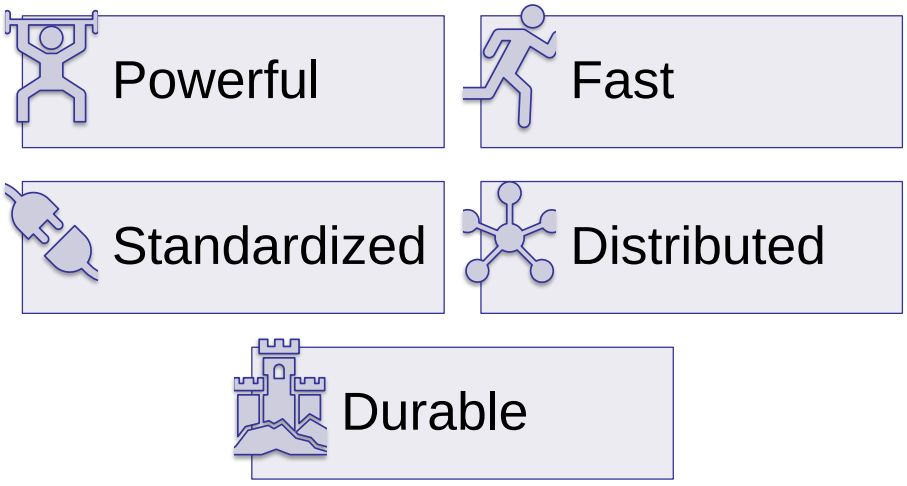
Obtaining Kafka

- Kafka is maintained and distributed from its top-level community home, https://kafka.apache.org:



19

OK, but why is Kafka such a big deal?



20

Use cases

- Artificial intelligence
- Infrastructure monitoring
- Transaction monitoring
- Real-time reporting
- Microservices
- Fraud/anomaly detection
- Heterogeneous applications

21

OK, but how do I stream/consume data?

- https://ibmi-oss-docs.readthedocs.io/en/latest/kafka/README.html
- Db2 Triggers and Apache Camel: stream events in real-time
- Kafka Connect JDBC Source connector: Simple, polling-based technique
- InfoSphere Data Replication and the CDC Replication Engine for Kafka
- Native ILE Kafka client (unsupported): call Kafka functions directly from ILE programs.
- Confluent Platform
  - ksqlDB, which provides an SQL interface
  - Kafka REST APIs, which provide a REST interface

22

Apache Camel

23

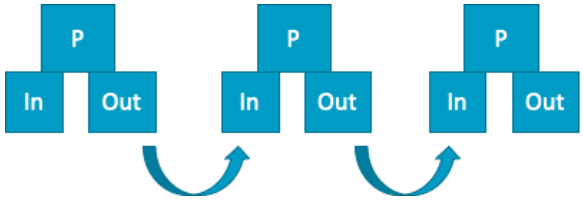
What is Apache Camel

- A Java-based integration framework
- As Jesse says, "it can be used to connect anything to anything"
- Information about Camel with IBM i:
  - https://ibmi-oss-docs.readthedocs.io/en/latest/camel/README.html

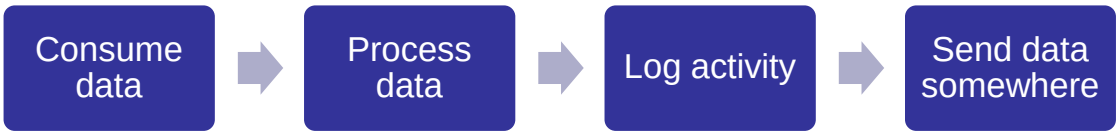
24

How Does It Work?

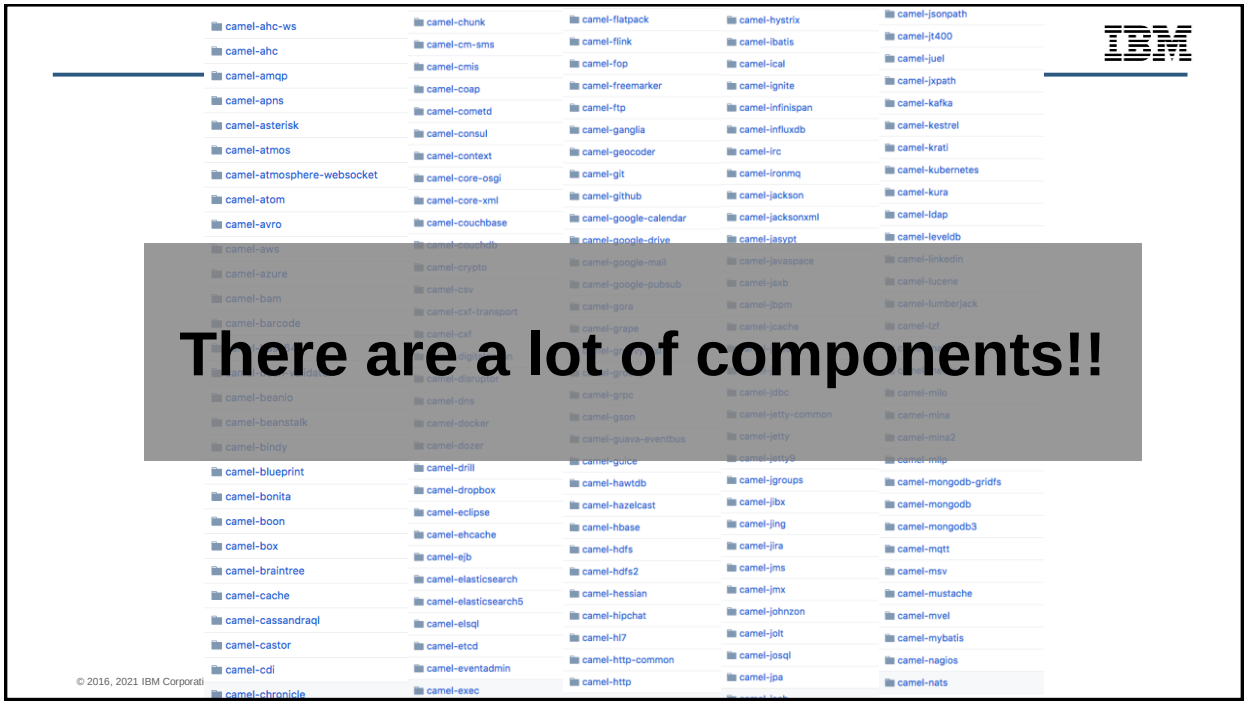
- Exchanges can be chained together – like piping commands through *NIX – to form a Camel Route
- The "Out" message of a previous Exchange becomes the "in" message of a new Exchange
- This defines the route



25



26



There are a lot of components!!

27

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

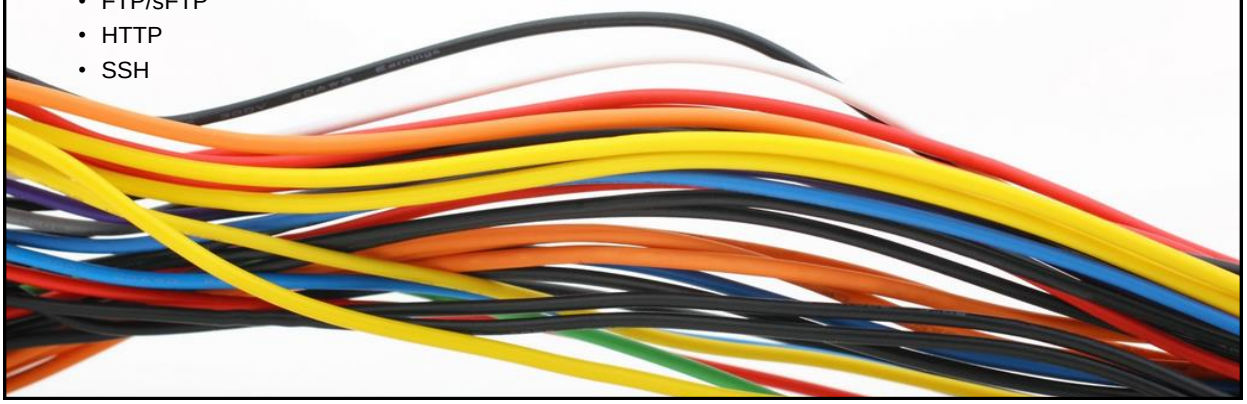
```
jt400://user:password@system/QSYS.LIB/LIBRARY.LIB/program.PGM[?options]
```

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

28

Many other components for interacting with IBM i, for instance…..

- JDBC/SQL
- Paho (mqtt/IoT)
- Exec (execute a command)
- File
- FTP/sFTP
- HTTP
- SSH



29

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');
```


30

Camel Source

- Camel routes can be coded in Java DSL

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

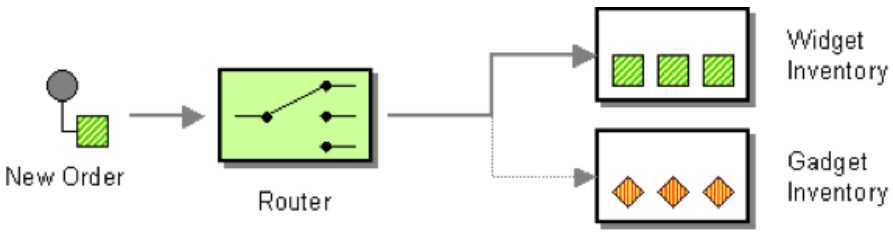
- Or by using Camel Spring DSL XML files

```
<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>
```


31

Integration Patterns – Content Based Router

- Message routing based on message criteria

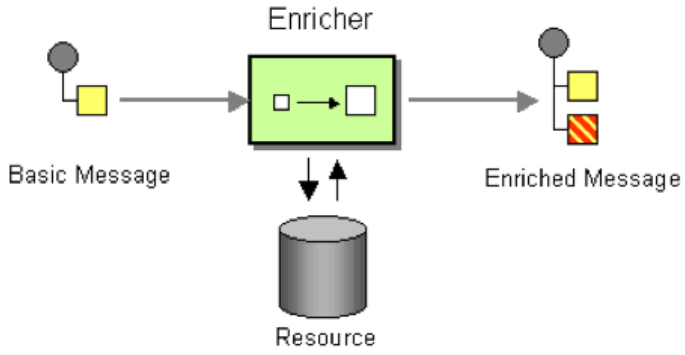


32

IBM

Integration Patterns – Content Enricher

- Pattern adds data to message from another source

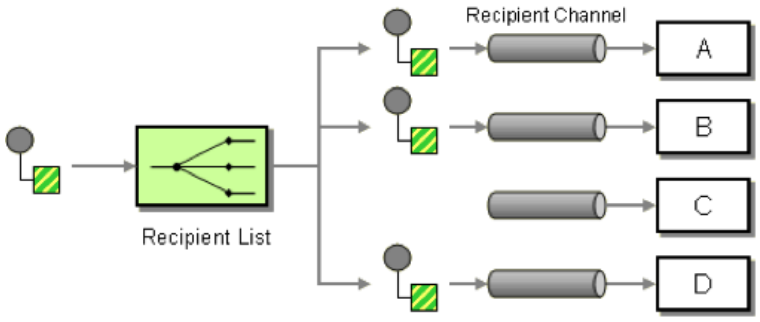


33

IBM

Integration Patterns – Recipient List

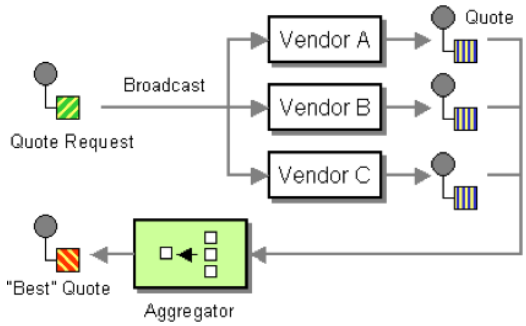
- Message broadcast to *n* channels based on message criteria
- Broadcast to all channels == Multicast Pattern



34

Integration Patterns – Scatter-Gather

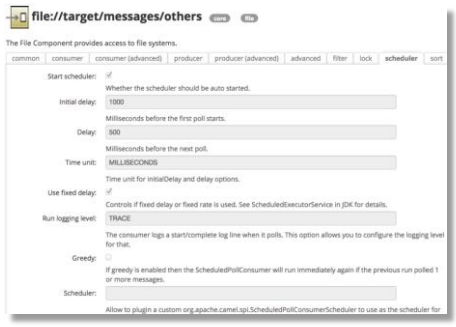
- Messages broadcast to disparate endpoints and results aggregated into a single message

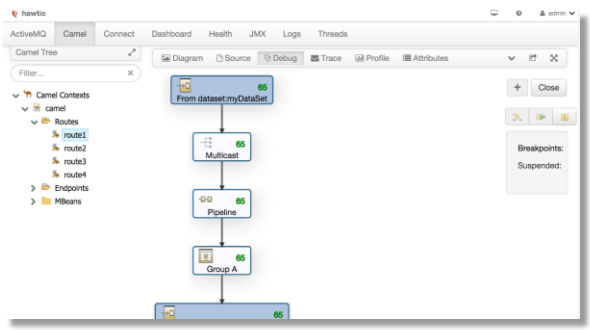


35

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





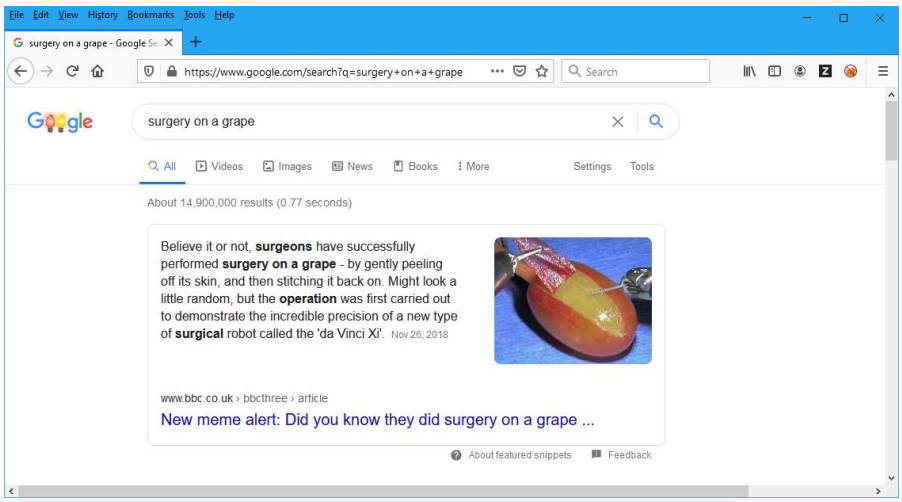
36

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)

37

URI Example (URL is a type of URI)



38

URI Example

Protocol

Location

Path Parameter

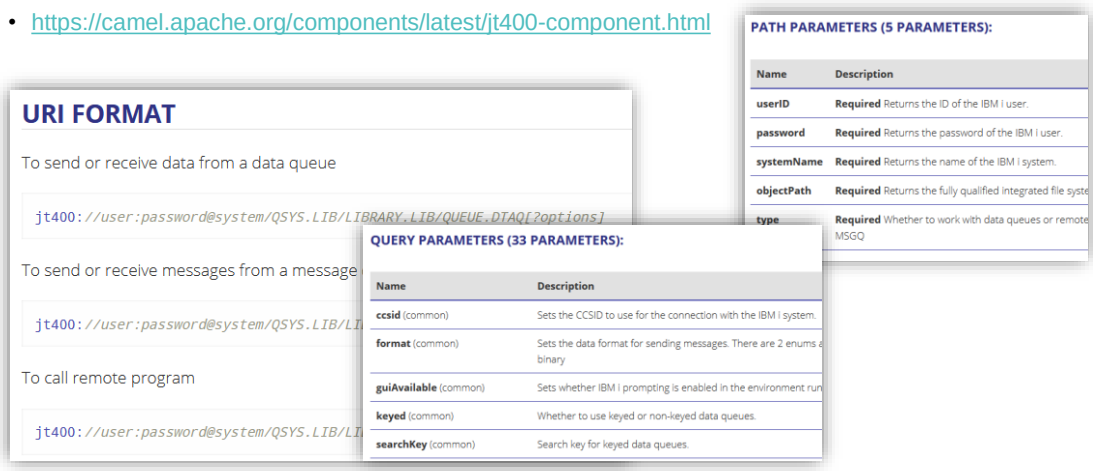
https://www.google.com/search?q=surgery+on+a+grape

Query Parameter
(name is "q", value is "surgery on a grape")

39

Apache Camel Component Documentation

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



40

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);
    }
});
```


41

Message Queue to Email Bridge

- This sample code uses a helper object to generate the URI (parsed from a config file)

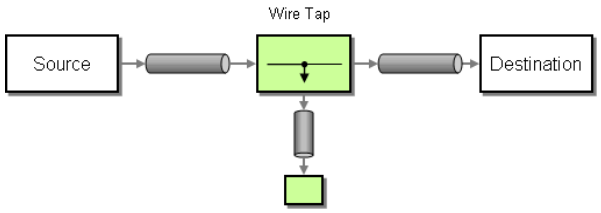
- Introduced the "Wire Tap" EIP

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


42

Wire Tap EIP

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



43

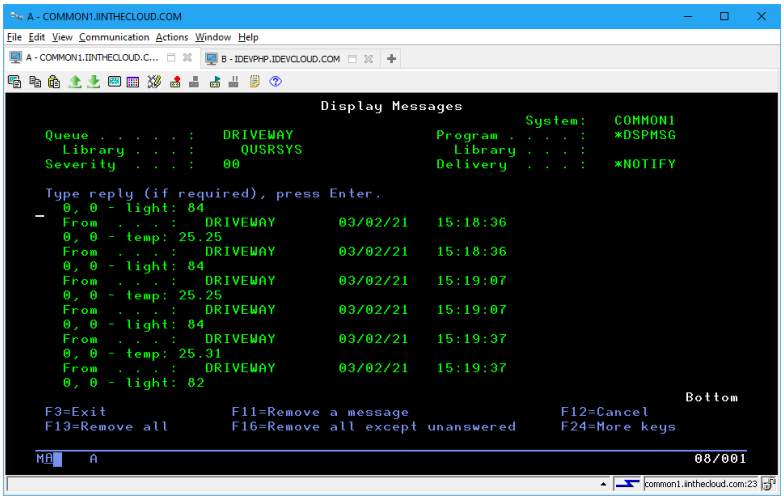
Data Queue to Kafka Bridge

- 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);
    }
});
```


44

Consume IoT Data? No Problem!



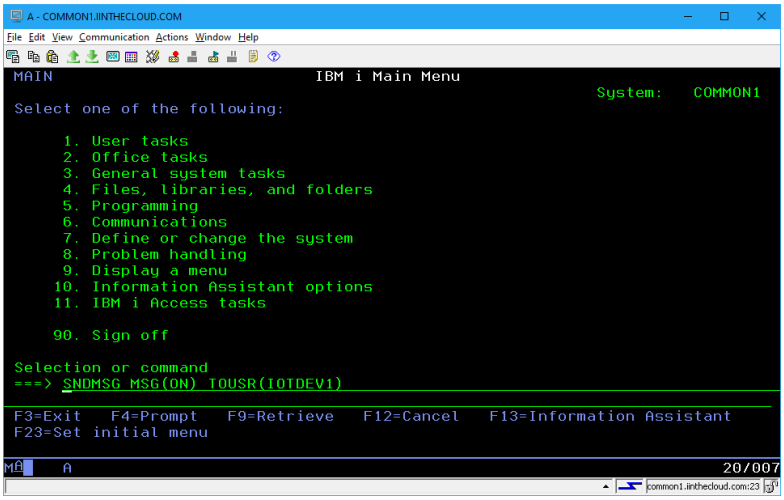
45

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");
    }
});
```


46

Control IoT Devices? No Problem!



47

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

48

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

49

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

50

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");
    }
});
```


51

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")
```


52

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"))
```


53

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")
```


54

Disk Monitor Example

Translate the results to a simple number (Message Translator 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) -> {
        Object diskUsageValue = ((ArrayList<HashMap<String, Object>>)exchange.getIn().getE
        exchange.getIn().setBody(diskUsageValue);
    })
```


55

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()
```


56

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
```


57

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!!")
```


58

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); })
```


59

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
- Also check out Kameleon: https://kameleon.dev
- 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
- How do I install Camel?
  - When doing a standalone Java route, just let Maven download Camel as a dependency!

60

Deploying Your First Simple Routes

- 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"
  - Message Queue to Slack bridge (listens on the *SYSOPR message queue and posts a message to slack when the severity of the message is greater than 45)
    - Requires Slack
  - db2_bash: allows you to call PASE commands in bash via an SQL table function.
    - Can be made to work "out of the box"

61

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

62

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

63

IBM

Kafka vs Camel

	Camel	Kafka
What is it?	A framework	A high-performing message bus
Interfaces with	Just about anything, including IBM i	Anything that knows Kafka, including Apache Camel
Distributed?	No	Yes
Protocol	Many	Kafka

64

65

IBM

Can this perform well?

- https://techchannel.com/SMB/09/2021/power10-ai-event-streaming
- Power 10 demo doing real-time stock price prediction (stock exchange running on IBM i)

66

HTTP Functions

67

New HTTP Functions for SQL (QSYS2)

- HTTP_GET, HTTP_POST, HTTP_PUT, HTTP_DELETE
- Two part blog series:
  - https://techchannel.com/SMB/09/2021/new-sql-http-functions-part-1
    - Overview of the new services
  - https://techchannel.com/Trends/09/2021/sql-http-part-2
    - Integrating with OSS

```
values QSYS2.HTTP_POST(
    'https://api.twilio.com/2010-04-01/Accounts/AC8f1fd0e6d288b6252f7d664e2cfb40a2/Messages.json',
    cast('To=15075559963' concat
    '&From=15556697987' concat
    '&Body=Order is ready' as varchar(855)),
    '{"basicAuth":"AC8f1fd0e6d288b6252f7d664e2cfb40a2,2a6f5a798b2803be2042ab4825d02570",
     "header":"content-type,application/x-www-form-urlencoded"}'
  );
```


68

Function	Description
HTTP_DELETE	Make an HTTP DELETE request and return CLOB data.
HTTP_DELETE_VERBOSE	Make an HTTP DELETE request and return CLOB data and response header.
HTTP_GET	Make an HTTP GET request and return CLOB data.
HTTP_GET_VERBOSE	Make an HTTP GET request and return CLOB data and response header.
HTTP_POST	Make an HTTP POST request and return CLOB data.
HTTP_POST_VERBOSE	Make an HTTP POST request and return CLOB data and response header.
HTTP_PUT	Make an HTTP PUT request and return CLOB data.
HTTP_PUT_VERBOSE	Make an HTTP PUT request and return CLOB data and response header.

Function	Description
BASE64_DECODE	Returns a bit data string that has been Base64 decoded.
BASE64_ENCODE	Returns the Base64 encoded version of a character string.
URL_DECODE	Decodes a URL encoded string.
URL_ENCODE	Encodes a string using URL encoding.

69

Contrasted with SYSTOOLS predecessors

```
select *
  from json_table(
       SYSTOOLS.HTTPGETCLOB('https://official-joke-api.appspot.com/jokes/ten
       'lax $'
       columns(
         "id" integer,
         "type" varchar(100),
         "setup" varchar(100),
         "punchline" varchar(100)
       )
    ) x;
-- Statement ran successfully   (6,693 ms  =  6.693 sec)

--
-- Count all threads in my job
--
select count(distinct(THREAD_ID))
  from table (
       QSYS2.STACK_INFO('*', 'ALL')
    ) as x;
-- ======================================================
-- 49 threads
-- ======================================================
```

```
select *
  from json_table(
       QSYS2.HTTP_GET('https://official-joke-api.appspot.com/j
       'lax $'
       columns(
         "id" integer,
         "type" varchar(100),
         "setup" varchar(100),
         "punchline" varchar(100)
       )
    ) x;
-- Statement ran successfully   (2,539 ms  =  2.539 sec)

--
-- Count all threads in my job
--
select count(distinct(THREAD_ID))
  from table (
       QSYS2.STACK_INFO('*', 'ALL')
    ) as x;
-- ======================================================
-- 1 thread
-- ======================================================
```


70

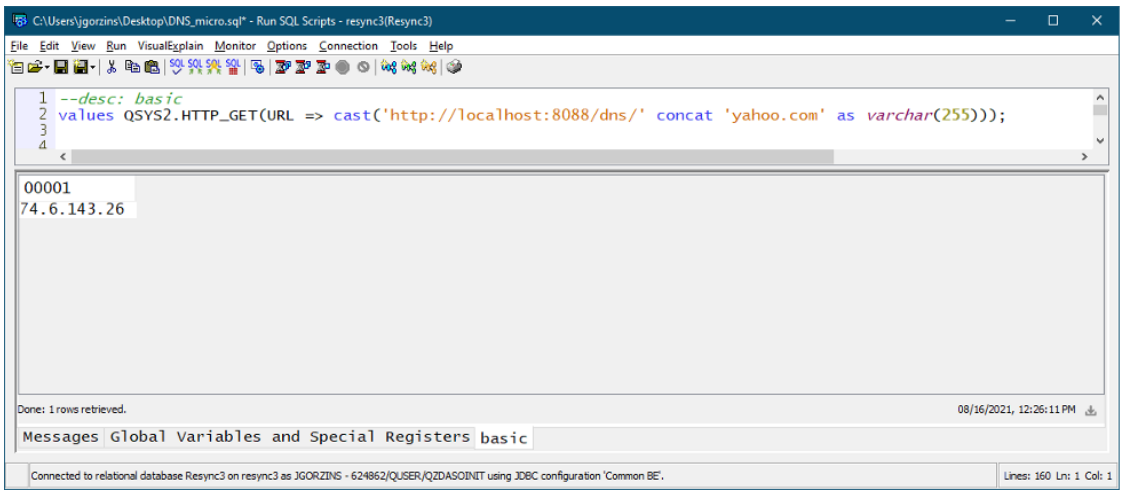
Another example: DNS Lookup

- Discussed in more detail in blog entry referenced earlier
- Simple DNS lookup can be done in a Node.js HTTP API:

```
fastify.get('/dns/:name', (request, reply) => {
  let name = request.params.name;
  dns.lookup(name, (err, address) => {
    if (err || address === undefined) {
      reply.code(404).header('Content-Type', 'application/text; charset=utf-8').send(err);
    } else {
      reply.code(200).header('Content-Type', 'application/text; charset=utf-8').send(address);
    }
  });
})
```

71

Simple DNS



72

IBM

Advanced DNS sample

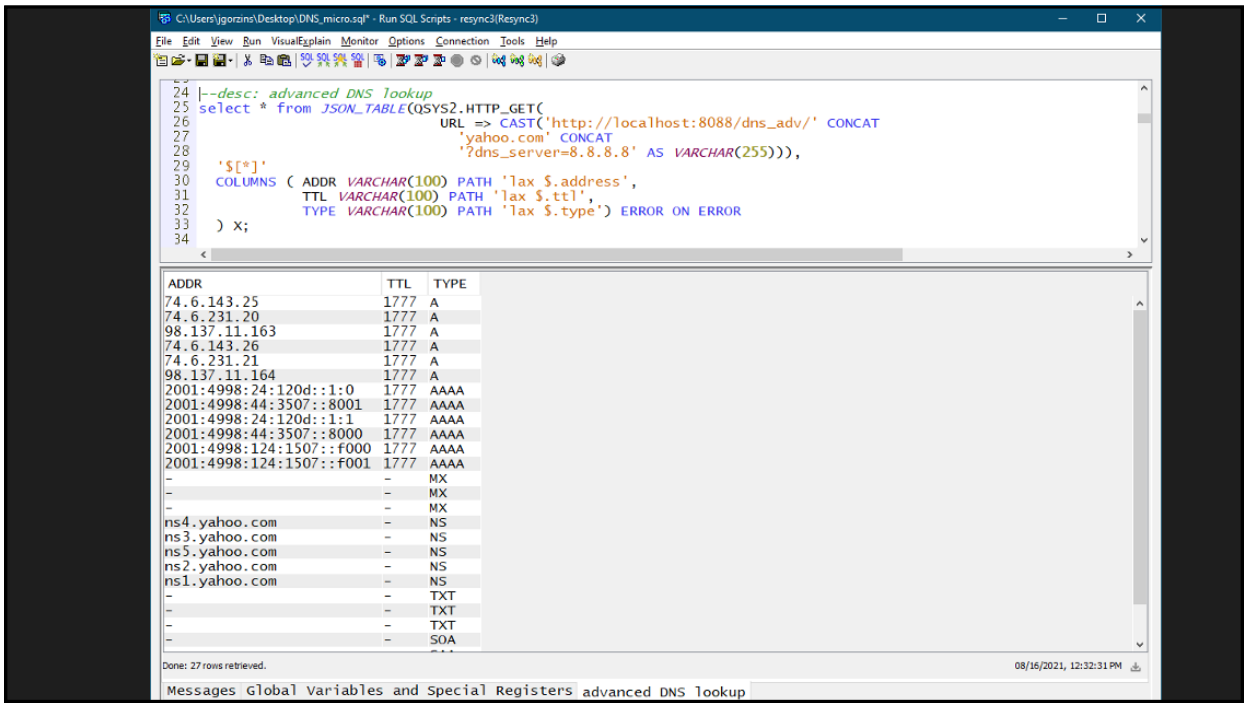
- Retrieve detailed DNS records
- Allow DNS server to be specified

73

IBM

```
fastify.get('/dns_adv/:name', (request, reply) => {
  let reso = new Resolver();
  let name = request.params.name;
  let dns_server = request.query.dns_server;
  if (dns_server === undefined || dns_server === null) {
    reso.setServers(['8.8.8.8']);
  } else {
    reso.setServers([dns_server]);
  }
  reso.resolveAny(name, (err, addresses) => {
    if (err || addresses === undefined) {
      reply.code(404).header('Content-Type', 'application/text; charset=utf-8').send(err);
    } else {
      let i;
      for (i = 0; i < addresses.length; i++) {
        if (addresses[i].address === undefined) {
          addresses[i].address = addresses[i].value;
        }
      }
      reply.code(200).header('Content-Type', 'application/json; charset=utf-8').send(addresses);
    }
  });
})
```


74

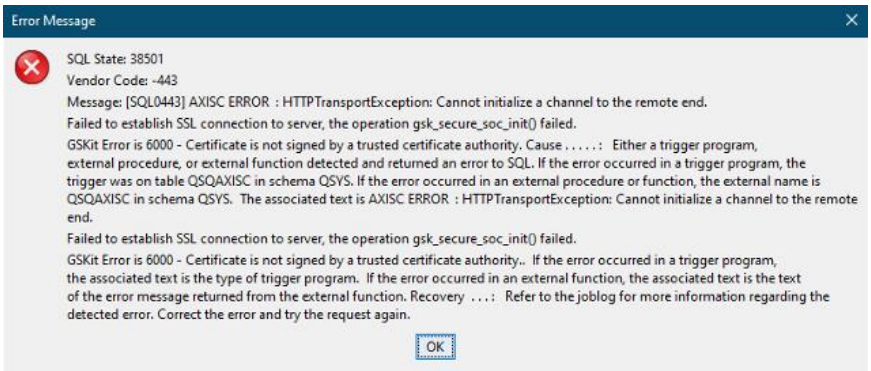


75

IBM

TLS Setup

- If the certificate isn't trusted, you may get an error



76

Trusting the certificate

- Option 1: Import into DCM (or click "Populate CA Certificates" in DCM GUI if it's a common cert)
- Option 2: Use DCM-Tools open source toolset
  https://github.com/ThePrez/DCM-tools/

  To trust the certificate from localhost port 8089:
  `dcmimport --fetch-from=localhost:8089`

  To trust the certificates installed into PASE OSS environment:
  `dcmimport --installed-certs`

77

One last example: publishing to Kafka…

```
VALUES QSYS2.HTTP_POST('http://myrestserver:8082/topics/fromrest',
                       '{"records":[{"key":"my_key","value":"Just some data to stream to Kafka"}]}',
                       '{"header":"Content-Type,application/vnd.kafka.json.v2+json" ,
                         "header":"Accept,application/vnd.kafka.v2+json"
                         }');
```


78

IBM

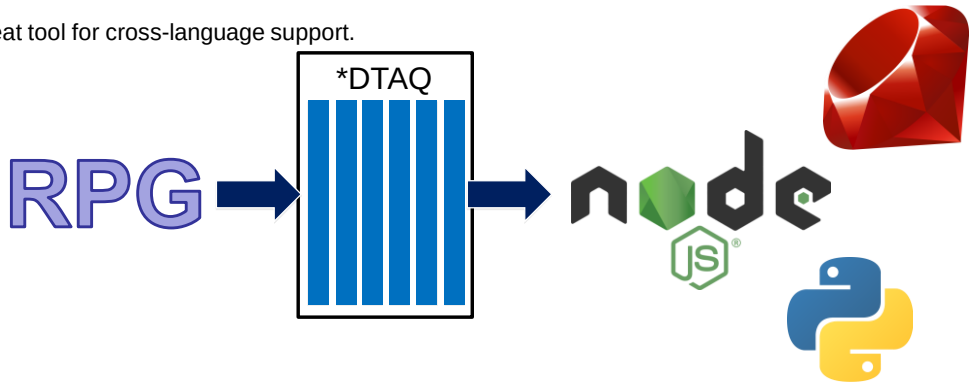
Etcetera

79

IBM

Data Queues

- Data Queues are great tools for general IPC.
- Also a great tool for cross-language support.



80

IBM

Data queues

RPG could form JSON that looks like this:

```
{"body":"Hello World", "to":"+123456789", "from":"+123456789"}
```

Sends it like this:

```
dcl-pr snddtaq      extpgm('QSNDDTAQ');
   ##dtqname   like(d#dtqname)   const;
   ##dtqlib    like(d#dtqlib)    const;
   ##dtqlength like(d#dtqlength) const;
   ##dtqdata   char(32766)       const options(*varsize);
end-pr;
```

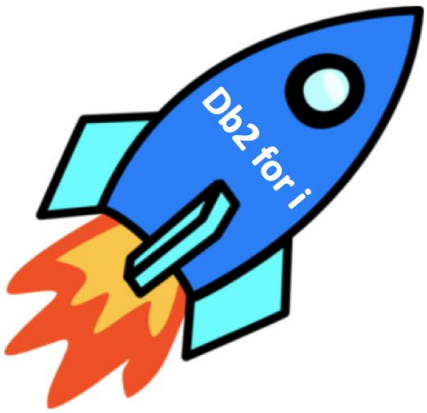

81

```
async function getFromQ() {
  try {
    while(true) {
      let results =
        await pool.prepareExecute(`SELECT MESSAGE_DATA_UTF8
                                   FROM TABLE(QSYS2.RECEIVE_DATA_QUEUE(
                                     DATA_QUEUE => 'SNDSMSQ',
                                     WAIT_TIME => 1,
                                     DATA_QUEUE_LIBRARY => 'JESSEG'))`);
      if(results != undefined) {
        let data = JSON.parse(results.resultSet[0].MESSAGE_DATA_UTF8);
        sendSMS(data.to, data.from, data.body);
      }
    }
  }catch(err) {
      console.log('error: '+err.stack);
  }
}
```

82

Connectivity libraries

- All major languages come with IBM i integration!!
- OSS programs can interact with:
  - Db2
  - Data Queues
  - CL commands
  - Data Areas
  - RPG Programs
  - QSH scripts
  - … and pretty much anything else!!



83

UNIXCMD

- https://www.scottklement.com/unixcmd/
- https://www.common.org/blogs/temporary-admin1/2018/01/22/a-powerful-way-to-run-unix-and-open-source-tools-f
- This is a tool to help simplify the process of running QShell or PASE commands from RPG or CL.
- Implements a SPECIAL file
- When you read the file, you're reading the output of the command.
- When you write to the file, you're writing data to the command. For CL, it provides OPNPIPE, RCVPIPE, SNDPIPE and CLOPIPE commands that open the Unix and send/receive data from it, very similar to the way you use files in CL.

84

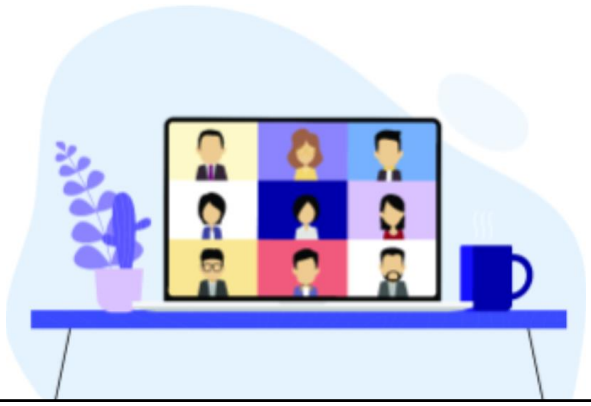
```
**free

dcl-f UNIX disk(1000) usage(*input:*output) handler('UNIXCMDOA': cmd) usropn;
dcl-s cmd char(5000);
dcl-s lat packed(11: 7);
dcl-s lon packed(11: 7);
dcl-ds record len(1000) end-ds;

cmd = 'PATH=$PATH:/usr/local/ZendSvr6/bin && +
       iconv -f 0 -t 819 | +
       php-cli /www/zendsvr/htdocs/geocode.php';

open UNIX;

record = '1600 W Pennsylvania Av, Washington DC';
write UNIX record;

read UNIX record;
lat = %dec(record: 11: 7);

read UNIX record;
lon = %dec(record: 11: 7);

close UNIX;

dsply ('lat=' + %char(lat));
dsply ('lon=' + %char(lon));
*inlr = *on;
```

85

IBM

What else?

- More ways than I can list!!
- Today's presentation are just a few of Jesse's favorites
- Collaboration and open-mindedness is key (the real "skills gap"?)
- With open source, you're never alone



86

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

87

Resources

- IBM i RPMs (RedHat Technology we use for building/distributing)
  - http://ibm.biz/ibmi-rpms
- IBM i Open Source Support
  - http://ibm.biz/ibmi-oss-support
- Jesse Gorzinski's blog
  - http://ibm.biz/open-your-i
  - https://ibmsystemsmag.com/Power-Systems/06/2020/common-open-source-questions-answered
- Open Source Examples
  - http://github.com/IBM/ibmi-oss-examples
- IBM i customer stories
  - http://ibm.biz/ibmistories
- Community chat
  - http://ibm.biz/ibmioss-chat (join at http://ibm.biz/ibmioss-chat-join)
- Jesse
  - jgorzins@us.ibm.com
  - http://twitter.com/IBMJesseG

Brand new landing page:
http://ibm.biz/ibmioss

88

Notices and disclaimers

U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed "as is" without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.** IBM products and services are warranted per the terms and conditions of the agreements under which they are provided.

IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those

customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

89

Notices and disclaimers

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products about this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, ibm.com and [names of other referenced IBM products and services used in the presentation] are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml

90