

QUSER discussion:

Connect RPG and Db2 to the World

Presented by:

Jesse Gorzinski

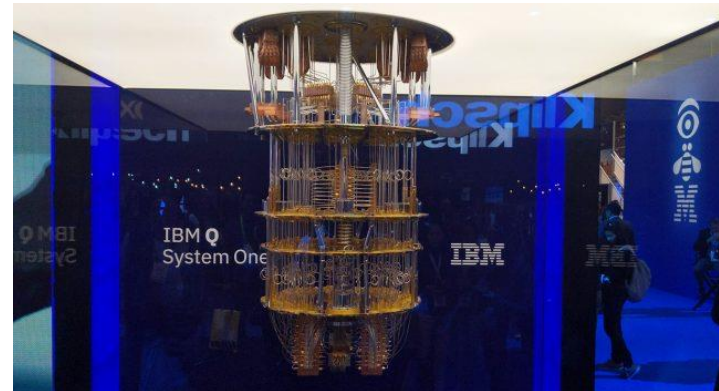
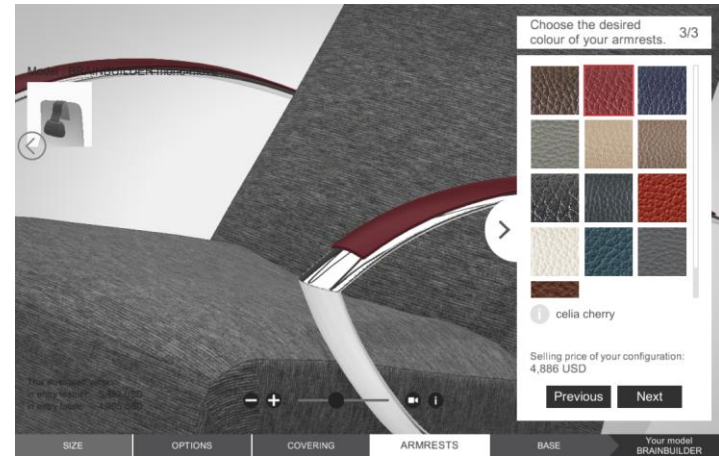
jgorzins@us.ibm.com

Twitter: @IBMJesseG




IBM i: Very different from 5 years ago!

- Stuff we've seen IBM i do with OSS:
 - Artificial intelligence
 - IoT
 - 3D visual rendering
 - REST & microservices
 - Interacting with social media
 - CI/CD pipelines
 - Interacting with Amazon, eBay, etc.
 - Modern web interfaces with Vue, React, Angular, etc
 - Quantum computing
 - Integration with video feeds
 - Report generation (PDF, Excel, etc)
 - git source control



Agenda

- 
- 1 The Hybrid Application
 - 2 High-level Approaches
 - 3 New HTTP Functions for SQL
 - 4 More about Apache Camel

The Hybrid Application

The Hybrid Approach

IBM i

World's Best RDBMS

COBOL+RPG

Lowest cost of ownership
(TCO)

Reliability, securability,
efficiency

Protection of investment



Open Source

Artificial Intelligence

Quantum Computing

Microservices / APIs

DevOps

Internet of Things

Web Technologies

IBM i Customer Stories

For over thirty years, IBM i has supported thousands of customers worldwide and continues to be the backbone of their infrastructure into the future. Read why customers are continuing to choose IBM i as their platform of choice.



<http://ibm.biz/ibmistories>

See innovation at work in the IBM i community

SAIB



Increasing Productivity with IBM i and Alexa for Business

[Read the customer story →](#)

IRIS Financial Services S.A., a Canon Company



Encouraging the Sustainability of Archived Financial Data with IBM i

[Read the customer story →](#)

Brain Staff Consultants



Improving the Student Experience with IBM i and AI

[Read the customer story →](#)

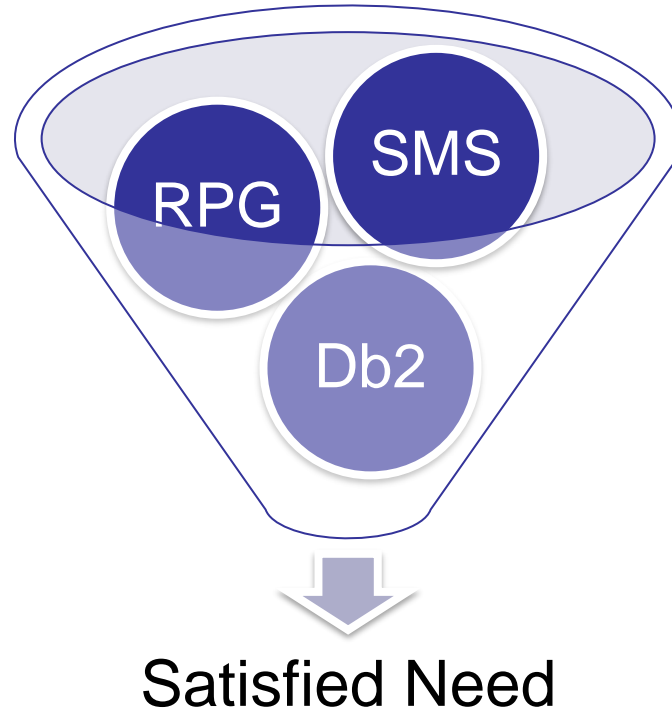
CRT Services



Speeding Up Reimbursement with IBM i

[Read the customer story →](#)

Satisfying Needs



RPG program sending SMS messages? Sure!

- As an example, today's presentation will focus on this use case
- RPG Code and/or Db2 tables
 - Knows customer data
- Open Source (Node.js)
 - Twilio, a third-party service, provides open source libraries
 - **Could be any OSS-solvable problem, any OSS language!**



What else can OSS augmentation provide?

- Could be anything from the millions of OSS solutions
 - Artificial Intelligence
 - IoT
 - Quantum Computing
 - report generation (Excel, PDF, etc)
 - REST/web/microservices
- Limitless possibilities
 - npm (Node.js repository) has over 1,000,000 packages
 - PyPI (Python package index) has over 250,000 packages

Getting Started at Twilio

- Free trial is available for development purposes
- Need to verify a "to" number
 - Personal # for development
- Need to also get a "Twilio number"
 - This will be the "from" number

```
var twilio = require('twilio');
```

```
var accountSid = 'AC8f52f7d664e3cf1fd0e6d288b62b40a2';
```

```
var authToken = '8f4fe0b24764e16f76cc9edbaebc87d4';
```

```
var client = new twilio(accountSid, authToken);
```

```
function sendSMS(to, from, body) {
  client.messages.create({
    body: body,    // SMS body
    to:   to,     // SMS to number
    from: from    // SMS from a valid Twilio number
  }, function(err, message) {
    if (err) {
      console.error(err.message);
    }
  });
}
```

Integration Techniques

Different approaches (Jesse's personal categorization)

ILE-First

OSS-First


Side-by-Side

ILE-First Approaches

Simple Node.js SMS-sending app...

```
function sendSMSMsg(to, from, body) {
  client.messages.create({
    body: body,      // SMS body
    to: to,          // SMS to number
    from: from       // SMS from a valid Twilio number
  }, function(err, message) {
    if (err) {
      console.error(err.message);
    }
  });
}

var myArgs = process.argv.slice(2);
sendSMSMsg(myArgs[0], myArgs[1], myArgs[2]);
```



Simple shell invocation

- QSH CMD('cd /home/JGORZINS/projects/oss-integration-pptx/ibmi-oss-examples/nodejs/Send-SMS-with-Twilio-from-IBM-i && /QOpenSys/pkg/bin/node sendSMSDataFromQueue.js +15071111111 +13231111111 "Hello there from Barbara"')

```
**free
// Don't call this program directly
// Use the SENDSMS command
dcl-pi *n;
    parm1 varchar(20);
    parm2 varchar(20);
    parm3 varchar(5000);
end-pi;

dcl-c QT '''; // single quote
dcl-c NODEJS_DIR '/home/JGORZINS/projects/oss-integration-pptx+
| | | | | /ibmi-oss-examples/nodejs/';
dcl-c TWILIO_DIR 'Send-SMS-with-Twilio-from-IBM-i';
dcl-c NODE_CMD '/QOpenSys/pkgs/bin/node';
dcl-c JS_SEND_DATA_FROM_QUEUE 'sendSMSDataFromQueue.js';
dcl-s cmd varchar(6000);
dcl-s qsh_cmd varchar(6100);
```

```
dcl-pr qcmdexc extpgm;  
  cmd char(6100) const;  
  cmdlen packed(15:5) const;  
end-pr;
```

```
cmd = 'cd ' + NODEJS_DIR + TWILIO_DIR // CD to the TWILIO directory  
  + ' && ' + NODE_CMD + ' ' // Run the node command  
  + JS_SEND_DATA_FROM_QUEUE // JS script to run  
  + ' "' + parm1 + '" '  
  + ' "' + parm2 + '" '  
  + ' "' + parm3 + '" ' ;  
qsh_cmd = 'QSH CMD(' + QT + cmd + QT + ')';
```

```
QCMDEXC (qsh_cmd : %len(qsh_cmd));  
return;
```

Can wrap it all with a CL command

```
JESSEG/SENDSMS DESTNUM(+15071111111)
TWILIONUM(+13231111111)
MSG('Hello from Barbara!')
```

```
B - OSS72DEV.RCH.STGLABS.IBM.COM
File Edit View Communication Actions Window Help
Send a message with SMS (SENDSMS)
Type choices, press Enter.
Text number, eg +1234567890 . . . > +15071111111
Twilio number, eg +1234567890 . . . > +13231111111
Parm 3 . . . . . > 'Hello from Barbara!'
Bottom
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys
M&B B 05/037
```

QSHONI

- <https://github.com/richardschoen/QshOni>

The following example calls the ls command to list files for the /tmp directory:

```
QSHEXEC CMDLINE('cd /tmp;ls')
DSPSTDOUT(*YES)
LOGSTDOUT(*NO)
PRTSTDOUT(*NO)
DLTSTDOUT(*YES)
IFSSTDOUT(*NO)
IFSFILE('/tmp/log.txt)
IFSOPT(*REPLACE)
PRTSPLF(QSHEXECLOG)
PRTUSRDTA(*NONE)
PRTTXT(*NONE)
```



Use when...

- Only integration needed is a simple task dispatch
- Output from OSS side doesn't need to be handled by RPG
 - You can do it but it's klunky
- Looking for the simplest solution
- OSS-first approaches aren't sensible



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.

```
**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;
```


Use when...

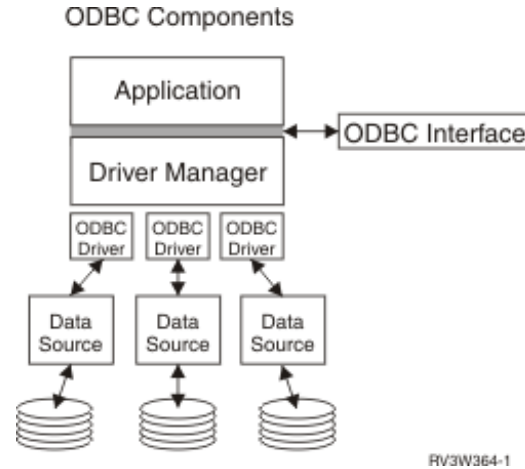
- Standard in/out is a reasonable way to talk to PASE stuff
 - Pretty easy refactor
- Coding RPG
- Want synchronous behavior and tighter integration



OSS-First Approaches

ODBC Driver

- IBM i Access ODBC driver now supported on:
 - Windows
 - Linux
 - IBM i

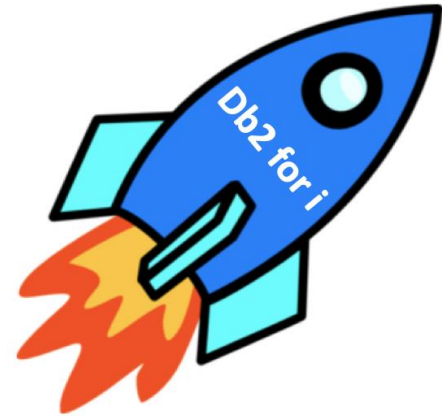


RV3W364-1

- Why ODBC on IBM i?
 1. Develop on Windows/Linux, deploy on IBM i
 2. Use standard ODBC libraries from various languages (PHP, Python, Node.js, Ruby, R, etc)

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!!



Toolkits

- Toolkits provide the ability to integrate with various features/functions/programs of IBM I
- Node.js toolkit:
 - <https://github.com/IBM/nodejs-itookit/>
- PHP Toolkit for IBM i:
 - <http://yips.idevcloud.com/wiki/index.php/XMLSERVICE/PHP>
 - <https://github.com/zendtech/lbmiToolkit/>
- Python itoolkit
 - <https://github.com/IBM/python-itookit/>
- Ruby itoolkit
 - <https://bitbucket.org/litmis/ruby-itookit>
- Swift
 - <https://bitbucket.org/litmis/swift-itookit>
- .NET
 - <https://github.com/richardschoen/lbmiXmlserviceStd>

NOTE: Documentation for Ruby, Swift, and .NET is not sufficient to complete the capability information provided on the following slides

```
**free
dcl-pi *n;
|   parm likeds(parm_t);
end-pi;
dcl-ds parm_t qualified template;
|   phone_number char(21);
|   msg_body char(120);
end-ds;
parm.phone_number = '+15071111111';
parm.msg_body = 'Hello from RPG!';
return;
```

OSS calling RPG via itoolkit

```
async function getFromPgm() {
  try {
    const conn = new Connection({
      transport: 'idb',
    });
    const receiver = {
      name: 'receiver',
      type: 'ds',
      io: 'out',
      len: 'rec1',
      fields: [
        { name: 'phone_number', type: '21A', value: '' },
        { name: 'msg_body', type: '120A', value: '' },
      ],
    };
    const program = new ProgramCall('GETSMS', { lib: 'JESSEG' });
    program.addParam(receiver);
    conn.add(program);
  }
}
```

```
conn.run((error, xmlOutput) => {
  if (error) {
    throw error;
  }
  parseString(xmlOutput, (parseError, result) => {
    if (parseError) {
      throw parseError;
    }
    let to=result.myscript.pgm[0].parm[0].ds[0].data[0]._;
    let body=result.myscript.pgm[0].parm[0].ds[0].data[1]._;
    sendSMS(to, twilioNum, body);
  });
});
} catch(err) {
  console.log('error: '+err.stack);
}
```


Use when...

- ILE/Db2 code is modularized
 - You have a callable function that returns what OSS side needs
- Want a direct, tight integration between OSS and ILE



OSS getting information via SQL

```
async function getFromSQL() {
  try {
    while(true) {
      let results = await pool.prepareExecute(
        'SELECT PHONE_NUM,MSG_BODY FROM TABLE(JESSEG.GETSMS2())');
      if(results == undefined) {
        console.log("nothing");
        continue;
      }
      for(var i in results.resultSet) {
        let to = results.resultSet[i].PHONE_NUM;
        let body = results.resultSet[i].MSG_BODY;
        sendSMS(to, '+13233753278', body);
      }
    }
  } catch(err) {
    console.log('error: '+err.stack);
  }
}
```

Use when...

- ILE/Db2 code is modularized
- Pursuing data-centric programming
- Want something cleaner (less entanglement) than direct program calls
- Looking for language-agnostic solution

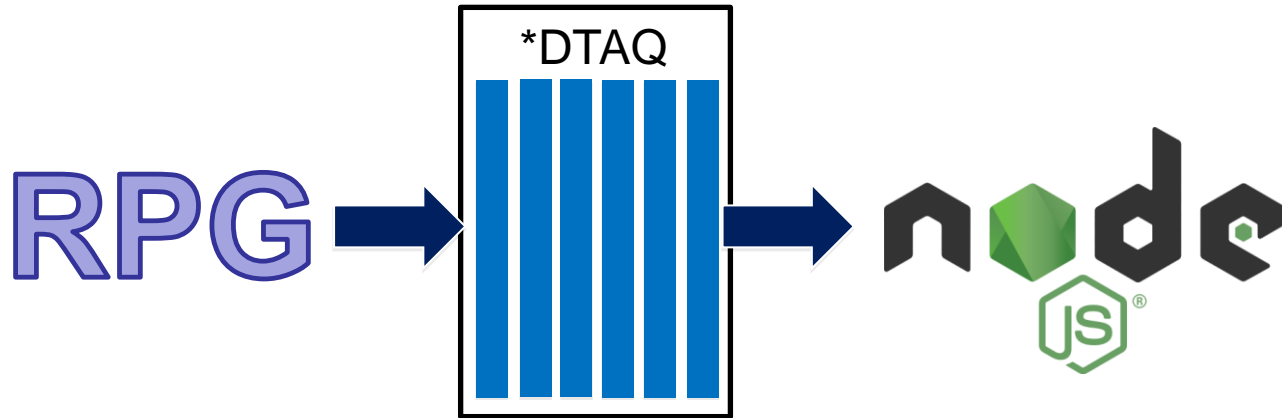


Side-by-Side Approaches

Data Queues

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

Data Queue example



Data queue example code

RPG forms 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;
```

```
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'))`);

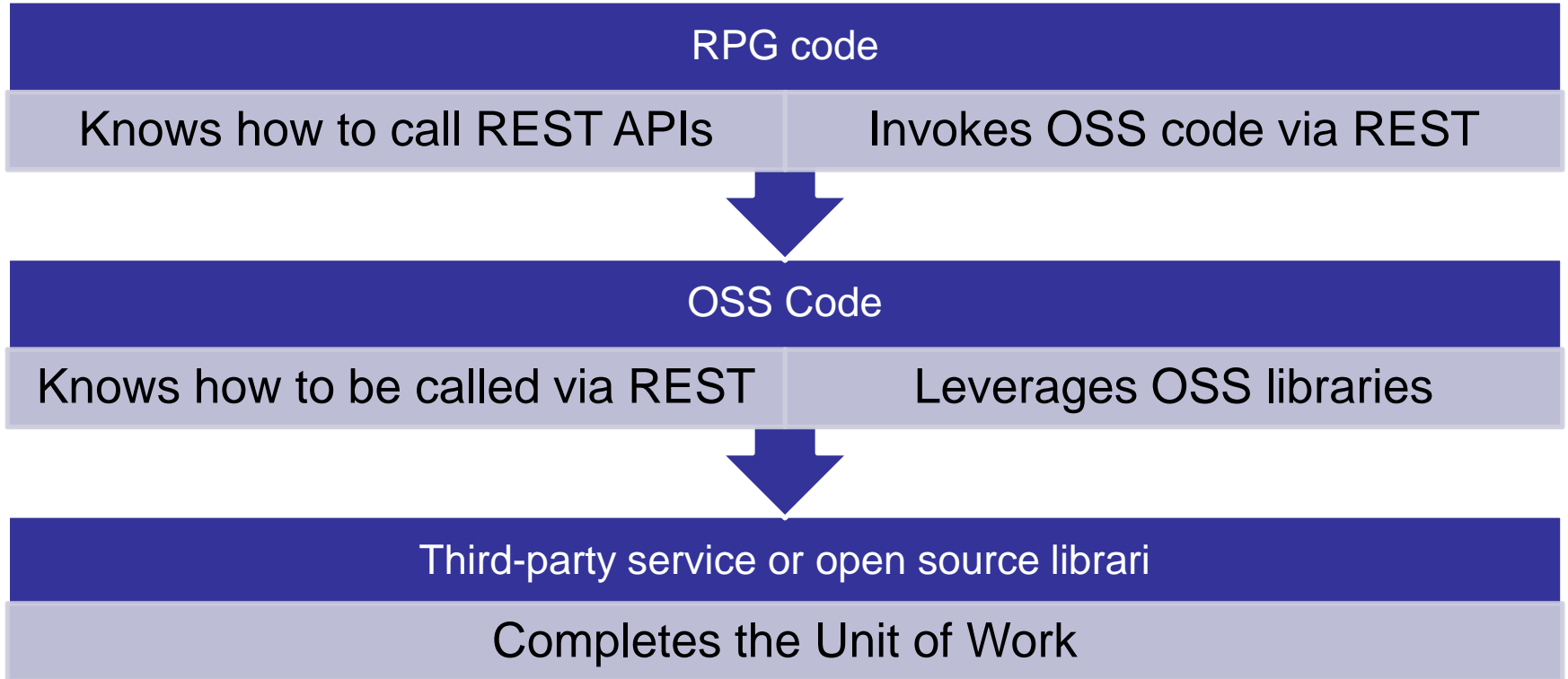
      if(results != undefined) {
        let data = JSON.parse(results.resultSet[0].MESSAGE_DATA_UTF8);
        sendSMS(data.to, data.from, data.body);
      }
    }
  } catch(err) {
    console.log('error: '+err.stack);
  }
}
```


Use when...

- Asynchronous processing is OK
 - Two-way communication can be done, but klunky
- You optionally want OSS jobs to handle requests in batch-like function
- It's easy to wire in data queues with existing logic



REST API call to OSS



In this example:

1. We expose a rest API in Node.js
 - `http://localhost/send/<phone_number>`
 - HTTP method POST
 - One parameter, "body"
2. We call that API from RPG
3. Node.js performs the task and returns the result

```
// Require the fastify framework and instantiate it
const fastify = require('fastify')({ logger: true })
const url = require('url');
```

```
// Declare a route
fastify.post('/send/:to', async (request, reply) => {
  const { params: { to } } = request;
  const queryObject = url.parse(request.raw.url, true).query;
  let msgbody = queryObject.body;
  await client.messages.create({
    body: msgbody,    // SMS body
    to: to,          // SMS to number
    from: '+13233753278' // SMS from a valid Twilio number
  }, function(err, message) {
    if (err) {
      console.error(err.message);
      reply.code(500)
        .header('Content-Type', 'application/json; charset=utf-8')
        .send({ error: err.message });
    } else {
      reply.code(200)
        .header('Content-Type', 'application/json; charset=utf-8')
        .send({ message: message });
    }
  });
})
```

```
// Run the server!
const start = async (port) => {
  try {
    await fastify.listen(port, '127.0.0.1');
    fastify.log.info(`server listening on ${fastify.server.address().port}`)
  } catch (err) {
    fastify.log.error(err)
    process.exit(1)
  }
}
```

Use when...

- You need a synchronous, function-call-like behavior
- You want to process return values in RPG/ILE
- You want loose coupling and language-agnosticism
- Looking for the most robust and industry-standard technique



Running OSS-y jobs 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
- TIP: For POJO camel applications, submit to batch with a custom job name

Using Service Commander for Kafka server

- Open source utility for starting/stopping/managing services(jobs) on IBM i

```
-bash-4.4$ sc start kafkaviz
Performing operation 'START' on service 'kafkaviz'
Attempting to start service dependency 'kafka' (Apache Kafka bootstrap server)...
Attempting to start service dependency 'zookeeper' (Apache Zookeeper Server using OpenJDK)...
Service 'Apache Zookeeper Server using OpenJDK' successfully started
For details, see log file at: /home/JGORZINS/.sc/logs/2021-03-12T12:46:29.320-0500.zookeeper.log
Service 'Apache Kafka bootstrap server' successfully started
For details, see log file at: /home/JGORZINS/.sc/logs/2021-03-12T12:46:29.319-0500.kafka.log
Service 'Apache Kafka Visualizer' successfully started
For details, see log file at: /home/JGORZINS/.sc/logs/2021-03-12T12:46:29.312-0500.kafkaviz.log
```



<https://github.com/ThePrez/ServiceCommander-IBMi>

New HTTP Functions

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"}'  
);
```

| 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. |

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
    '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
```

Why use OSS to accomplish a task, then?

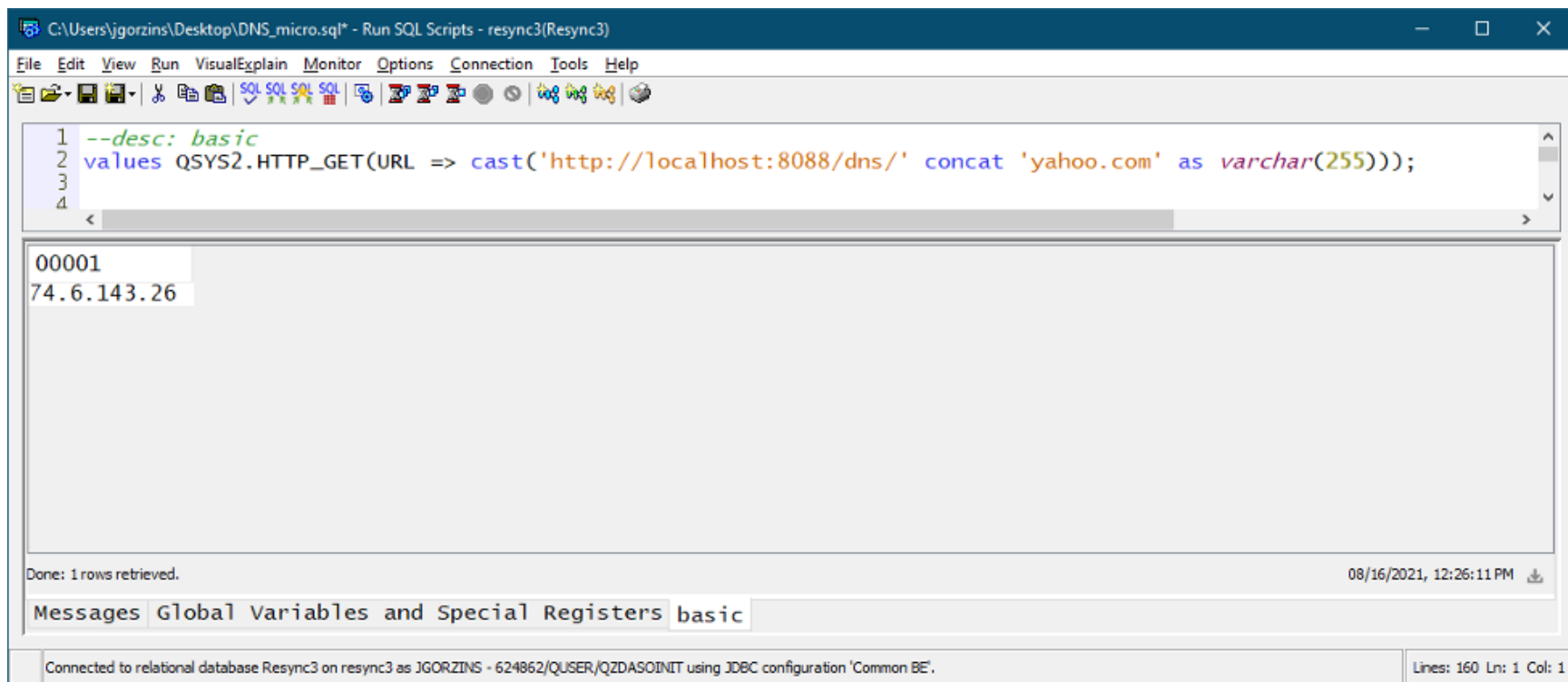
- Sending SMS with Twilio is an atypical example
 - REST APIs only available for certain services, made easy with OSS libraries
 - Requirements might involve a number of things that really can't (or shouldn't!) be done in RPG
 - Might have a number of tasks that make sense to chain into a single OSS-language function
- APIs can change
 - Easier to update the OSS library than to update your code
 - Twilio (and other smart ones) use date-based versioning, so not an issue
- Might be easier to secure communication
 - DCM vs. open source
- You might already have an OSS component of your application

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);
    }
  });
})
```

Simple DNS



The screenshot shows a SQL client window titled "C:\Users\jgorzins\Desktop\DNS_micro.sql* - Run SQL Scripts - resync3(Resync3)". The window has a menu bar with "File", "Edit", "View", "Run", "Visual Explain", "Monitor", "Options", "Connection", "Tools", and "Help". Below the menu bar is a toolbar with various icons. The main area contains a SQL script editor with the following code:

```
1 --desc: basic
2 values QSYS2.HTTP_GET(URL => cast('http://localhost:8088/dns/' concat 'yahoo.com' as varchar(255)));
3
4
```

The execution result is displayed in a table with one row:

| |
|-------------|
| 00001 |
| 74.6.143.26 |

Below the table, it says "Done: 1 rows retrieved." and "08/16/2021, 12:26:11 PM". At the bottom, there is a status bar with "Messages Global Variables and Special Registers basic" and "Connected to relational database Resync3 on resync3 as JGORZINS - 624862/QUSER/QZDASOINIT using JDBC configuration 'Common BE'". The bottom right corner shows "Lines: 160 Ln: 1 Col: 1".

Advanced DNS sample

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

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



```

24 |--desc: advanced DNS lookup
25 |select * from JSON_TABLE(QSYS2.HTTP_GET(
26 |           URL => CAST('http://localhost:8088/dns_adv/' CONCAT
27 |           'yahoo.com' CONCAT
28 |           '?dns_server=8.8.8.8' AS VARCHAR(255))),
29 | '$[*]'
30 | COLUMNS ( ADDR VARCHAR(100) PATH 'lax $.address',
31 |           TTL VARCHAR(100) PATH 'lax $.ttl',
32 |           TYPE VARCHAR(100) PATH 'lax $.type') ERROR ON ERROR
33 | ) x;
34 |

```

| ADDR | TTL | TYPE |
|--------------------------|------|------|
| 74.6.143.25 | 1777 | A |
| 74.6.231.20 | 1777 | A |
| 98.137.11.163 | 1777 | A |
| 74.6.143.26 | 1777 | A |
| 74.6.231.21 | 1777 | A |
| 98.137.11.164 | 1777 | A |
| 2001:4998:24:120d::1:0 | 1777 | AAAA |
| 2001:4998:44:3507::8001 | 1777 | AAAA |
| 2001:4998:24:120d::1:1 | 1777 | AAAA |
| 2001:4998:44:3507::8000 | 1777 | AAAA |
| 2001:4998:124:1507::f000 | 1777 | AAAA |
| 2001:4998:124:1507::f001 | 1777 | AAAA |
| - | - | MX |
| - | - | MX |
| - | - | MX |
| ns4.yahoo.com | - | NS |
| ns3.yahoo.com | - | NS |
| ns5.yahoo.com | - | NS |
| ns2.yahoo.com | - | NS |
| ns1.yahoo.com | - | NS |
| - | - | TXT |
| - | - | TXT |
| - | - | TXT |
| - | - | SOA |

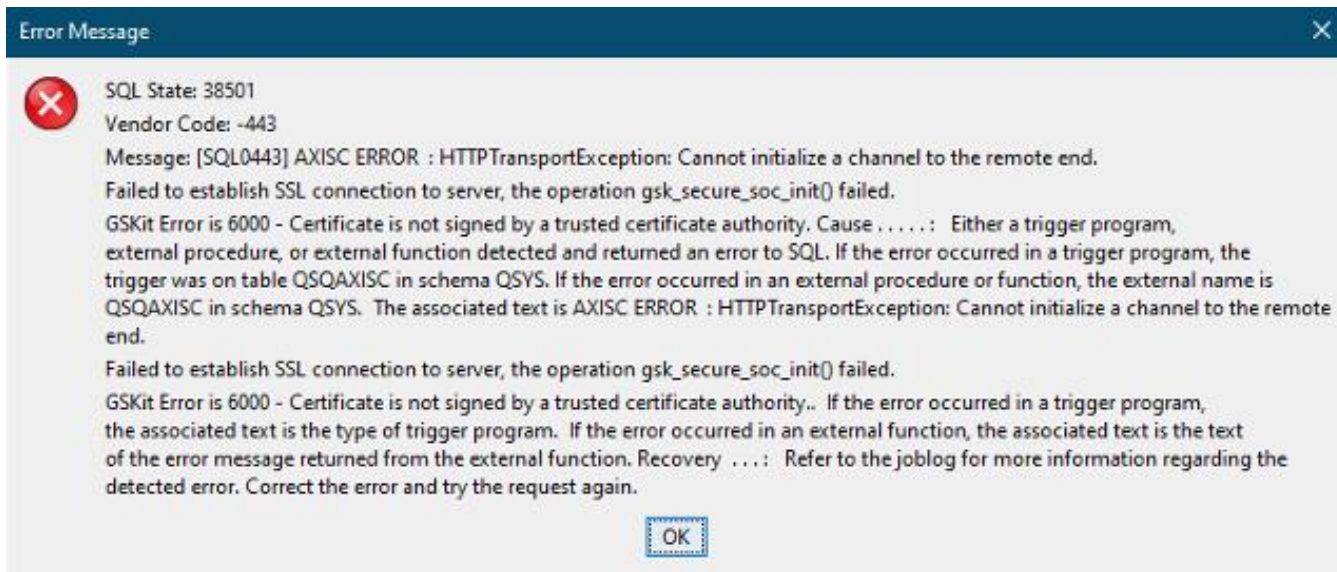
Done: 27 rows retrieved.

08/16/2021, 12:32:31 PM

Messages Global variables and Special Registers advanced DNS lookup

TLS Setup

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



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
```

A few more thoughts on Apache Camel

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>



Apache Camel JT400 component

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, indicating that the message queue functionality is a new feature.

```
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&...`

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

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



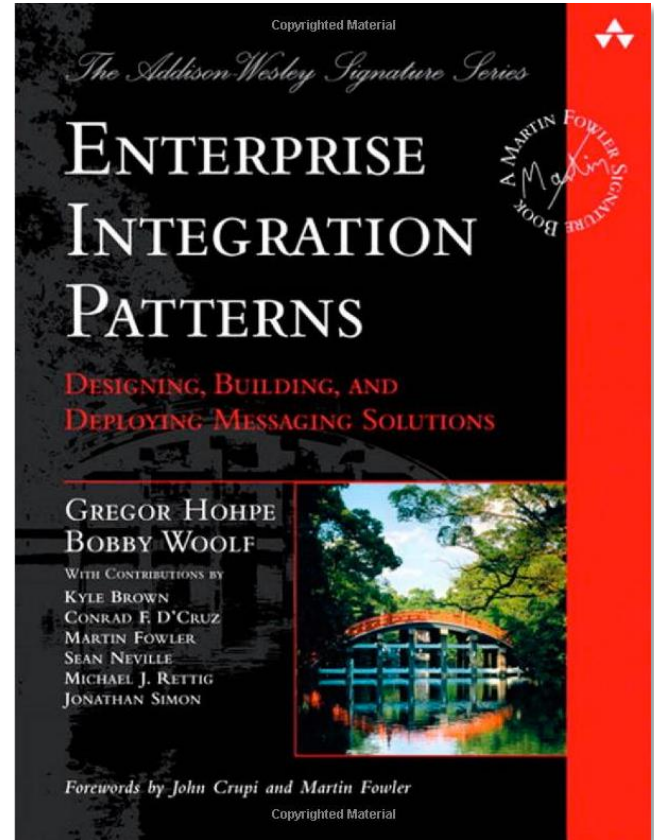
What's Camel?

- Camel is an open source MOM language developed by Apache and based on Enterprise Integration Patterns
- Centers around 60+ common patterns found in Enterprise integration projects
- Defines these common patterns in a vendor neutral way
- Camel aims to provide a language which implements these patterns, in a style modeled after the UNIX pipeline

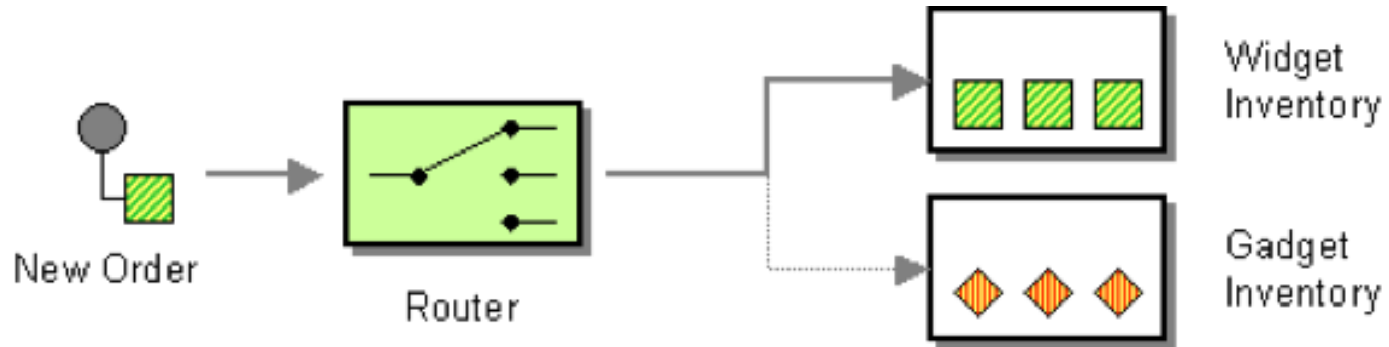


APACHE®

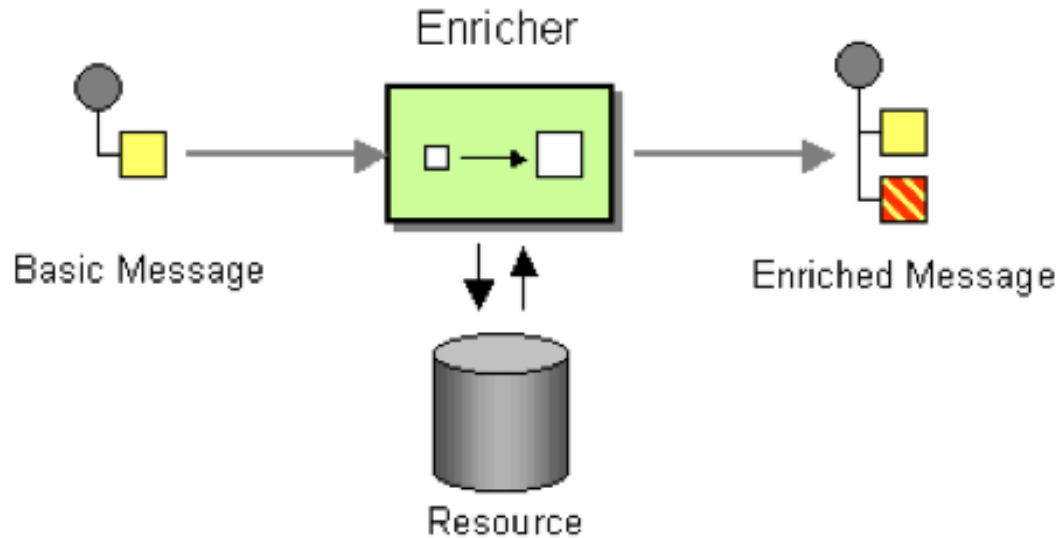
Camel



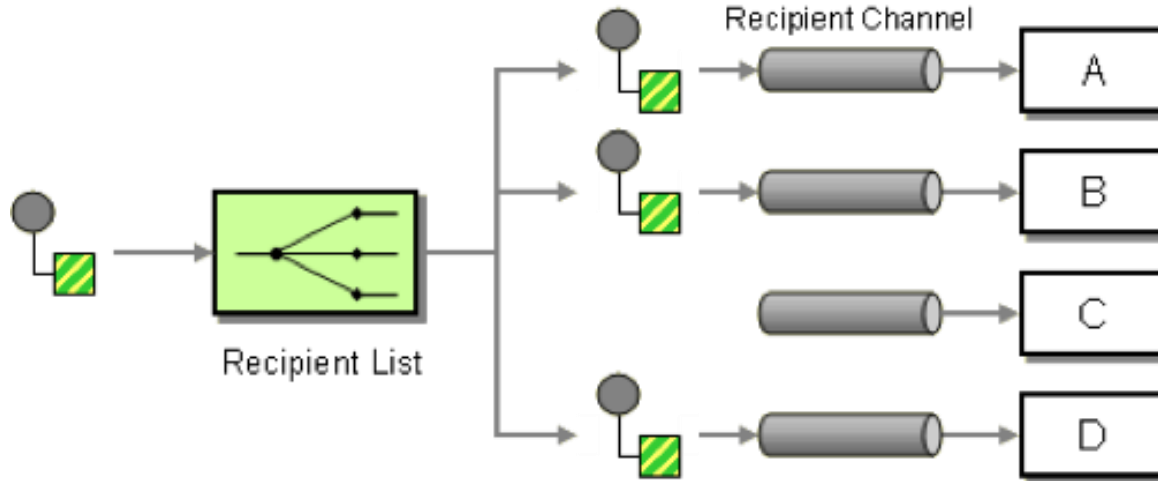
Integration Patterns – Content Based Router



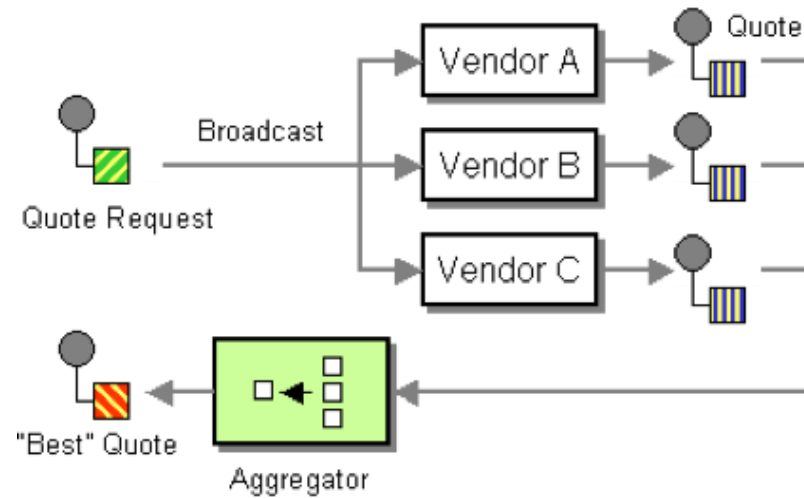
Integration Patterns – Content Enricher



Integration Patterns – Recipient List



Integration Patterns – Scatter-Gather



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)

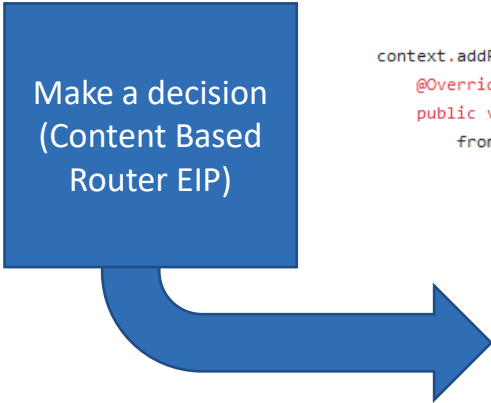
```

process((exchange) -> {
    Object diskUsageValue = ((ArrayList<HashMap<String, Object>>)exchange.getIn().getB
    exchange.getIn().setBody(diskUsageValue);
})

```

Disk Monitor Example

Make a decision
(Content Based
Router EIP)

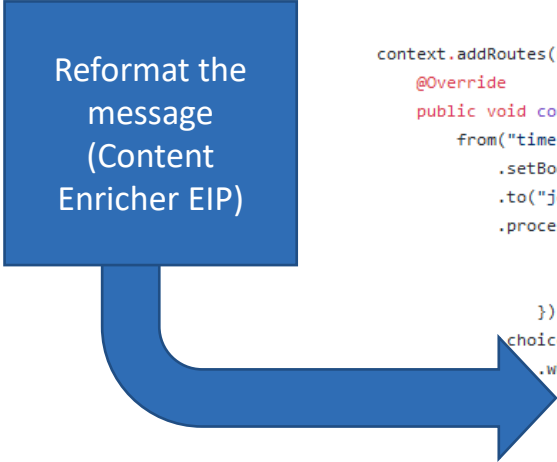


```
context.addRoutes(new RouteBuilder() {
    @Override
    public void configure() {
        from("timer://foo?period=5000").routeId("diskmon")
            .setBody(constant("SELECT AVG(PERCENT_USED)FROM QSYS2.SYSDISKSTAT A"))
            .to("jdbc:jt400")
            .process((exchange) -> {
                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)

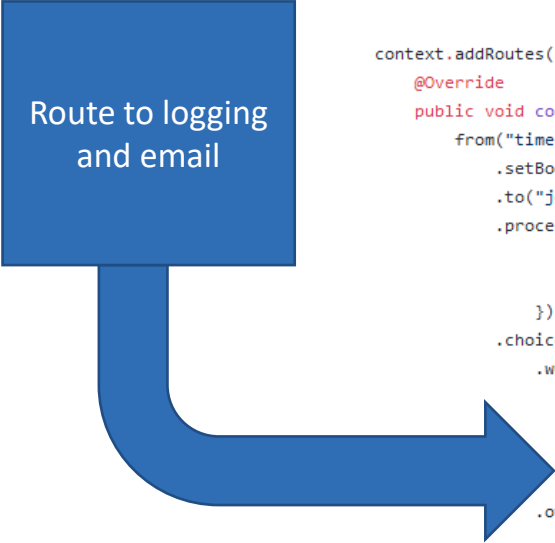


```
context.addRoutes(new RouteBuilder() {
    @Override
    public void configure() {
        from("timer://foo?period=5000").routeId("diskmon")
            .setBody(constant("SELECT AVG(PERCENT_USED)FROM QSYS2.SYSDISKSTAT A"))
            .to("jdbc:jt400")
            .process((exchange) -> {
                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 "+ exch
```

Disk Monitor Example

Route to logging
and email



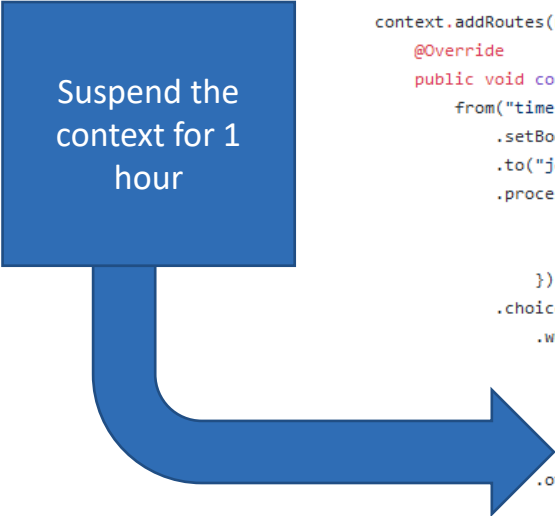
```
context.addRoutes(new RouteBuilder() {
    @Override
    public void configure() {
        from("timer://foo?period=5000").routeId("diskmon")
            .setBody(constant("SELECT AVG(PERCENT_USED)FROM QSYS2.SYSDISKSTAT A"))
            .to("jdbc:jt400")
            .process((exchange) -> {
                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); })
```

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 "n".

Closing thoughts

Where can I get this code?

- Most of the code discussed today is published at <https://github.com/ThePrez/OSS-integration-pptx>
 - Needs your own setup (Twilio account, etc)
 - Questions or problems? Just open an issue!
- DNS Lookup sample code:
 - https://github.com/IBM/ibmi-oss-examples/tree/master/nodejs/fastify/dns_api



Open Source 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/ibmiooss-chat> (join at <http://ibm.biz/ibmiooss-chat-join>)
- Jesse
 - jgorzins@us.ibm.com
 - <http://twitter.com/IBMJesseG>

Brand new landing page:
<http://ibm.biz/ibmiooss>

Jesse's Favorites

REST

UNIXCMD

Questions?

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



Notices and disclaimers

© 2021 International Business Machines Corporation. No part of this document may be reproduced or transmitted in any form without written permission from IBM.

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.⁸⁶

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