

V O L V O

Common Europe Congress 2024

June 3-6, Milan

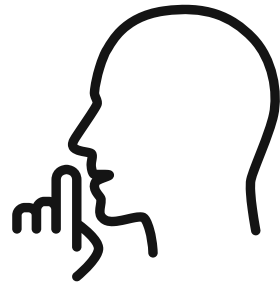
IBM i App Dev

Does Apache Camel like IBMi?

MARCIN OGOŃ

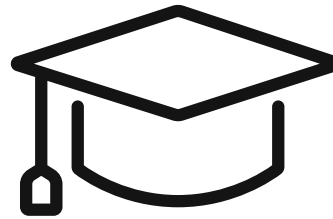


Practicalities



Presentation

Structured presentation of slides. Please hold all questions until the end to maintain focus and flow.



Demo

Explore a practical demonstration of the concepts with an example application. Observe the practical application of theoretical knowledge.



Discussion

Open discussion. Feel free to ask questions, share insights, and engage in conversation about the presented topics.

V O L V O

INTRODUCTION AND MOTIVATION

Does Apache Camel like IBMi | Marcin Ogoń

2025-10-16

Meet Camel

Apache Camel is an open-source integration framework that simplifies the process of connecting different applications, systems, or data sources

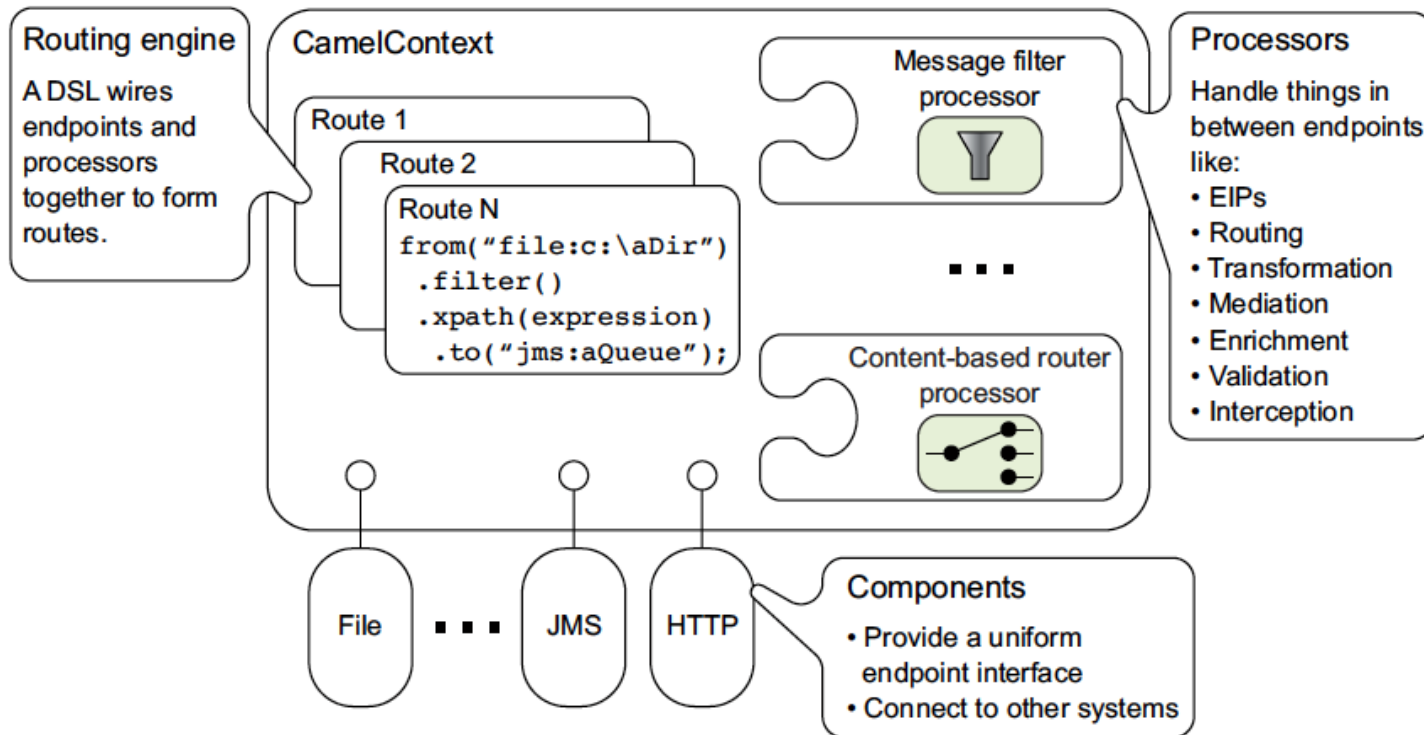
Project in numbers

- First release: June 27, 2007; 16 years ago
- Contributors: 1048
- Releases: 252 tags
- License: Apache-2.0 license
- 5.4k stars
- 4.9k forks



Meet Camel

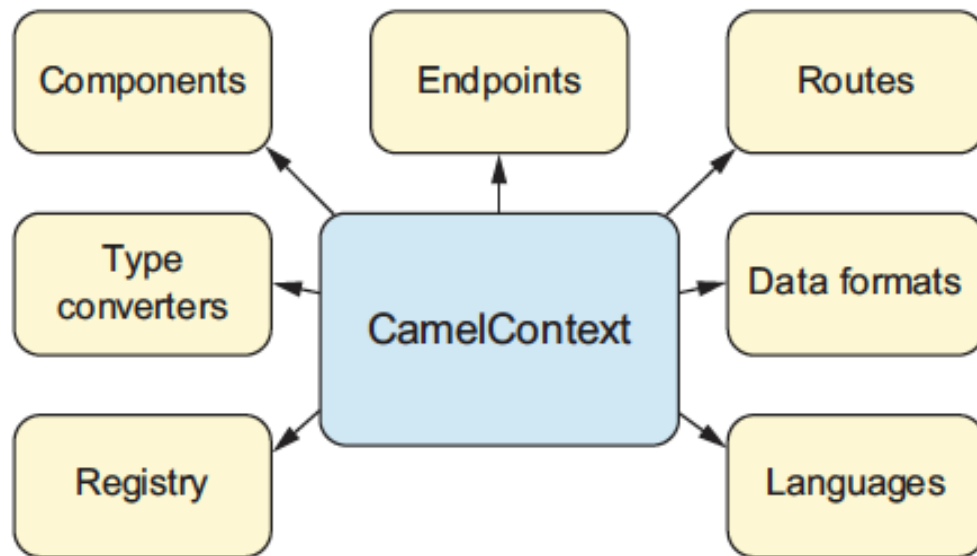
Architecture



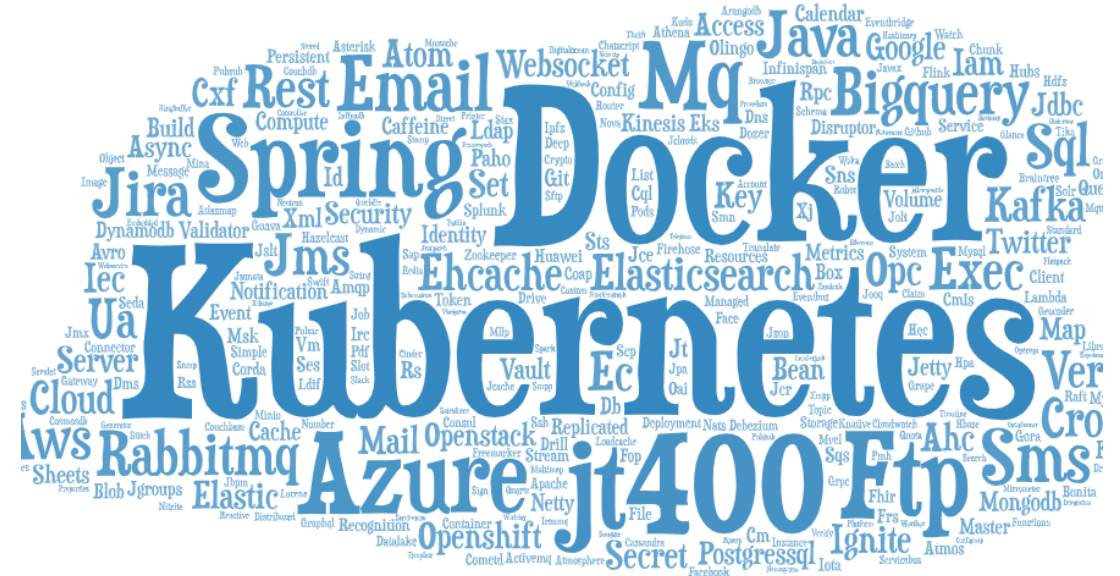
- Camel Context
 - Routing engine
Domain Specific Language
 - Components
 - Processors
Enterprise Integration Patterns (EIP)

Meet Camel

Camel Context and components



Camel Context



315 Stable Components

Meet Camel

Processors

- Transformer

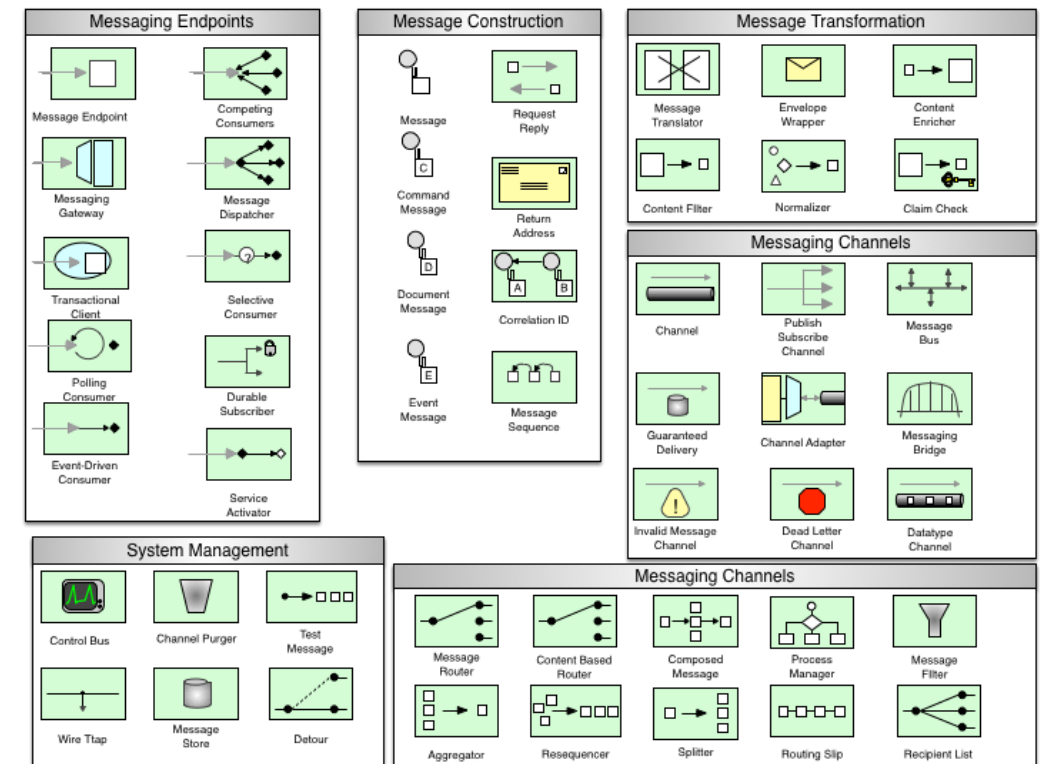
```
from("direct:start")
    .process(new Processor() {
        public void process(Exchange exchange) {
            Message msg = exchange.getMessage();
            msg.setBody(msg.getBody(String.class) +
                " World!"); } })
    .to("mock:result");
```

- Content Enrichment

```
from("activemq:My.Queue")
    .bean("myBeanName", "doTransform")
    .to("activemq:Another.Queue");
```

- Validator

```
from("file:inbox")
    .validate(body(String.class)
        .regex("^\\w{10}\\|\\|\\d{2}\\|\\|\\w{24}$"))
    .to("bean:myServiceBean.processLine");
```



When the Camel Met Spring Boot

An Integration Love Story

- Self-contained application
 - Single executable JAR - ibmismart-0.0.1.jar
 - Easy to run „java -jar ibmismart-0.0.1.jar”
- Embedded Server
 - Tomcat
 - Jetty
 - Undertow
- Easier to configure the application for different environments
- Production-ready features like health checks, metrics and monitoring capabilities out of the box

When the Camel Met Spring Boot

JT400 component



When the Camel Met Spring Boot

SQL component

Configuration

```
spring.datasource.username= ${ibmi.username}
spring.datasource.password= ${ibmi.password}
spring.datasource.driver-class-name=com.ibm.as400.access.AS400JDBCDriver
spring.datasource.url=jdbc:as400://${ibmi.hostname};libraries=*LIBL,ibmismart,qgpl,sysibm,qsys2;
errors=full;hold statements=true;naming=system
```

Using named parameters

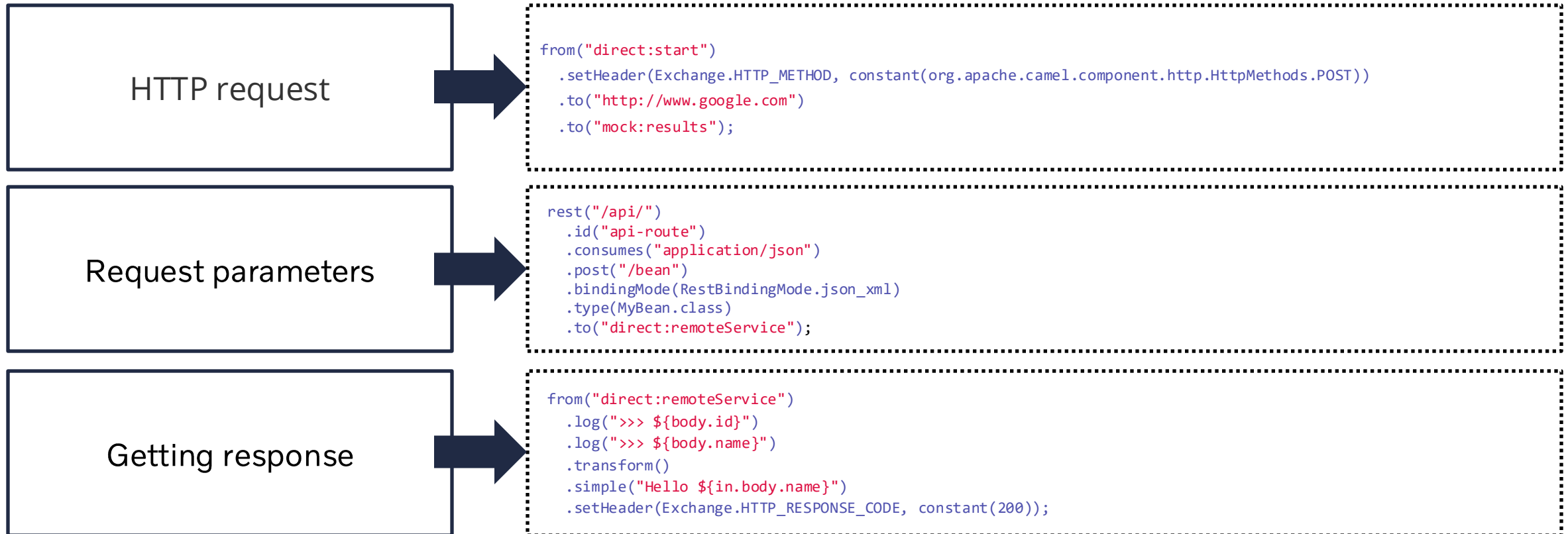
```
from("direct:projects")
  .setBody(constant("ASF"))
  .setProperty("min", constant(123))
  .to("sql:select * from projects where license = :${body} and id > :${exchangeProperty.min} order by id")
```

SQL queries from files in
the classpath

```
from("direct:query")
  .to("sql:classpath:sql/selectProjectsIn.sql")
  .to("log:query")
  .to("mock:query");
```

When the Camel Met Spring Boot

HTTP i REST component



Invite Camel to your home

Concept

- Calming green screen to manage IoT
- Communication
 - Database polling
 - HTTP request for state change
 - Get current state with REST
 - Change log in DTAQ
- Smart Home Automation
 - LED Mood Light
 - Power Switch



Invite Camel to your home

- OpenAPI documentation for REST API
- REST component usage

```
.apiContextPath("/api-doc")
.apiProperty("api.title", "State API")
.apiProperty("api.version", "1.0.0");
```

```
rest("/state").description("Device state REST service")
    .get().description("State").outType(String.class)
    .responseMessage().code(200)
    .message("Device state successfully returned")
    .endResponseMessage();
```

```
{
  "openapi" : "3.0.2",
  "info" : {
    "title" : "State API",
    "version" : "1.0.0"
  },
  "servers" : [ {
    "url" : "/api"
  } ],
  "paths" : {
    "/api/state" : {
      "get" : {
        "tags" : [ "/state" ],
        "responses" : {
          "200" : {
            "content" : {
              "text/plain" : {
                "schema" : {
                  "format" : "string",
                  "type" : "string"
                }
              }
            }
          }
        },
        "description" : "Device state successfully returned"
      }
    },
    "operationId" : "verb1",
    "summary" : "State"
  }
},
"components" : { },
"tags" : [ {
  "name" : "/state",
  "description" : "Device state REST service"
} ]
}
```

JT400 component usage

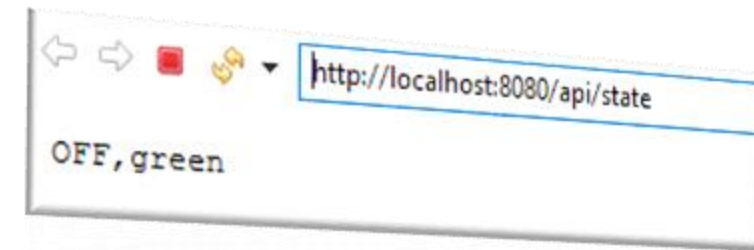
Response from call output fields

- Parameters
- RPGLE program call

```
.to("jt400://{{ibmi.username}}:{{ibmi.password}}@{{ibmi.hostname}}"  
+ "/QSYS.LIB/IBMISMART.LIB/IBMISMART.PGM?"  
+ "outputFieldsLengthArray=3,8&outputFieldsIdxArray=0,1")
```

- Send data to a data queue

```
.to("jt400://{{ibmi.username}}:{{ibmi.password}}@{{ibmi.hostname}}"  
+ "/QSYS.LIB/IBMISMART.LIB/IBMISMART.DTAQ")
```



Source codes

- <https://github.com/marcin-ogon/ibmi-smart-green>
- <https://github.com/marcin-ogon/ibmi-smart-camel>

V O L V O