VOLVO

APACHE CAMEL I IBMI

integracja usług

Apache Camel i IBMi – integracja usług / Marcin Ogoń

2025-10-17

VOLVO GROUP DIGITAL & IT

WHO WE ARE I OUR STRATEGIC DIRECTION I WHAT WE DO

Volvo Group Digital & IT

Volvo Group Digital & IT manages overall digital and IT strategies and plans for the Volvo Group and has the end-to-end responsibility for all development, delivery, and support of IT solutions and services globally.

VOLVO PENTA

VOLVO ENERGY

VOLVO AUTONOMOUS SOLUTIONS

VOLVO FINANCIAL SERVICES

ARQUUS

MACK TRUCKS

RENAULT TRUCKS

VOLVO TRUCKS

VOLVO CONSTRUCTION EQUIPMENT

VOLVO BUSES

GROUP TRUCKS TECHNOLOGY

GROUP TRUCKS OPERATIONS

GROUP TRUCKS PURCHASING

GROUP PEOPLE & CULTURE

GROUP FINANCE

We are colleagues from around the globe

Global presence:

~4800 colleagues located in more than 30 countries

- Main sites
- Other sites

Poznajemy Wielbłąda

Ogólna charakterystyka

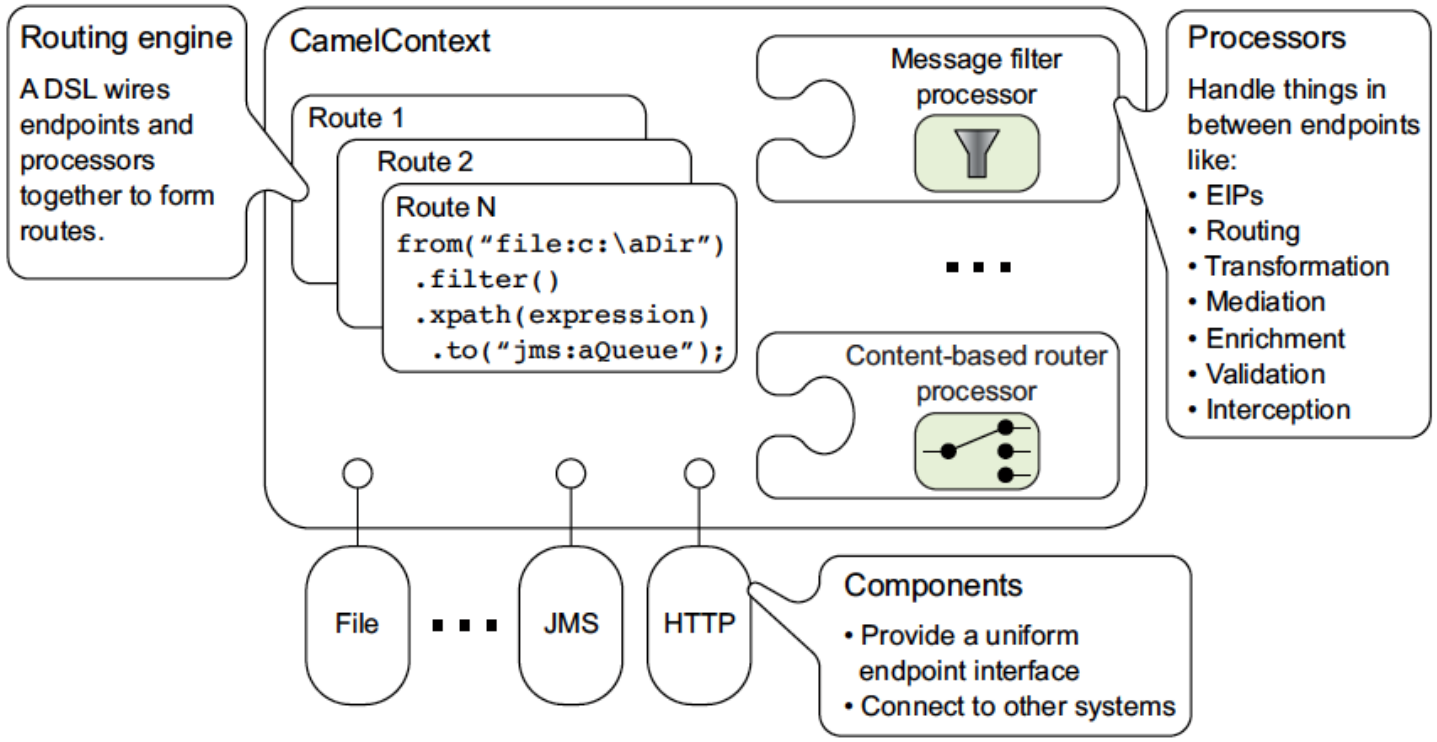
Apache Camel to framework do integracji wspierający szybkie i łatwe łączenie przeróżnych systemów konsumujących lub produkujących dane.

Projekt w liczbach:

- Pierwsze wydanie: June 27, 2007; 15 years ago
- Współtwórcy: 900
- Wydania: Releases 204 tags
- Licencja: Apache-2.0 license
- Ocena: Stars 4.6k stars / Forks 4.6k forks

Poznajemy Wielbłąda

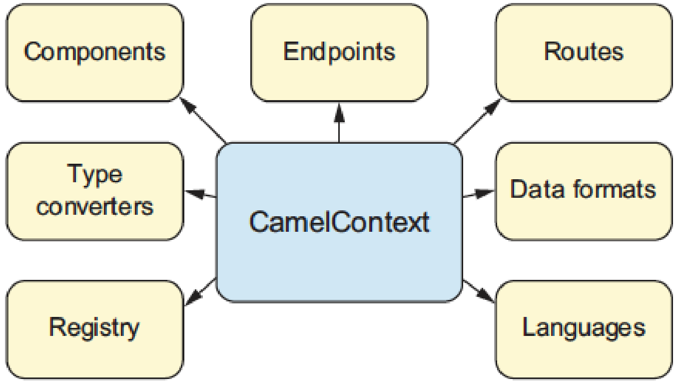
Architektura



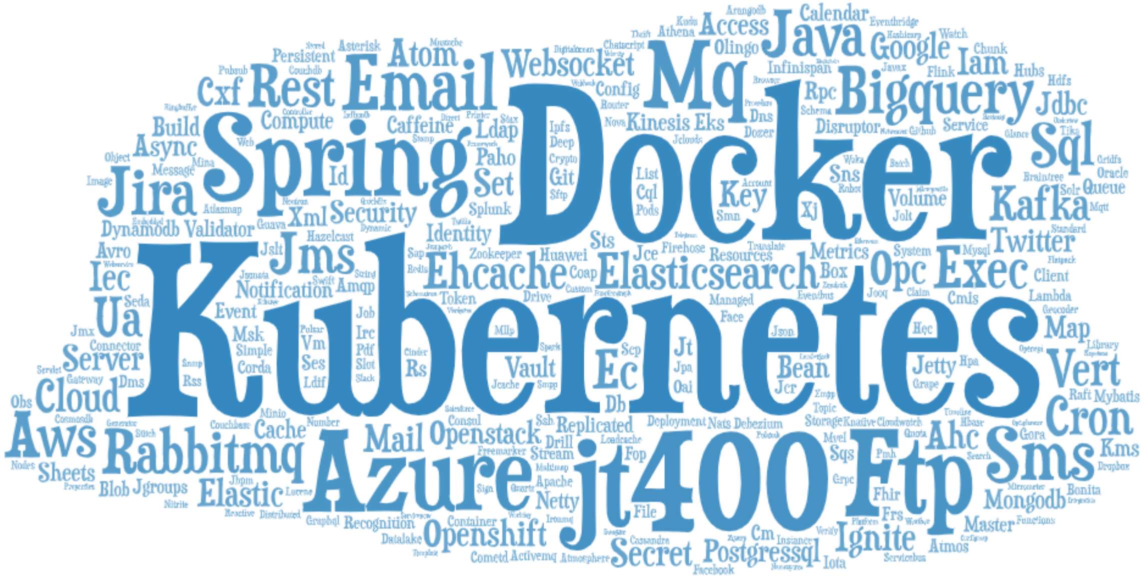
- Camel Context
  - Silnik routingu
    - Język dziedzinowy
  - Komponenty
  - Procesory
    - Wzorce integracji (EIP)

Poznajemy Wielbłąda

Camel Context i komponenty



Camel Context



342 dostępne komponenty

Poznajemy Wielbłąda

Procesory

- Przykładowe użycia
  - Transformacja

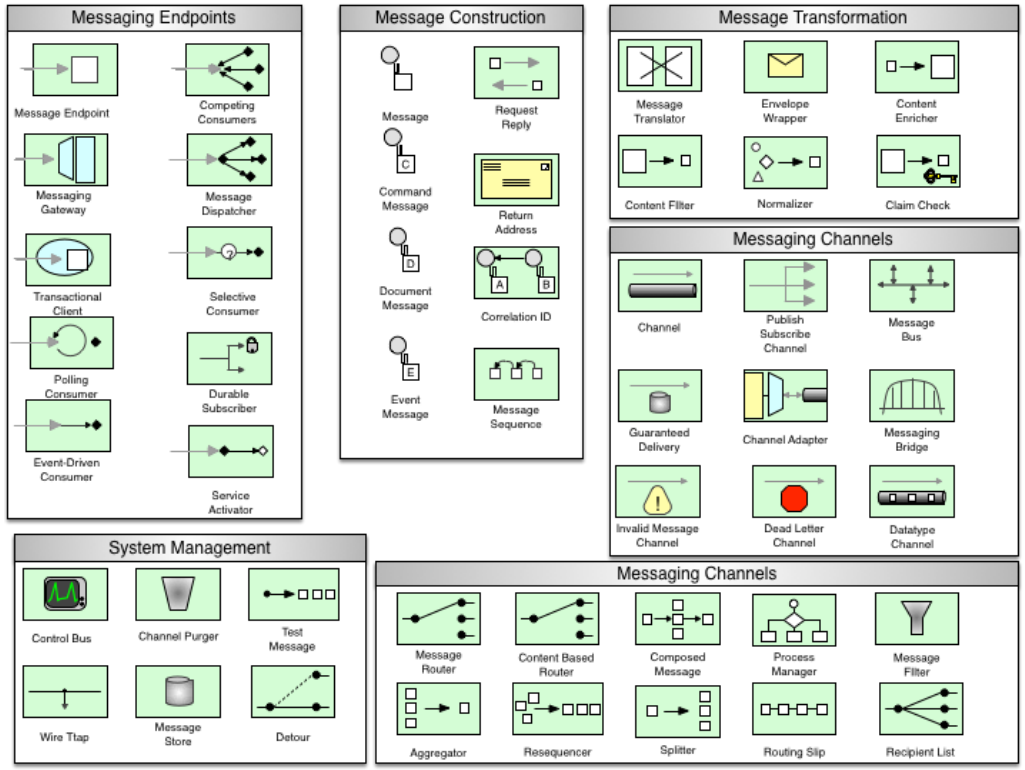
```
from("direct:start")
  .process(new Processor() {
  public void process(Exchange exchange) {
    Message msg = exchange.getMessage();
    msg.setBody(msg.getBody(String.class) +
    " World!"); } })
.to("mock:result");
```

  - Wzbogacanie

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

  - Validacja

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



Wzorce integracji - Enterprise Integration Patterns - EIP

Wiosenny Wielbład – przyjaźń ze Spring Boot'em

- Utworzenie niezależnej aplikacji Spring
  - Pojedyńcze archiwum ibmismart-0.0.1.jar
  - Proste uruchamianie „java –jar ibmismart-0.0.1.jar”
- Wbudowany serwer aplikacji webowych Tomcat, Jetty lub Undertow
- Zapewnia konfiguracje podstawowe
- Posiada dodatki pomagające w utrzymaniu oraz zarządzaniu aplikacją na poziomie produkcyjnym

Spring Boot = Spring Framework + Kontener Aplikacji + Konfiguracja

Wielbłąd zaprzyjaźnia się z IBMi

Komponent JT400

Uruchamianie programów

```
public class Jt400RouteBuilder extends RouteBuilder {
  @Override
  public void configure() throws Exception {
    from("direct:work")
    .to("jt400://GRUPO:ATWORK@server/QSYS.LIB/assets.LIB/compute.PGM?fieldsLength=10,10,512&ouputFieldsIdx=2,3")
    .to(“direct:play”);
} }
```

Odczyt/Zapis kolejek

Data Queue

```
public class Jt400RouteBuilder extends RouteBuilder {
  @Override
  public void configure() throws Exception {
    from("direct:george").to("jt400://GEORGE:EGROEG@LIVERPOOL/QSYS.LIB/BEATLES.LIB/PENNYLANE.DTAQ");
    from("jt400://RINGO:OGNIR@LIVERPOOL/QSYS.LIB/BEATLES.LIB/PENNYLANE.DTAQ").to("mock:ringo");
} }
```

Odczyt/Zapis i odpowiedzi na komunikaty w kolejce komunikatów

```
from("jt400://username:password@localhost/qsys.lib/qusrsys.lib/myq.msgq?sendingReply=true")
.choice()
  .when(header(Jt400Constants.MESSAGE_TYPE).isEqualTo(AS400Message.INQUIRY))
    .process((exchange) -> {
      String reply = // insert reply logic here
      exchange.getIn().setBody(reply);
    }).to("jt400://username:password@localhost/qsys.lib/qusrsys.lib/myq.msgq");
```

Wielbłąd zaprzyjaźnia się z IBMi

Komponent SQL

Konfiguracja

```
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
```

Przekazywanie parametrów do zapytania

```
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")
```

Zapytania jako osobny plik w ścieżce klas

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

Wielbłąd zaprzyjaźnia się z IBMi

Komponent HTTP i REST

Zapytanie HTTP

```
from("direct:start")
  .setHeader(Exchange.HTTP_METHOD, constant(org.apache.camel.component.http.HttpMethods.POST))
  .to("http://www.google.com")
  .to("mock:results");
```

Przekazywanie parametrów do zapytania

```
rest("/api/")
  .id("api-route")
  .consumes("application/json")
  .post("/bean")
  .bindingMode(RestBindingMode.json_xml)
  .type(MyBean.class)
  .to("direct:remoteService");
```

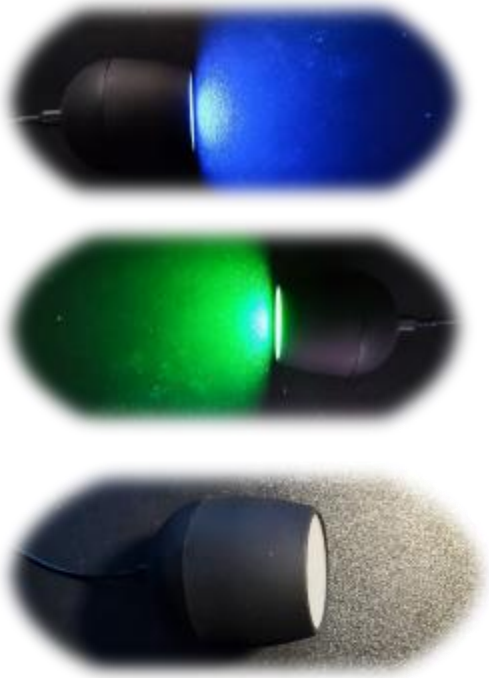
Przygotowanie odpowiedzi

```
from("direct:remoteService")
  .log(">>> ${body.id}")
  .log(">>> ${body.name}")
  .transform()
  .simple("Hello ${in.body.name}")
  .setHeader(Exchange.HTTP_RESPONSE_CODE, constant(200));
```

Zaproś Wielbłąda do domu

Koncepcja

- Zielono kojący ekran do obsługi urządzenia IoT
- Interfejs komunikacji
  - Monitoring zmian na pliku bazy danych
  - Żądanie HTTP zmiany stanu urządzenia
  - Status zmian jako usługa REST
  - Log zmian w DTAQ
- Urządzenie SMART Lampka



Zaproś Wielbłąda do domu

- Dokumentaja REST OpenApi
- Użycie komponentu REST

```
.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"
 } ]
}
```

Zaproś Wielbłąda do domu

Użycie komponentu JT400

- Odpowiedź API jako wynik wołania programu
  - Przygotowanie parametrów
  - Uruchomienie programu RPGLE

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

  - Wysłanie komunikatu do DTAQ

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

VOLVO

Marcin Ogoń