Apache Camel

Enterprise Integration and Data Workflow Made Easy
by Matthew Tyler
Source Material Credits

**Claus Ibsen** - Principal Software Engineer, RedHat, author of “*Camel in Action*” (a must read)

**Christian Posta** - Principal Middleware Specialist/Architect, RedHat
Agenda

- What is Apache Camel?
- Features and Benefits
- Deployment Options
- Some Examples
- Q and A
What is Camel?
What is Apache Camel?

From the Apache Camel Web site:

Apache Camel is a powerful Open Source Integration Framework based on known Enterprise Integration Patterns.
What is Integration?

- Why do we need integration?
  - Critical for your business to integrate

- Why Integration Framework?
  - Framework do the heavy lifting
  - You can focus on business problem
  - Not "reinventing the wheel"

- But really, why do we need integration?
wrong!
When do you need it?

Use Camel to integrate disparate systems that speak different protocols and data formats.
What is Enterprise Integration Patterns?
EIP - Order Processing
**Logical Description**

from **newOrder**

choice

when **isWidget** to **widget**
when **isGadget** to **gadget**
otherwise to **inquiry**
import org.apache.camel.builder.RouteBuilder;

public class MyRoute extends RouteBuilder {

    public void configure() throws Exception {

        from("activemq:queue:newOrder")
            .choice()
                .when(xpath("/order/product = 'widget'"))
                    .to("activemq:queue:widget")
                    .otherwise()
                        .to("activemq:queue:gadget")
            .end();
    }
}
<route>
  <from uri="activemq:queue:newOrder"/>
  <choice>
    <when>
      <xpath>/order/product = 'widget'</xpath>
      <to uri="activemq:queue:widget"/>
    </when>
    <otherwise>
      <to uri="activemq:queue:gadget"/>
    </otherwise>
  </choice>
</route>
Components and Patterns

http://camel.apache.org/components.html

“What is it” Summary

- Integration Framework
- Enterprise Integration Patterns (EIP)
- Routing (using DSL)
- Easy Configuration (endpoint as uri's)
- Payload Agnostic
- No Container Dependency
- A lot of components
Features & Benefits
Features

- Enterprise Integration Patterns (EIPs)
- Domain Specific Language to write “flows” or “routes”
- Large collection of adapters/components for legacy systems, B2B, and SaaS
- Strong Unit test/Integration test framework
- Expression languages
- Data Formats
- Tooling with JBoss Developer Studio
Open Source

- Apache Camel is 100% open source
- JBoss Fuse (built with Camel and other Apache projects) is 100% open source
- Vibrant communities
  - Mailing lists
  - Code commits
  - Issue trackers
  - Visible community members
  - Blogging, books, social media
“I cannot understand the benefit of Apache Camel as a lot of code is required”
Benefits

- Significantly reduces the amount of code required to achieve complex integrations
- Comes with all the tooling “built-in” that provides:
  - Connection handling
  - Retry logic
  - Distributed transactions
  - Logging
  - Remote management
Six organizations studied
Telecommunications, IT, shipment and logistics, and document management

Conducted by RedHat
RedHat Study

- 51.5% more applications integrated per year
- 40.8% fewer FTEs per application integration
- 62.8% less downtime related to integration
- 18.1% improved performance
- 34.2% less costly than previous middleware integration solution stackres
“Apache Camel doesn’t do what ESB X does”
Camel is not an ESB

- Pick the right tool (architecture) for the job!
- Not forced into expensive, mountainous suites of applications
- If you want to compare ESBs, don’t compare with Camel
- For ESB, look at JBoss Fuse
<table>
<thead>
<tr>
<th>Animal</th>
<th>Typical weight kN (kgf)</th>
<th>Pull-weight ratio</th>
<th>Typical pull N (kgf)</th>
<th>Typical working speed m/s</th>
<th>Power output W</th>
<th>Working hours per day</th>
<th>Energy output per day MJ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ox</td>
<td>4.5 (450)</td>
<td>0.11</td>
<td>500 (50)</td>
<td>0.9</td>
<td>450</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>Buffalo</td>
<td>5.5 (50)</td>
<td>0.12</td>
<td>650 (65)</td>
<td>0.8</td>
<td>520</td>
<td>5</td>
<td>9.5</td>
</tr>
<tr>
<td>Horse</td>
<td>4.0 (400)</td>
<td>0.13</td>
<td>500 (50)</td>
<td>1</td>
<td>500</td>
<td>10</td>
<td>18</td>
</tr>
<tr>
<td>Donkey</td>
<td>1.5 (150)</td>
<td>0.13</td>
<td>200 (20)</td>
<td>1</td>
<td>200</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Mule</td>
<td>3.0 (300)</td>
<td>0.13</td>
<td>400 (40)</td>
<td>1</td>
<td>400</td>
<td>6</td>
<td>8.5</td>
</tr>
<tr>
<td>Camel</td>
<td>5.0 (500)</td>
<td>0.13</td>
<td>650 (65)</td>
<td>1</td>
<td>650</td>
<td>6</td>
<td>14</td>
</tr>
</tbody>
</table>

Deployment
Deployment Options

- Standalone
- Spring (Boot, WAR)
- OSGi (Karaf/FUSE, Blueprint)
- JEE (CDI)
- Guice
- Easy to custom integrate with any container
Some Examples
Examples From Camel Distribution
More questions than answers?