JBoss Fuse to JBoss Data Virtualization Integration

Bill Kemp
Kenny Peeples
Cojan van Ballegooijen

June 24, 2015
SUMMIT BY DAY
PARTY BY NIGHT

JOIN OUR JBOSS, OPENSHEIF, AND MOBILE TEAMS ON WED. JUNE 24 FOR A NIGHT OF GAMES, DANCING, AND OPEN CONTAINERS

Visit the Red Hat booth in Hall D for location and invitation.

An invitation doesn't guarantee entrance. Admission determined by city of Boston fire code.
Agenda

• Introductions

• JBoss Fuse Overview

• JBoss Data Virtualization Overview

• Camel components to integrate JBoss Fuse with JBoss Data Virtualization

• Demo Time
JBoss Fuse Overview
JBoss Fuse Overview

- Apache project originally created by Fusesource
- Karaf OSGi container
- Camel mediation framework for connectivity and transformation
- Active MQ messaging
- Apache CXF services
- Provides support for over 150 protocols for integration through Camel Components
- Provides EIP via Camel Processors to transform and route messages based on the Hohpe & Wolfe 'Enterprise Integration Patterns'
- De Facto standard for enterprise integration
JBoss Fuse Architecture
JBoss Fuse – Includes Apache Camel

Camel
Integration Engine And Router

Camel Endpoints
* Camel can send messages to them
* Or receive messages from them

Camel Components
* Provide a uniform Endpoint Interface
* Act as connectors to all other systems

Camel Processors
* Are used to wire Endpoints together
  * Routing
  * Transformation
  * Mediation
  * Interception
  * Enrichment
  * Validation
  * Tracking
  * Logging

JMS Component
HTTP Component
File Component
JBoss Data Virtualization Overview
JBoss Data Virtualization Overview

- Enterprise implementation of the Teiid project
- Allows data virtualization of large set of data sources
- RDBMS via JDBC & ODBC
- Files, Web Services, RESTful services, NoSQL, Data Grid, SAP, and others
- Client exposure via JDBC, ODBC, SOAP, REST
- Preserves 'in place' data via native queries to underlying datasources
- Data modeling through Teiid Designer Eclipse plugin
- RBAC to virtual models that expose controlled slices of physical data
- Caching of result sets, code tables using Infinispan
- Caching of views as Materialized Views through single click configuration in Teiid Designer
JBoss Data Virtualization Architecture

DATA CONSUMERS
- Business intelligence tools and analytical applications
- Mobile and enterprise applications
- Enterprise service bus (ESB), extract transform load (ETL)
- Service-oriented architecture (SOA) applications and portals

DATA VIRTUALIZATION
- CONSUME: Standards-based data provisioning
  - JDBC, ODBC, SOAP, REST, OData
- COMPOSE: Unified virtual database/Common data model
  - Unified customer view
  - Unified product view
  - Unified supplier view
- CONNECT: Standards-based data connectivity

DATA SOURCES
- NoSQL
- Hadoop
- Databases and data warehouse
- Mainframe
- Enterprise applications
- Excel, CSV or XML files
- SaaS and cloud applications

Design tools
Dashboard
Optimization
Caching
Security
Metadata
Camel Components to Integrate Fuse with JDV
Fuse-Data Virtualization Architecture

Fuse provides APIs and Integration Services

Multiple components can be used for access to data services including:
- JDBC Component
- SQL Component
- CXF Component
- Olingo Component

Data Virtualization provides Data Services

- ID
- Calls
- Sentiment

CustomerContextVDB

- ID
- Name
- Credit Score

Virtual Databases can be deployed in a container or in the cloud or both. VDBs can expose data through REST APIs

Apache Camel

Teiid

#redhat #rhsummit
Camel JDBC Component

- Allows JDBC access to a database
- SQL queries and operations are passed in the message body
- Producer endpoints only
- URI format → jdbc:datasourcename[?options]
- Options to configure the query
- Results returned as ArrayList<HashMap<String, Object>> in OUT body of message
- List entries for each row returned
- Map String=ColumnName, Object=ColumnValue
Camel JDBC Route
Camel SQL Component

- Also allows JDBC access to databases
- Uses spring-jdbc 'under the covers'
- Query is defined in the URI to the endpoint
- URI → sql:select * from table where id=:#myId order by name[?options]
- Named parameters in message body passed as java.util.Map in message body or in as named parameters in the message header
- Query results returned as List<Map<String, Object>> in the message body
- Update operation return Integer in the message body for the count of rows affect
- Easy, peasy
  from("jms:order.inbox")
    .to("sql:select order_seq.nextval from dual?
         outputHeader=OrderId&outputType=SelectOne")
    .to("jms:order.booking");
Camel SQL Route

timer://foo?fixed...

log

sql:select * from...

log

split[bean dvBean...]

direct:mysplit

log
Camel Olingo2 Component

• Available in Camel 2.14
• The Olingo2 component utilizes Apache Olingo version 2.0 API’s to interact with OData 2.0 and 3.0 component service
• URI format - olingo2://endpoint/<resource-path>?[options]
• Will be part of JBoss Fuse 6.2
• Easy integration with Data Virtualization
Camel Olingo2 Route
Camel Jetty Component

- The jetty component provides HTTP-based endpoints for consuming and producing HTTP requests
- the Jetty component behaves as a simple Web server, or
- Jetty can also be used as a http client, or Camel producer
- URI format – jetty:http://hostname[:port][/resourceUri][?options]
Camel Jetty Route
Fuse – Data Virtualization Integration
Demo Time
Q&A + Discussion
LEARN. NETWORK. EXPERIENCE OPEN SOURCE.