Ernhofer, Kopec

Exposee

In diesem Dokument wird ein ETL-Example (Extract, Transform, Load) mit Hilfe von Apache Camel ausführlich beschrieben.

Enterprise Application Integration

DEZYS-09

Inhalt

[Angabe 2](#_Toc412468821)

[Zeitaufzeichnung & Arbeitsaufteilung 2](#_Toc412468822)

[Identifikation und Beschreibung der EAI Patterns 2](#_Toc412468823)

[Beschreibung der Funktionsweiße von Apache Camel 3](#_Toc412468824)

[Protokollierung der Inbetriebnahme des Beispiels 4](#_Toc412468825)

# Angabe

"The ETL (Extract, Transform, Load) is a mechanism for loading data into systems or databases using some kind of Data Format from a variety of sources; often files then using Pipes and Filters, Message Translator and possible other Enterprise Integration Patterns. So you could query data from various Camel Components such as File, HTTP or JPA, perform multiple patterns such as Splitter or Message Translator then send the messages to some other Component.

To show how this all fits together, try the ETL Example."

ETL ist ein wichtiger Prozess bei einem Datawarehouse. Zeigen Sie wie Enterprise Integration Patterns dabei eingesetzt werden können (8 Punkte, nur jene, die in dem Beispiel vorkommen). Verwenden Sie dazu das ETL Example. Dokumentieren Sie die Implementierung sowie alle notwendigen Schritte ausführlich in einem Protokoll (8 Punkte). Fügen Sie den verwendeten Code nach den Metaregeln an und geben Sie alles als ZIP-Archiv (Gesamtes Framework mit Anleitung, wie das System gestartet werden kann) ab.

**Resources**:

1. Extract Transform Load (ETL); Apache Camel; Online: http://camel.apache.org/etl.html; abgerufen 13.02.2015
2. Enterprise Integration Patterns; G.Hohpe, B.Woolf; 2003; Online: http://www.enterpriseintegrationpatterns.com/toc.html; abgerufen 13.02.2015
3. Extract Transform Load (ETL) Example; Apache Camel; Online: http://camel.apache.org/etl-example.html; abgerufen 13.02.2015

# Zeitaufzeichnung & Arbeitsaufteilung

|  |  |  |  |
| --- | --- | --- | --- |
| TASK | ZEIT GESCHÄTZT | ZEIT BENÖTIGT | ZUSTÄNDIG |
| Recherche zu Apache Camel | 1h | 1h | ERNHOFER  &  KOPEC |
| Beschreibung von Apache Camel | 30min | 15min | ERNHOFER  &  KOPEC |
| Recherche für das Ausführen des Beispiels auf MAC OSX | 1h | 45min | KOPEC |
| Ausführen auf MAC OSX | 15min | 15min | KOPEC |
| Recherche für das Ausführen des Beispiels aus Linux | 1h | 45min | ERNHOFER |
| Ausführen auf Linux | 15min | 15min | ERNHOFER |
| Recherche für das Ausführen des Beispiels in Eclipse | 1h | 2h | ERNHOFER  &  KOPEC |
| In Eclipse Importieren und Ausführen | 15min | 30min | ERNHOFER  &  KOPEC |
| Dokumentation der Inbetriebnahme | 30min | 30min | ERNHOFER  &  KOPEC |
| Dokumentation des Source-Codes | 1h | 3h | ERNHOFER  &  KOPEC |

# Identifikation und Beschreibung der EAI Patterns

ExchangePattern

# Beschreibung der Funktionsweiße von Apache Camel

**Beschreibung**:

Apache Camel ist eine freie, regelbasierte Routing- und Konvertierungsengine. Mit Apache Camel kann man Routing- und Konvertierungsregeln deklarativ in Java oder Scala basierend auf einer domänenspezifischen Sprache, oder mittels Spring basierter XML-Konfiguration definieren.

**Funktionsweise**:

Apache Camel basiert auf Enterprise Integration Patterns - Entwurfsmuster welche für den Entwurf von Enterprise Application Integration und Message Oriented Middleware basierten Systemen geschaffen wurden. Apache Camels Bean Binding unterstützt dabei Plain Old Java Objects und JavaBeans. Dadurch integriert es einfach mit Dependency Injection Frameworks wie Spring oder Google Guice.

Apache Camel verwendet Uniform Resource Identifiers und kann somit direkt mit unterschiedlichen Transport- und Messageprotokollen wie beispielsweise HTTP, JMS oder AMQP umgehen. Es kann so beispielsweise mit JBI, SCA, Apache ActiveMQ, RabbitMQ, Apache MINA oder Apache CXF zusammenarbeiten. Somit kann basierend auf der Apache Camel Programmierschnittstelle gearbeitet werden, obwohl die darüber angesprochenen Komponenten technologisch unterschiedliche Schnittstellen verwenden.

Apache Camel wird häufig zusammen mit Apache ServiceMix (Enterprise Service Bus), Apache CXF (Web Service Framework) und Apache ActiveMQ (Java Message Service Provider) in SOA Infrastruktur Projekten eingesetzt. Darüber hinaus wird oft auch Apache MINA (Framework für Netzwerkapplikationen) zusammen mit Apache Camel verwendet.

# Protokollierung der Inbetriebnahme des Beispiels

Protokoll Inbetriebnahme:

* git repository erstellt

link: https://github.com/jkopec/ETL-with-EAI.git

* Neues Java Projekt in Eclipse erstellt und mit git repository verbunden
* Eclipse: Help->Marketplace nach "Maven" gesucht
* Eclipse: M2Eclipse war aber bereits installiert
* apache-camel-2.14.1.tar.gz heruntergeladen

link: <http://tweedo.com/mirror/apache/camel/apache-camel/2.14.1/apache-camel-> 2.14.1.tar.gz

* apache-camel-2.14.1 entpackt
* src Ordner aus: "apache-camel-2.14.1/examples/camel-example-etl/" kopiert und den von Eclipse generierten

Ordner durch den eben kopierten ersetzt

* pom.xml aus: "apache-camel-2.14.1/examples/camel-example-etl/" kopiert und in die Eclipse workspace eingefügt
* dependency in pom.xml hinzugefügt

link: http://camel.apache.org/camel-eclipse.html

dependency:

<dependency>

<groupId>org.apache.camel</groupId>

<artifactId>camel-eclipse</artifactId>

<version>2.14.1</version>

<!-- use the same version as your Camel core version -->

</dependency>

* Eclipse geschlossen und in eclipse.ini mehr Arbeitsspeicher zugewiesen

link: http://camel.apache.org/building.html

unter mac osx: /Applications/Eclipse Java/Eclipse.app/Contents/MacOS/eclipse.ini

Eintrag von -Xmx512m auf -Xmx1536m geändert

* Eclipse wieder geöffnet
* Rechtsklick auf das Projekt Configure -> Convert to Maven Project
* Programm ausgeführt