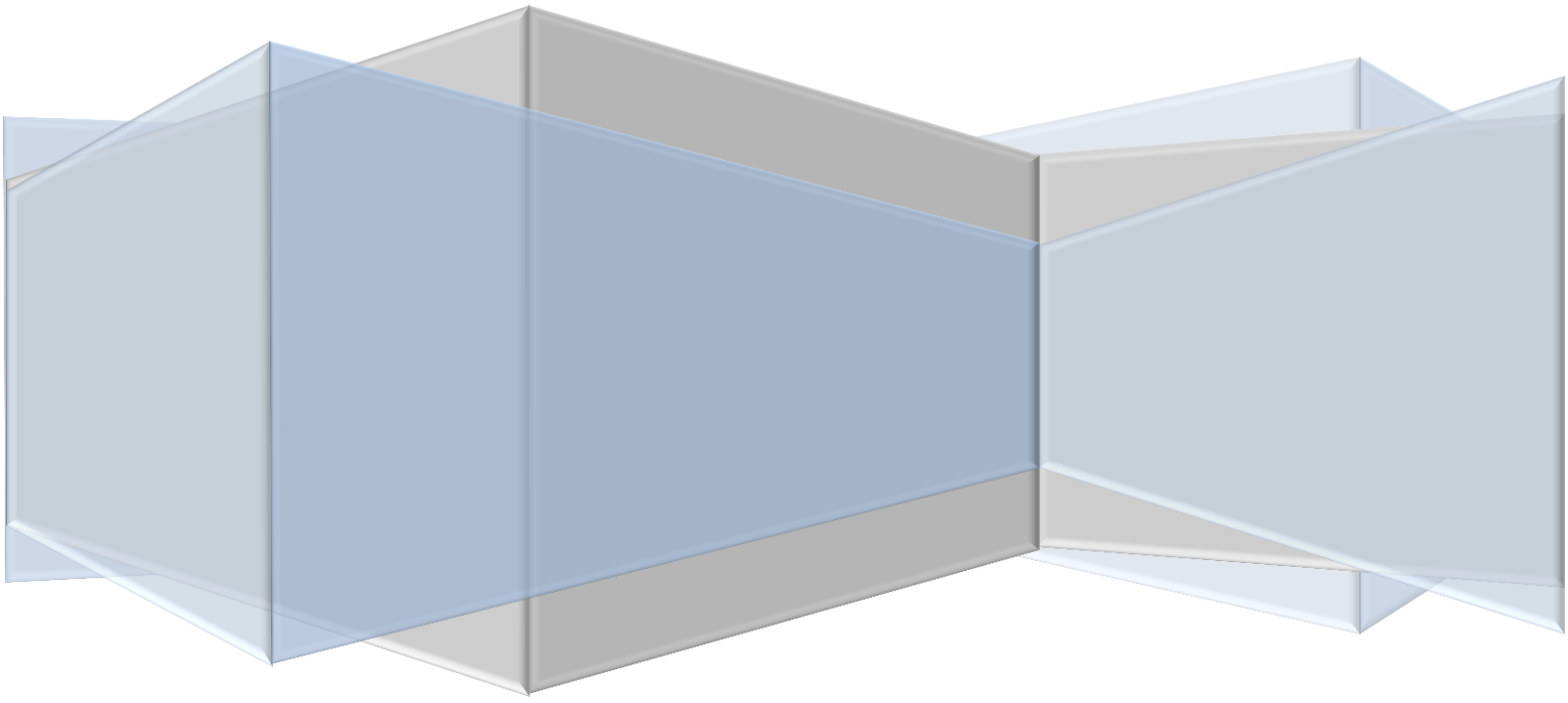
TGM Wien

DEZSYS09 - "EXTRACT, TRANSFORM AND LOAD (ETL) WITH EAI"

SYT 2014/2015 | Stand: 20. Feb. 2015

Michael Weinberger & Thomas Taschner 4AHIT



Inhalt

Aufgabenstellung

Beschreibung auf Moodle

Enterprise Application Integration

Gruppenaufgabe (2 Leute)

"*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." [1]

ETL ist ein wichtiger Prozess bei einem Datawarehouse. Zeigen Sie wie Enterprise Integration Patterns [2] dabei eingesetzt werden können (8 Punkte, nur jene, die in dem Beispiel vorkommen). Verwenden Sie dazu das ETL Example [3]. 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

Durchführung

Identifikation und Beschreibung der EAI Patterns

Integration Styles: File Transfer

Das im Example verwendete Integration Style Pattern entspricht einem File Transfer Pattern.
Es werden XML Dateien konsumiert und in POJO (**P**lain **O**ld **J**ava **O**bject) und mit Hilfe der JPA (**J**ava **P**ersistence **A**PI) transformiert.

Um verschiedene Programme miteinander kommunizieren und Informationen austauschen zu lassen, muss jede dieser Applikationen Dateien erstellen, die von anderen Programmen benötigt werden. Das Transformieren von Dateien in verschiedene Formate wird von ‚Integratoren' übernommen. [1]

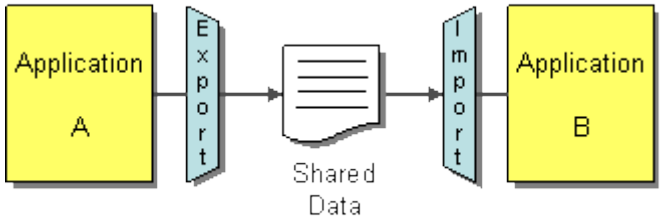


Abbildung 1: Ex- und Import von Daten zwischen zwei verschiedenen Applikationen [1]

Messaging System: Pipes and Filters

Das Verarbeiten von Daten kann in *Apache Camel* mit Hilfe von verschiedenen, voneinander unabhängigen, miteinander verknüpfbaren Instanzen von Endpunkten realisiert werden. [2]
Beispielsweise baut der Message Translator auf diesem Prinzip auf.

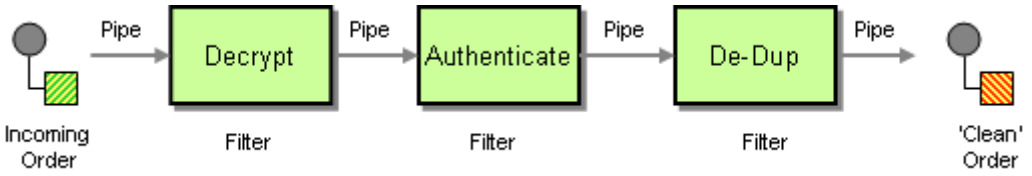


Abbildung 2: Verarbeitungsvorgang einer Nachricht mit Hilfe von Weiterleitungen und Filtern [2]

Messaging System: Message Translator

Zum Übersetzen von Nachrichten kann in *Apache Camel* der „Message Translator“, ein spezieller Filter eingesetzt werden. Dank ihm können Nachrichten zwischen 2 Filtern bzw. Anwendungen übersetzt und so miteinander kompatibel gemacht werden. [3]
Im Example findet sich der Message Translator in der **CustomerTransformer** Klasse wieder.

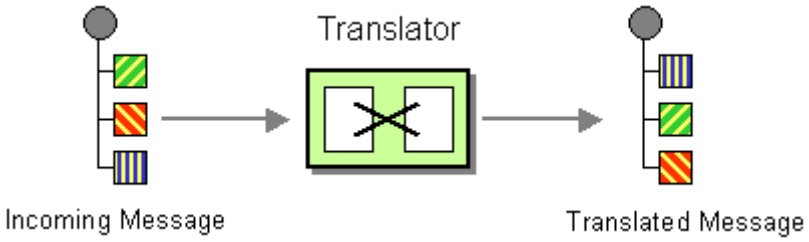


Abbildung 3: Übersetzen einer Nachricht mit Hilfe des Message Translators [3]

Messaging System: Message Endpoint

In *Apache Camel* können Nachrichten an Clients mit Hilfe eines Message Endpunktes geschickt werden. Anwendungen können mit Hilfe eines solchen Message Endpoints mit einem Messaging Channel (Nachrichtenkanal) verbunden werden. Dies ermöglicht schlussendlich einen Nachrichtenaustausch zwischen Applikationen. [4]
Angewendet wird dieses Prinzip im Example beispielsweise in der Klasse **EtlRoutes**.

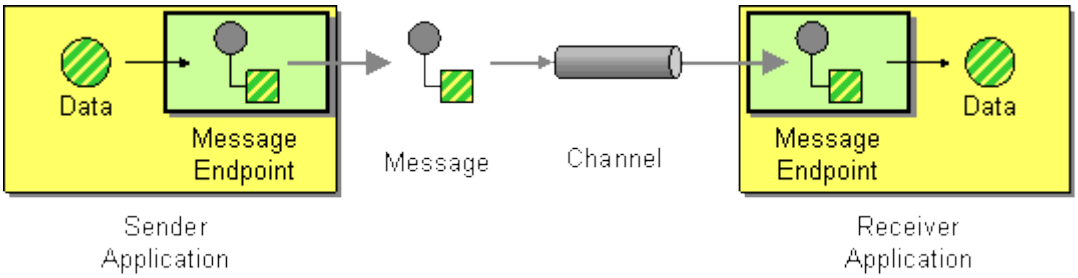


Abbildung 4: Übermitteln einer Nachricht mit Hilfe eines Endpoints über einen Datenkanal [4]

Beschreibung der Funktionsweise von Apache Camel

Apache Camel ist ein umfangreiches Framework, welches einem ermöglicht Code zu schreiben, der auf EIP aufbaut. EIPs (Enterprise Integration Pattern) sind Vorlagen bzw. neue Entwurfsmuster, die uns erlauben möglichst effektiv große komponentenbasierte Systeme zu entwerfen. Die Komponenten dieses Systems können dabei sowohl im gleichen Prozess, als auch auf einem gänzlich anderen Rechner laufen. Camel bietet hierfür häufig benötigte Pattern Implementierungen, Debugging Werkzeuge, ein Konfigurationssystem und vieles andere Kleinigkeiten an, die einem Entwickler viel Zeit sparen können. MVC (auch, wenn es nicht wirklich ein Pattern ist) könnte prinzipiell leicht implementiert werden. Mit Hilfe eines Frameworks, welches bereits eine sofort verwendbare Struktur anbietet, werden dem Entwickler bestimmte Aufgaben abgenommen, sodass er sich auf das Wesentliche konzentrieren kann. [5]

Protokollierung der Inbetriebnahme des Beispiels

1.) Herunterladen der Camel-Binaries von unserem Freund Klaus-Uwe.

```
schueler@debian:~/Downloads$ wget ftp://mirror2.klaus-uwe.me/apache/camel/apache
-camel/2.14.1/apache-camel-2.14.1.tar.gz
```

2.) Da kein Download zustande kommt, vertrauen wir lieber auf die TU Wien.

```
schueler@debian:~/Downloads$ wget ftp://gd.tuwien.ac.at/pub/infosys/servers/http
/apache/dist/camel/apache-camel/2.14.1/apache-camel-2.14.1.tar.gz
--2015-02-23 11:02:04--  ftp://gd.tuwien.ac.at/pub/infosys/servers/http/apache/d
ist/camel/apache-camel/2.14.1/apache-camel-2.14.1.tar.gz
           => 'apache-camel-2.14.1.tar.gz'
Resolving gd.tuwien.ac.at (gd.tuwien.ac.at)... 192.35.244.50
Connecting to gd.tuwien.ac.at (gd.tuwien.ac.at)|192.35.244.50|:21... connected.
Logging in as anonymous ... Logged in!
==> SYST ... done.    ==> PWD ... done.
==> TYPE I ... done.  ==> CWD (1) /pub/infosys/servers/http/apache/dist/camel/ap
ache-camel/2.14.1 ... done.
==> SIZE apache-camel-2.14.1.tar.gz ... 10300535
==> PASV ... done.    ==> RETR apache-camel-2.14.1.tar.gz ... done.
Length: 10300535 (9.8M) (unauthoritative)

apache-camel-2.14.1 100%[====================>]   9.82M  1.87MB/s   in 6.0s

2015-02-23 11:02:14 (1.64 MB/s) - 'apache-camel-2.14.1.tar.gz' saved [10300535]

schueler@debian:~/Downloads$
```

3.) Das Archiv wird mit tar -xf entpackt.

```
schueler@debian:~/Downloads$ ls
apache-camel-2.14.1.tar.gz  Buttons-master  TGM-SCHUELER.pcf
schueler@debian:~/Downloads$ tar xf apache-camel-2.14.1.tar.gz
schueler@debian:~/Downloads$ ls
apache-camel-2.14.1         Buttons-master
apache-camel-2.14.1.tar.gz  TGM-SCHUELER.pcf
schueler@debian:~/Downloads$ rm apache-camel-2.14.1.tar.gz
schueler@debian:~/Downloads$ ls
apache-camel-2.14.1  Buttons-master  TGM-SCHUELER.pcf
schueler@debian:~/Downloads$
```

4.) Wir navigieren zu /apache-camel-…/examples/camel-example-etl.

```
camel-example-cxf-tomcat       camel-example-spark-rest-tomcat
camel-example-docs             camel-example-splunk
camel-example-etl              camel-example-spring
camel-example-ftp              camel-example-spring-javaconfig
camel-example-gae              camel-example-spring-jms
camel-example-gauth            camel-example-spring-security
camel-example-guice-jms        camel-example-spring-ws
camel-example-jdbc             camel-example-spring-xquery
camel-example-jms-file         camel-example-sql
camel-example-jmx              camel-example-sql-blueprint
camel-example-loadbalancing    camel-example-ssh
camel-example-loan-broker      camel-example-ssh-security
camel-example-management       camel-example-tracer
camel-example-mybatis          camel-example-twitter-websocket
camel-example-netty-http       camel-example-twitter-websocket-blueprint
camel-example-osgi             pom.xml
camel-example-osgi-rmi         README.txt
schueler@debian:~/Downloads/apache-camel-2.14.1/examples$ cd camel-example-
bash: cd: camel-example-: No such file or directory
schueler@debian:~/Downloads/apache-camel-2.14.1/examples$ cd camel-example-etl
schueler@debian:~/Downloads/apache-camel-2.14.1/examples/camel-example-etl$ ls
pom.xml  README.txt  src
schueler@debian:~/Downloads/apache-camel-2.14.1/examples/camel-example-etl$
```

5.) Die README.txt gibt Aufschluss, was zu tun ist.

```
  GNU nano 2.2.6               File: README.txt                          Modified

Extract Transform Load (ETL) Example
====================================

This example shows how to use Camel as an ETL tool
  http://camel.apache.org/etl.html

For a full description of this example please see
  http://camel.apache.org/etl-example.html

You will need to compile this example first:
  mvn compile

To run the example type
  mvn camel:run

You can see the routing rules by looking at the java code in the src/main/java
directory and the Spring XML configuration lives in
  src/main/resources/META-INF/spring
                              [ Read 47 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

6.) Maven ist auf unserer VM bereits installiert, zu Dokumentationszwecken gilt es folgende Zeile auszuführen, wenn Maven nicht vorhanden ist.

```
schueler@debian:~$ apt-get install maven
```

7.) Wie im Readme zu lesen, kompilieren wir das Example mit *mvn compile*. Alles wird erfolgreich abgeschlossen.

```
s/3.0/plexus-utils-3.0.jar
Downloading: http://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-comp
iler-manager/1.9.1/plexus-compiler-manager-1.9.1.jar
Downloading: http://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-comp
iler-javac/1.9.1/plexus-compiler-javac-1.9.1.jar
Downloaded: http://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-compi
ler-javac/1.9.1/plexus-compiler-javac-1.9.1.jar (14 KB at 202.6 KB/sec)
Downloaded: http://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-compi
ler-api/1.9.1/plexus-compiler-api-1.9.1.jar (21 KB at 282.3 KB/sec)
Downloaded: http://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-compi
ler-manager/1.9.1/plexus-compiler-manager-1.9.1.jar (5 KB at 60.9 KB/sec)
Downloaded: http://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils
/3.0/plexus-utils-3.0.jar (221 KB at 2567.1 KB/sec)
[INFO] Compiling 5 source files to /home/schueler/Downloads/apache-camel-2.14.1/
examples/camel-example-etl/target/classes
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 53.517s
[INFO] Finished at: Mon Feb 23 11:27:38 CET 2015
[INFO] Final Memory: 15M/241M
[INFO] ------------------------------------------------------------------------
schueler@debian:~/Downloads/apache-camel-2.14.1/examples/camel-example-etl$
```

8.) Dem Readme ist auch zu entnehmen, dass das Example mit *mvn camel:run* gestartet werden kann.

```
xample.etl.CustomerEntity, Body:<?xml version="1.0" encoding="UTF-8" standalone=
"yes"?>
<customer id="2">
    <userName>james</userName>
    <firstName>James</firstName>
    <surname>Strachan</surname>
    <city>London</city>
</customer>

2015-02-23 23:53:34,603 [.CustomerEntity] INFO  Tracer                         -
 ID-debian-57083-1424731991397-0-24 >>> (route2) setHeader[CamelFileName] --> fi
le://target/customers <<< Pattern:InOnly, Headers:{CamelEntityManager=org.apache
.openjpa.persistence.EntityManagerImpl@2fe79a2b, breadcrumbId=ID-debian-57083-14
24731991397-0-23, CamelFileName=james.xml}, BodyType:org.apache.camel.example.et
l.CustomerEntity, Body:<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<customer id="2">
    <userName>james</userName>
    <firstName>James</firstName>
    <surname>Strachan</surname>
    <city>London</city>
</customer>
```

Das Beispiel konnte erfolgreich ausgeführt werden.

Dokumentation des Source-Codes

&

Erstellung von Testfällen (laut Metaregeln)

Siehe beigefügten Source-Code.

Detaillierte Arbeitsaufteilung (Aufwandsabschätzung, Endzeitaufteilung)

Aufgabentrennung

Weinberger: Protokollierung der Inbetriebnahme des Beispiels, Dokumentation des Source-Codes, Erstellung von Testfällen (laut Metaregeln), Protokoll-Erstellung

Taschner: Identifikation und Beschreibung der EAI Patterns, Beschreibung der Funktionsweise von Apache Camel, Erstellung von Testfällen (laut Metaregeln), Protokoll-Erstellung

Aufwandabschätzung

Endzeitaufteilung

Fazit

Arbeitsdurchführung

Resultate

Abbildungsverzeichnis

Quellenangaben

[1] Enterprise Integration Patterns – File Transfer, Gregor Hohpe & Bobby Woolf, 18. Jänner 2015, *http://www.eaipatterns.com/FileTransferIntegration.html*, zuletzt aufgerufen am 24. Februar 2015

[2] Apache Camel: Pipes and Filters, The Apache Software Foundation, 16. März 2014, *http://camel.apache.org/pipes-and-filters.html*, zuletzt aufgerufen am 24. Februar 2015

[3] Apache Camel: Message Translator, The Apache Software Foundation, 16. März 2014, *http://camel.apache.org/message-translator.html*, zuletzt aufgerufen am 24. Februar 2015

[4] Apache Camel: Message Endpoint, The Apache Software Foundation, 16. März 2014, *http://camel.apache.org/message-endpoint.html*, zuletzt aufgerufen am 24. Februar 2015

[5] What exactly is Apache Camel?, Amr Mostafa, 18. Juli 2012, *http://stackoverflow.com/questions/8845186/what-exactly-is-apache-camel*, zuletzt aufgerufen am 24. Februar 2015