

# EAI

## Extract, Transform and Load (ETL) with EAI

20.02.2015

4BHIT

Melanie Göbel, Tobias Perny

## Inhaltsverzeichnis

Aufgabenstellung .....	2
Resources .....	2
Aufwand und Zeit .....	3
Schätzung und Realität .....	3
Zeitaufzeichnung Göbel.....	3
Zeitaufzeichnung Perny .....	3
Installation und Inbetriebnahme.....	4
Ausführen .....	4
Technologiebeschreibung .....	5
EAI.....	5
EIP .....	5
Insbesondere ETL .....	5
Verwendete Patterns .....	6
File Transfer .....	6
Message Translator .....	6
Message Endpoint .....	6
Quellen .....	7

## Aufgabenstellung

*"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

## Aufwand und Zeit

### Schätzung und Realität

Arbeit	Geschätzte Zeit (in min)	Zuständig	Benötigte Zeit (in min)
Ausführen von EAI	30	Göbel, Perny	30
Technologiebeschreibung	60	Göbel	40
Sourcecodedokumentation	60	Perny	70
Verwendete Patterns	120	Göbel	150
Dokumentation (Zeit, Deckblatt, Formatierung)	30	Perny, Göbel	20
Insgesamt	300 = 6 Stunden		310 = 6 std. 10min.

### Zeitaufzeichnung Göbel

Arbeit	Datum	Dauer (in min)
Ausführen von EAI, Dokumentation	15.02.2015	30
Technologiebeschreibung	22.02.2015	40
Verwendete Patterns	26.02.2015	150

### Zeitaufzeichnung Perny

Arbeit	Datum	Dauer (in min)
Sourcecodedokumentation	26.02.2015	70

# Installation und Inbetriebnahme

## Ausführen

Runterladen von Apache-Camel [1]

Entpacken mit *tar -xvf*

*Cd examples/camel\_example-etl*

Falls noch nicht vorhanden Maven installieren. *Apt-get install maven*

Lesen des README und ausführen von mvn compile und mvn camel:run

```
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

To stop the example hit ctrl + c
```

Fehler: Zwischen den Datensätzen eine NullPointerException:

```
1856 camel INFO [org.apache.camel.spring.Main.main()] openjpa.jdbc.JDBC - Connected to Apache Derby version 10.10 using JDBC driver Apache Derby Embedded JDBC I
rver version 10.10.2.0 - (1582446).
1815-02-20 14:19:55.340 [CustomerEntity] WARN JpaConsumer - Consumer Consumer[jpa://org.apache.camel.example.etl.CustomerEntity?consumeDelete=
false&consumeLockEntity=false&consumer.initialDelay=3000&delay=3000] failed polling endpoint: Endpoint[jpa://org.apache.camel.example.etl.CustomerEntity?consumeDele
te=false&consumeLockEntity=false&consumer.initialDelay=3000&delay=3000]. Will try again at next poll. Caused by: [java.lang.NullPointerException - null]
java.lang.NullPointerException
    at org.apache.camel.component.jpa.JpaConsumer$1.doInTransaction(JpaConsumer.java:95)
    at org.springframework.transaction.support.TransactionTemplate.execute(TransactionTemplate.java:131)
    at org.apache.camel.component.jpa.JpaConsumer.poll(JpaConsumer.java:92)
    at org.apache.camel.impl.ScheduledPollConsumer.doRun(ScheduledPollConsumer.java:187)
    at org.apache.camel.impl.ScheduledPollConsumer.run(ScheduledPollConsumer.java:114)
    at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:471)
    at java.util.concurrent.FutureTask.runAndReset(FutureTask.java:304)
    at java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.access$301(ScheduledThreadPoolExecutor.java:178)
    at java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.run(ScheduledThreadPoolExecutor.java:293)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1145)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:615)
    at java.lang.Thread.run(Thread.java:745)
1815-02-20 14:19:55.393 [ing.Main.main()] INFO SpringCamelContext - Route: route2 started and consuming from: Endpoint[jpa://org.apache.camel.example.
etl.CustomerEntity?consumeDelete=false&consumeLockEntity=false&consumer.initialDelay=3000&delay=3000]
```

## Technologiebeschreibung

### EAI

Enterprise Application Integration, kurz EAI, ist eine IT-Infrastruktur zur Kopplung von IT-Systemen z.B: ERP, CRM, SCM und andere betriebswirtschaftliche Systeme. [2]

Das Hauptprinzip von EAI ist eine loose Kopplung und beschreibt nicht wie das Softwareprodukt selbst auszusehen hat. [3]

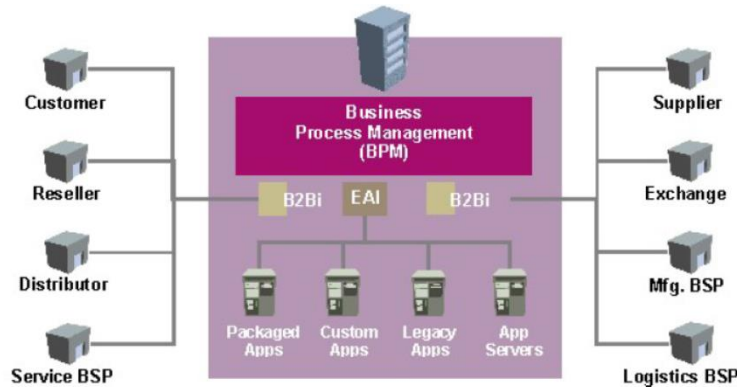


Abb.1: Übersicht EAI Aufbau [4]

### EIP

Durch den Einsatz von EIP (Enterprise Integration Patterns) können Probleme bereits im Vorhinein vermieden werden. Denn die meisten EIP bieten die "best practice" in diesem Bereich und bereits fertig funktionierende Beispiele.

File Transfer: Austausch von Files im standardisierten Format

Message Translator: Baustein der die Übersetzung übernimmt

Content Enricher: Lädt notwendige Daten einfach von einer anderen Stelle nach

Event Message: Eigener Message Typ

Selective Consumer: Konsumiert genau nur die Daten, die vom Empfänger gebraucht werden.

[3]

### Insbesondere ETL

ETL ist ein Prozess, bei dem Daten aus mehreren auch unterschiedlich strukturierten Datenquellen in einer Zieldatenbank vereinigt werden. ETL bedeutet einzeln:

- **Extract:** Extraktion der relevanten Daten aus verschiedenen Quellen
- **Transformation:** Transformation der Daten in das Schema und Format der Zielbank
- **Load:** Laden der Daten in das Data Warehouse

[5]



Abb.2: ETL Übersicht [6]

## Verwendete Patterns

### File Transfer

Ein File Transfer von XML Files in JPA und POJO findet in dem ETL-Beispiel statt.

### Message Translator

Die Klasse CustomerTransformer dient als Message Translator, da sie die Informationen in ein Customer speichert. Siehe Kommentar:

```
/**  
 * A transformation method to convert a person document into a customer entity  
 */
```

### Message Endpoint

Senden von PersonDocument zum JPA-Endpoint in EtlRoutes.java

## Sourcecode-Dokumentation

### PersonDocument.java

In diesem File wird eine Person erstellt.

Eine Person besitzt einen Usernamen, einen Vornamen, einen Nachnamen und eine Stadt.

### Main.java

Diese Klasse ist zum ausführen des Programmes da.

### CustomerEntity.java

In dieser Klasse wird eine Entity-Bean erstellt. Eine Entity-Bean ist eine Komponente von Enterprise JavaBeans. Mit Entity-Beans lassen sich dauerhafte Daten einer Datenbank, zum Beispiel Benutzer oder Adressen, darstellen. In diesem Fall wird ein Customer erstellt. Er besteht aus einem Usernamen, einem Vornamen, einem Nachnamen, einer Straße, Stadt, Postleitzahl und einer Telefonnummer. Dieser wird dann in einer Datenbank abgespeichert.

### CustomerTransformer.java

In dieser Klasse wird ein PersonDocument Objekt zu einem CustomerEntity Objekt umgewandelt.

### EtlRoutes.java

In dieser Klasse wird eine Route von src/data aus erstellt. Es wird der „noop“-Modus verwendet. Im noop Modus werden bearbeitete Files weder verschoben noch gelöscht. Wenn man das Programm also erneut startet arbeitet es dieselben Files nochmal ab.

## Quellen

[1] Installation Camel Apache

Verfügbar bei: <https://camel.apache.org/camel-2141-release.html> (zuletzt gesehen: 20.02.2015)

[2] Thorsten Horn – „EAI“

Verfügbar bei: <http://www.torsten-horn.de/techdocs/eai.htm> (zuletzt gesehen: 25.02.2015)

[3] Lukas Hörting, Martin Gawlowski – „Enterprise Application Integration“

Verfügbar bei: <https://elearning.tgm.ac.at/mod/resource/view.php?id=32658> (zuletzt gesehen: 25.02.2015)

[4] M.Borko/T.Micheler – „Architekturmodelle“

Verfügbar bei: <https://elearning.tgm.ac.at/mod/resource/view.php?id=29644> (zuletzt gesehen: 25.02.2015)

[5] Dr. Klaus Manhart - „BI-Datenmanagement (Teil 1): Datenaufbereitung durch den ETL-Prozess“

Verfügbar bei:

[http://www.tecchannel.de/server/sql/1746250/bi\\_datenmanagement\\_teil\\_1\\_datenaufbereitung\\_durch\\_den\\_etl\\_prozess/](http://www.tecchannel.de/server/sql/1746250/bi_datenmanagement_teil_1_datenaufbereitung_durch_den_etl_prozess/) (zuletzt gesehen: 25.02.2015)

[6] IT-Wissen - „Extraktion-Transformation-Lade-Prozess“

Verfügbar bei: <http://www.itwissen.info/definition/lexikon/ETL-extract-transfer-load.html> (zuletzt gesehen: 24.02.2015)