2014/15

ETL mit EAI

TGM WIEN xx | 22.02.2015

4BHIT

Adler/Karic

Inhaltsverzeichnis

[1. Identifikation und Beschreibung der EAI Patterns 2](#_Toc412368010)

[2. Funktionsweise von Apache Camel 2](#_Toc412368011)

[3. Inbetriebnahme des Beispiels 2](#_Toc412368012)

# Identifikation und Beschreibung der EAI Patterns

# Funktionsweise von Apache Camel

# Inbetriebnahme des Beispiels

Der erste Schritt ist das Downloaden vom Apache Camel von der offiziellen Website [*http://camel.apache.org/download.html*](http://camel.apache.org/download.html)

Hier wählt man die neueste Version für Unix/Linux –Systeme aus (in unserem Fall 2.14.1). Dieses Verzeichnis entpackt man nun im Zielordner.





Im Verzeichnis findet man mehrere txt-Dateien sowie drei Unterordner. Wir wollen das etl-example zum Laufen bringen daher ist der example-Ordner für uns von Bedeutung. Hier navigieren wir zum etl-example (siehe unten)

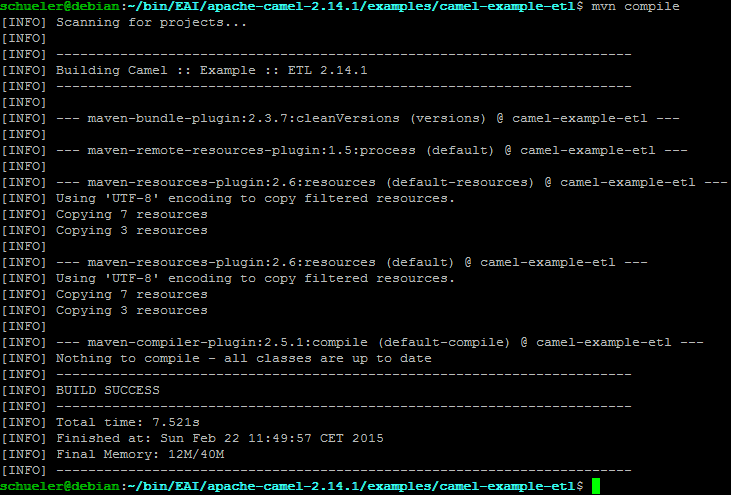


Wie wir im README lesen können müssen wir um das Beispiel zu starten zwei Befehle ausführen:

*mvn compile*

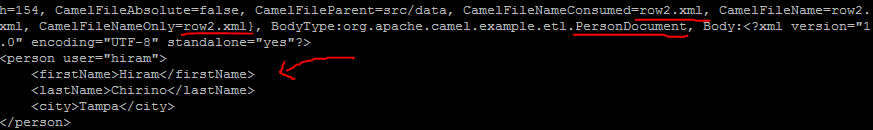
*mvn camel:run*

Damit man diese Befehle ausführen kann ist es aber nötig maven zu installieren (maven2 hat für uns nicht ganz funktioniert!). Wenn dies gemacht wurde können wird den ersten Befehl *mvn compile* ausführen. Wir sehen nun folgende Ausgabe:

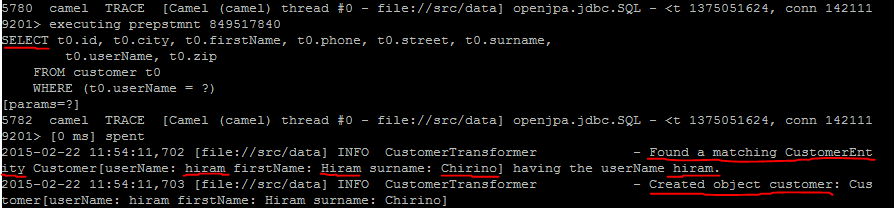


Wir erhalten ein BUILD SUCCESS und führen nun den Befehl *mvn camel:run* aus:

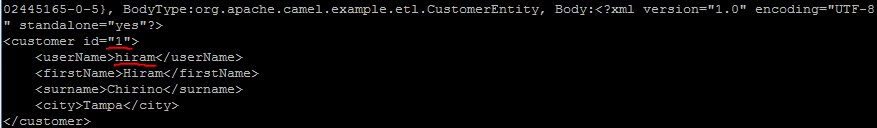
Wir sehen zunächst (nach mehreren Infos) die Ausgabe einer Person (als XML-Struktur). Man sieht auch von wo diese Daten kommen: row2.xml und wie sie intern repräsentiert werden nämlich als PersonDocument.



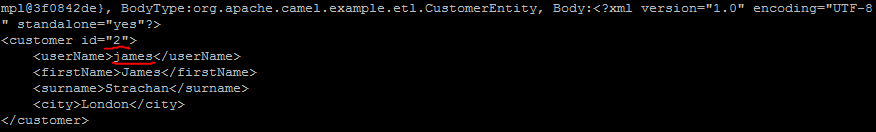
Danach folgt ein SELECT aus der Datenbank und anschließend findet der CustomerTransformer eine passende CustomerEntity (hiram). Zuletzt wird noch das Customer Objekt erzeugt. Derselbe Prozess passiert auch bei der 2.Person (James)



Hier sieht man dann die Ausgabe des neu erzeugten Customers hiram (id=1)



Hier die Ausgabe des 2. erzeugten Customers James (id=2)



Die beiden Personen (PersonDocument) wurden also durch einen CustomerTransformer in Customer gewandelt (CustomerEntity).