

# Übung: Transformation in HL7 FHIR mit Mirth-connect

**Vor Übungsbeginn:** 15 min

Docker Installation, z.B mit Docker Desktop

Mirth-connect Installation (benötigt Java)

Installationsanweisungen siehe GitHub:

<https://github.com/IMISE/MI-Lab-CSV-FHIR-transformation>

## Aufgabe 1: Mirth-connect

### 1. Anmeldung in Mirth-connect

Folgen Sie der Readme im GitHub, anschließend können Sie sich in Mirth-connect anmelden.

*user:* admin, *password:* admin

Nach der Erstanmeldung werden Sie aufgefordert, den User und das Passwort neu zu setzen. Folgen Sie den Anweisungen.

### 2. Erstellen von Channels

Gehen Sie zu *Channels* und unter Channel Tasks klicken Sie auf *New Channel*.

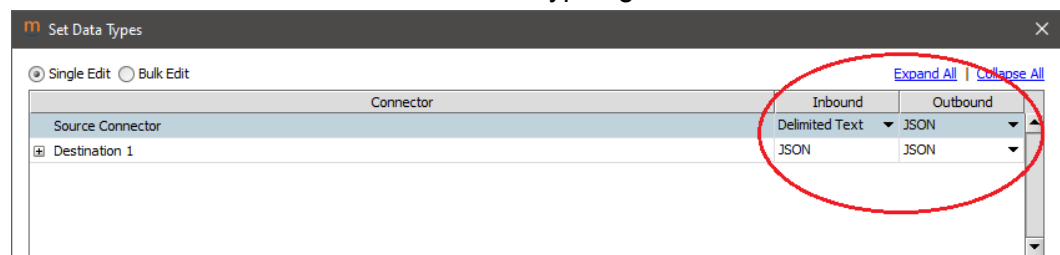
Fügen Sie jeweils einen Channel für Patient und Encounter hinzu.

### 3. Configuration der Channels

Editieren Sie die zuvor hinzugefügten Channels und führen Sie jeweils die folgenden Schritte durch:

#### 3.1. Data Types

Setzen Sie nun unter *"Summary"*->*"Channel Properties"* die Data Types der Source und Destination Connectors wie auf dem Bild. Mit dem Dropdown Pfeil können die Inbound und Outbound Types geändert werden.



**Erklärung:** Der Source Connector hat als Inbound eine CSV (Data Type: Delimited Text) und als Outbound die gemappte FHIR-JSON Nachricht. Die Destination halt als Inbound und Outbound jeweils die JSON Nachricht.

#### 3.2. Source Connector

Öffnen Sie den jeweiligen Channel und klicken Sie auf *Source*.

1. Setzen Sie den Connector Type auf "File Reader".
2. Ändern Sie unter "Source Settings" den Process Batch auf Yes. Dadurch werden alle Reihen der CSV ausgelesen und nicht nur der Header.

**Source Settings**

Source Queue: OFF (Respond after processing) ▼

Queue Buffer Size: 1000

Response: None ▼

Process Batch: ☒ Yes ☐ No

Batch Response: ☐ First ☒ Last

Max Processing Threads: 1

3. Ändern Sie unter *File Reader Settings* das *Directory*:

- Patient Channel: „/opt/connect/appdata/MII-Patient/“
- Encounter Channel: „/opt/connect/appdata/MII-Fall/“

**Erklärung:** Das Verzeichnis „/opt/connect/appdata/“ in der Docker Umgebung wird nach außen auf das Verzeichnis „Setup/mirth-connect/“ gemappt.

- Setzen Sie das *Filename Filter Pattern* auf „\*.csv“, damit nur CSV-Dateien eingelesen werden.
- Ändern Sie die *After Processing Action* auf *Move* und definieren Sie folgendes Verzeichnis für beide Channels:  
“/opt/connect/appdata/Processed/“

After Processing Action: Move ▼

Move-to Directory: /opt/connect/appdata/Processed/

Move-to File Name:

Error Reading Action: None ▼

Error in Response Action: After Processing Action ▼

Error Move-to Directory:

Error Move-to File Name:

channelName
channelId
DATE
COUNT
UUID
SYSTIME
originalFilename

**Erklärung:** Dadurch wird die CSV Datei nachdem Mirth-Connect es einliest in ein anderes Verzeichnis verschoben. Das verhindert ein multiples Einlesen des Datensatzes.

### 3.3. Destination Connector

Damit Mirth Connect die transformierten FHIR-Daten auf einen Hapi FHIR Server laden kann, müssen Sie eine die Destination für den FHIR Server anpassen.

- Klicken Sie auf *Destinations* und setzen Sie den *Connector Type* auf „HTTP Sender“
- In den *HTTP Sender Settings*:
  - Setzen Sie die *URL* auf „<http://hapi:8080/fhir/>“  
Denn wir befinden uns im Docker-Compose-Netzwerk.
  - Ändern Sie den *Content Type* zu „application/json“, damit die Daten im JSON-Format gesendet werden.
  - Setzen Sie den *Content* auf  
“\${JsonUtil.prettyPrint(\${message.rawData})}”, damit die Nachrichten formatiert als JSON gesendet werden.

**Wichtig:** Diese Schritte müssen für beide Channels (Patient, Encounter) durchgeführt werden!

## 4. Import von Transformers

Um die CSV Dateien nach FHIR zu transformieren, muss für die Channels ein Source Transformer hinzugefügt werden.

Klicken Sie jeweils im Channel unter dem Reiter "Source"->"Channel Tasks"->"Edit Transformer". Dies führt zum Source Transformer. Importieren Sie nun bereits bestehende Transformer die in der nächsten Aufgabe erweitert werden müssen.

Klicken Sie dafür unter "Transformer Tasks" auf "Import Transformer". Die zu importierten Transformer sind im Übungs Ordner unter "Material/Transformer/":

*MII-Patient Transformer.xml* und *MII-Encounter Transformer.xml*

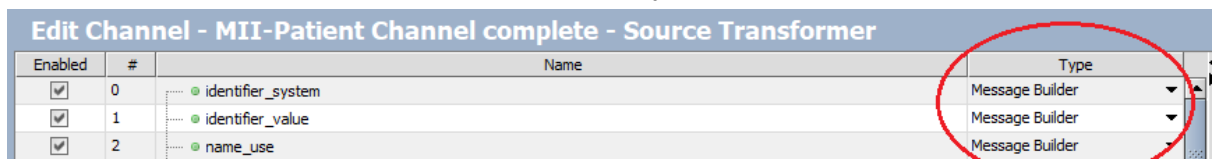
## 5. Configuration der Transformer

Im Folgenden werden Spalten aus dem CSV Datensatz auf die entsprechenden FHIR-Felder in einem JSON-Message Template gemappt. Das JSON-Template ist eine Vorlage für die Nachrichten an einen FHIR Server und wurde bereits im importierten Transformer mitgeliefert.

Navigieren Sie wieder im jeweiligen Channel zu "Source"->"Channel Tasks"->"Edit Transformer".

Fügen Sie im Source Transformer unter "Transformer Tasks"-> "Add new Step" jeweils einen neuen Schritt hinzu. Jeder Transformer Schritt mappt ein Attribut der CSV Datei auf ein FHIR-Feld. Es müssen für den Patienten und Encounter Transformer nur zwei Schritte hinzugefügt werden. Die anderen Schritte sind schon vorhanden.

**Wichtig:** Jeder neue Transformer Schritt muss den Typ "**Message Builder**" haben. Mit dem Drop Down Pfeil rechts können Sie den Type ändern.



Enabled	#	Name	Type
<input checked="" type="checkbox"/>	0	identifier_system	Message Builder
<input checked="" type="checkbox"/>	1	identifier_value	Message Builder
<input checked="" type="checkbox"/>	2	name_use	Message Builder

**Tipp:** Klicken Sie rechts auf den Reiter "Message Template" um das CSV-Template (oben) und FHIR Message-Template (unten) anzuschauen.

Unter dem Reiter "Message Trees" sehen Sie die Templates als Baumstruktur und können mit dem Plus diese aufklappen und per Drag and Drop in die entsprechenden Mapping Felder ziehen.

1. Für den *Patient Transformer* mappen Sie folgendes nach FHIR, indem Sie jeweils einen neuen Schritt hinzufügen für:
  - a. Name\_family
  - b. Gender
2. Für den *Encounter Transformer* mappen Sie folgendes nach FHIR, indem Sie jeweils einen neuen Schritt hinzufügen für:

- a. Aufnahmeummer

**Tipp:** Hier ist die Aufnahmeummer im FHIR Message Template identifier.value.

- b. Referenz auf Subject (Patient)

**Tipp:** Da jeder Fall einen Patienten referenziert soll die Referenz im Mapping Feld mit:

'Patient?identifier=' + msg['row'][0]... beginnen. Wichtig ist hier, dass nur 'Patient?identifier=' in einfachen Anführungszeichen geschrieben ist.

## 6. Deployen der Channels und laden der Daten

Speichern und deployen Sie die Channels. Kopieren Sie danach die entsprechenden CSV-Dateien der zu mappenden Daten (*MIMIC IV Patients.csv* und *MIMIC IV Fall.csv*) in die in Aufgabe 1.3a) erstellten Ordner.

**Wichtig:** Es ist wichtig, dass die Patienten zuerst geladen werden, da die Fälle eine Referenz auf vorhandene Patienten benötigen.

**Hinweis:** Durch das Deployen der Channels führen Inputs in die von Mirth-Connect 'Source Connector' überwachten Ordnern direkt zum Versenden von POST-Requests an den unter 'Destination Connector' verknüpften FHIR Server. Versendete Requests und ggf. Fehlermeldungen können unter 'View Messages' angesehen werden. Das erfolgreiche Übermitteln der Daten wird in Aufgabe 2 durch ClinFHIR überprüft.

## Zusatz: FHIR-Daten leeren

Wenn es Fehler gab oder der Datensatz mehrmals eingelesen wurde kann man die Daten des FHIR-Server leeren.

Dies geht über den Terminal in dem Ordner "Setup/":

1. docker-compose down
2. docker volume rm setup\_hapi\_data
3. docker-compose up -d

Dadurch wird die HAPI-FHIR-Datenbank komplett zurückgesetzt, einschließlich aller Patienten und Encounter.

**Wichtig:** Das startet ebenfalls Mirth-Connect neu dadurch muss man den Launcher erneut öffnen und sich wieder anmelden. Alle gespeicherten Mirth-Connect Daten bleiben erhalten.

## Aufgabe 2: ClinFHIR

1. Öffnen Sie ClinFHIR lokal im Browser: <http://localhost:8000/launcher.html>

Fügen Sie den Hapi FHIR (Version R4) Server hinzu:

URL: <http://localhost:8090/fhir/>

Add new server

Name	<input type="text" value="Mi-Lab Hapi"/>	The name of the server (will be displayed in server selection dialogs)
Url	<input type="text" value="http://localhost:8090/fhir/"/>	The full url to the server (including port). Include a trailing /
FHIR Version	<input type="text" value="R4"/>	The version of FHIR that the server supports
Support \$everything	<input checked="" type="checkbox"/>	Set true if this server supports the \$everything operation against Patient (Needed for Patient Data viewer)
Is Terminology Server	<input type="checkbox"/>	Set true if this server supports Terminology services (especially \$expand)
4.0.1		

[Cancel](#) [Add](#)

2. Sehen Sie sich mit 'Patient Viewer' die hinzugefügten Patienten und Encounter an und beantworten Sie die folgenden Fragen:

- 2.1. Wie alt ist der Patient mit ID 10015860 (ID 58)?
- 2.2. Wie viele Fälle gibt es zu dem Patienten mit dem Identifier 10003400 (ID 2)?
- 2.3. Ist der Patient mit ID 10018845 verstorben? Lösung: Ja

3. Sehen Sie sich mit 'clinFHIR **Server Query**' die hinzugefügten Patienten und Encounter an und beantworten Sie die folgenden Fragen:
  - 3.1. Wie viele Patienten sind weiblich?
  - 3.2. Wie viele Patienten sind bereits verstorben?
  - 3.3. Wie viele Fälle gehören zu den bereits verstorbenen Patienten?