050033 PR

Praktikum Medizinische Informationssysteme

*Benedek Gal – a1268262*

Universität Wien, Informatik

**Abstract.** Das Ziel von Praktikum ist die Erkenntnis von Fhir Open Server Implementationen, die Erstellung einer Fhir-Umgebung, das Testen der Umgebung mit Random-Daten am Windows 7.

**Keywords:** Fhirbase, Fhir Hapi, Server, Client, RESTful, API, Ressource, Patient, Encounter

1. Einleitung

Fhir ist die nächste Generation von Standard Framework von HL7 Organisation. Es kombiniert der Beste Merkmalen von HL7 Version 2, Version 3 und CDA Produkte mit den neuesten Web Standards mit starkem Fokus an Umsetzbarkeit.

1. **Ursprüngliches Ziel**:

Populationsauswertungen machen (Aggregierte Information sammeln) mithilfe eine der Fhir-Implementierungen. Krankenhausaufenthalt der Bevölkerung zu visualisieren.

1. **Ziel des Praktikums:**

* eine der Fhir-Entwicklungsumgebungen wählen, aufsetzen, kennenlernen,
* FHIR-Server aufbauen, mit Testdaten befüllen,
* Ressourcen wählen: **Patient** (Altersgruppe: 0-14, 15-29, 30-44, 45-59, 60-74, 75-89, Geschlecht), Encounter, Krankenhaus, **Bundesland** (9), Bezirk

1. Fhirbase in Docker

Fhirbase ist an PostgreSQL aufgebaute Extension und benötigt PostgreSQL v9.4 oder höher. Erstes Mal beschäftigte ich mich mit Fhirbase unter Docker, leider konnte ich die Umgebung mit der Anleitung an Webseite dennoch nicht zum Laufen bringen.

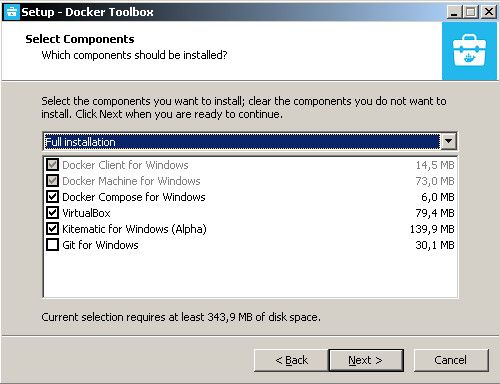
**2.1 Installation von Docker am Windows 7**

Ich versuchte die Umgebung durch Docker zu installieren. Das ist schon in Win7 bisschen kompliziert.

Die offizielle Anleitung sagt, dass man am Win7 Microsoft Hyper-V braucht. Falls man diese Richtung wählt dann gelangt man zu einem Streit ob Win7 tatsächlich Hyper-V nutzen oder nicht nutzen kann. Nach ausgiebigen Recherchen am Google entschied ich mich nicht an Hyper-V zu suchen, weil Docker Packet sowieso VirtualBox beinhaltet. Man arbeitet entweder mit Linux oder bleibt beim Win7 und sucht Umwege.

Docker ist nicht einfach am Windows zu nutzen. Docker ist an Linux geschrieben [[1]](#footnote-1). Im Endeffekt braucht es also ein Linux System zu laufen. Am Windows Systems geht es leicht mit Linux VM. Man meldet sich an dem Virtual Maschine an und nutzt Docker als man am Linux arbeitete.

**2.1 Installation von Docker Toolbox**

Die offizielle Anleitung für Win7 bietet ein Docker Toolbox an VirtualBox aber nachdem Hochfahren des Servers konnte ich nicht Zugang zu Sever aufbauen.

Die Anleitung von GitHub ist klar formuliert, wurde durchgemacht, aber in Praxis der Zugang zum Server ist nicht gelungen.

Ich versuchte mithilfe Kitematic, die am Docker Server stehende offizielle Images zu starten. Die funktionierten, aber den selbst aufgebaute Fhirbase Server konnte ich nicht kontaktieren.

Was den Aufbau eines Servers angeht, gibt es viele Wege, aber im Endeffekt ging was immer schief.

1. Fhirbase in Vagrant

Für Fhirbase aufzubauen bietet sich noch die Möglichkeit mithilfe Vagrant und Virtualbox zu starten. Um die Umgebung von Fhirbase installieren zu können, braucht man die folgenden Anwendungen zu installieren.:

* Git
* VirtualBox v5.1.4
* Vagrant v1.8.5
* PostgreSQL v9.5

Die Reihe der Installation ist nicht von Bedeutung. Alle obigen Anwendungen kommt mit ausführbare Installation-file. Die Installation deswegen schnell und einfach.

**3.1 Fhirbase-plv8 Settings-änderung**

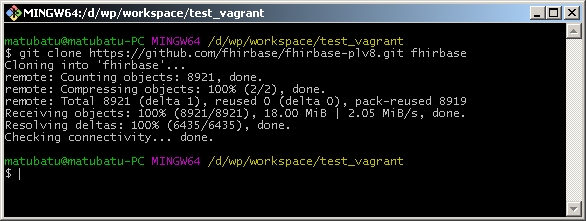
Die vollständigen Installationsschritte werden im nächsten Absatz detailliert geschrieben. Der Unterschied hier ist nur die Beschaffung von Fhirbase, die jetzt mithilfe Git geklont wird.

Git-Bash starten im Workspace wohin Fhirbase-plv8 herunterladen wird.

1. Fhirbase von GitHub klonen:

*git clone https://github.com/fhirbase/fhirbase-plv8.git fhirbase*

Oder einfach von GitHub als ZIP herunterladen.



1. In *fhirbase/vagrant/provision* Folder: beide Datei öffnen und die folgende Änderungen vornehmen:



* **provision-dependencies.sh bearbeiten:**

Die letzte Zeile: *pgxn install plv8 || exit 1* mit Folgenden ersetzen:

*pgxn install plv8=1.4.4 || exit 1*

* **provision-** **environment.sh bearbeiten:**

die folgenden Zeilen mit # auskommentieren:

*# echo "curl --location* [*https://github.com/fhirbase/fhirbase-plv8/releases/download/v1.4.0.0/fhirbase-*](https://github.com/fhirbase/fhirbase-plv8/releases/download/v1.4.0.0/fhirbase-)

*1.4.0.0.sql.zip| funzip | psql -d fhir" \*

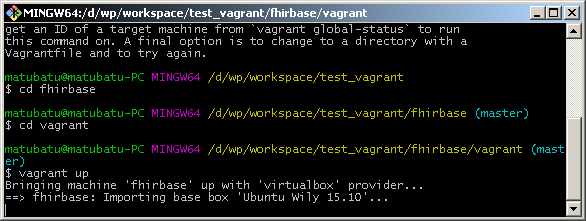
*# | sudo -u postgres sh || exit 1*

Falls man diese Bearbeitung nicht macht, kommt zu einem Error beim Starten des Virtual Machines.

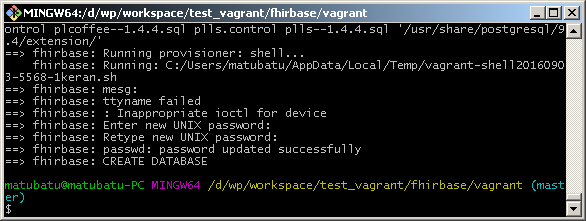
Das kann man leicht beheben nach der Änderung mit *vagrant provision* Befehl.

3. In Git-Konsol fhirbase/vagrant Folder öffnen, wo das Vagrantfile ist:

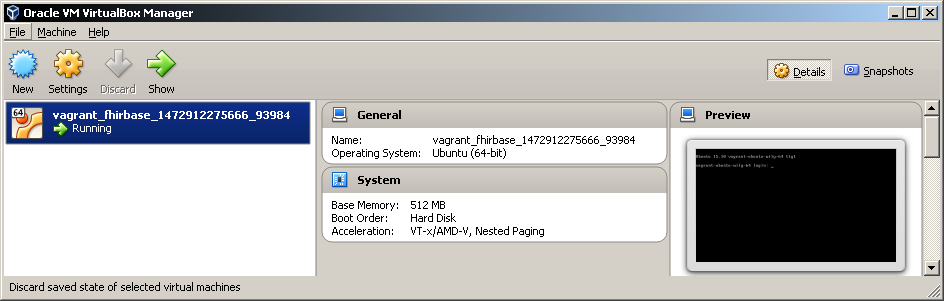
*vagrant up*



4. Eine Ubuntu Virtuelle Maschine wird in VirtualBox erzeugt, an dem ein PostrgreSQL Datenbank mit allen Abhängigkeiten zur Verfügung steht.

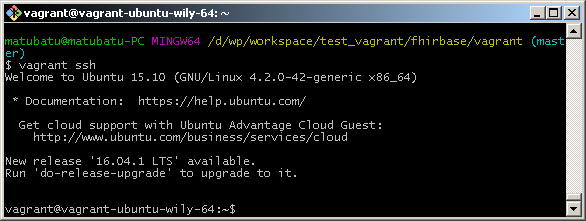


In VirtualBox Manager sieht man auch die Ubuntu Virtuelle Maschine:



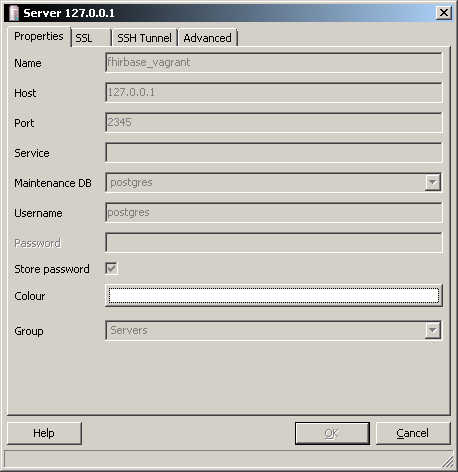
5. Mit *vagrant ssh Befehl:*

SSH-Konsol von Virtuelle-Maschine wird gestartet.



Fhirbase ist ein PostgreSQL Extension für Speicherung und Abfragung von FHIR-Resourcen. Das Interaktion mit Fhirbase ist mit pgAdmin III der einfachste.

6. pgAdmin III öffnen und ein neuer Server schaffen:

Danach kann man den Zugang zu Server testen.

Die Datenbank aufzufüllen, nutzt man am besten ein Konsol Fester, wo die Tabellen von Fhir mithilfe Script-dateien eingelesen wird.

7. Die SQL-Script-Dateien können von https://mias.meduniwien.ac.at/owncloud/index.php/s/ict8kElMfxmC8pv heruntergeladen werden. Passwort: *fhirbase*.

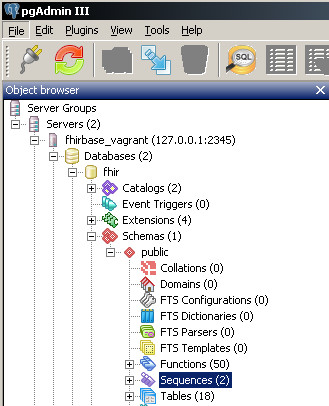
8. Konsol öffnen in *C:\Program Files\PostgreSQL\9.5\bin>*

Folgender Befehl eingeben:

*psql -d fhir -h 127.0.0.1 -p 2345 -U postgres*

Bekommt man jetzt die Fhir-DB Prompt wo der Path der jeweiligen SQL-Scriptfile eingegeben wird:

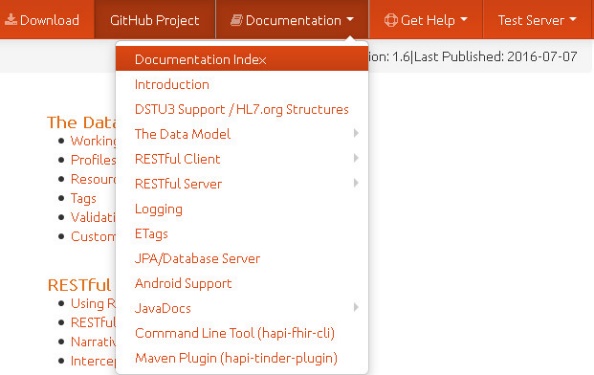
Die drei Script-Dateien wurden ins Scripts Ordner kopiert:

*\i 'c:/Program Files/PostgreSQL/9.5/scripts/fhirbase-1.4.0.0\_c\_utf\_1.sql'*

*\i 'c:/Program Files/PostgreSQL/9.5/scripts/fhirbase-1.4.0.0\_c\_utf\_2.sql'*

*\i 'c:/Program Files/PostgreSQL/9.5/scripts/fhirbase-1.4.0.0\_c\_utf\_3.sql'*

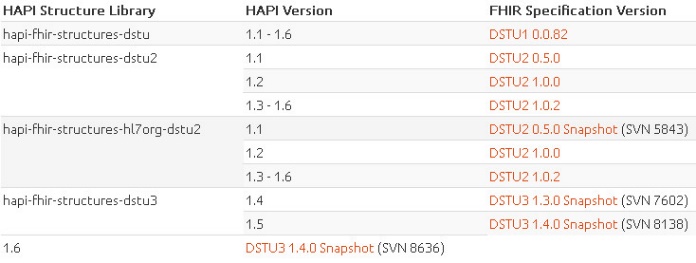
Das Ergebnis von den Scripten sind 50 Funktionen und 20 Tables:

1. FHIR Hapi

Das Hapi Fhir ist eine mächtige Bibliothek, damit Fhir-messaging in eigenen Anwendungen implementieren zu können. Das Fhir Hapi Dokumentation in einem Menu erreichbar und gut benutzbar.

Die Fhir Hapi Umgebung wird mithilfe Maven aufgebaut. Das Version von HAPI 1.5 wurde gewählt:

* **Structure Library:**   
  hapi-fhir-structures-dstu2
* DSTU2 1.0.2

****Die benötigten Bibliotheken und Dependencies wurde mithilfe Maven gebaut.

Das Project am GitHub:

<https://github.com/matubatu/Praktikum_FhirHapi>

**1.** **Introduction, erste Schritte mit Hapi**

In der Anleitung gezeigten Beispielen waren leicht erzeugbar und gut annotiert:

<http://hapifhir.io/doc_intro.html>

* *Parsing a resource from a String*
* *Encoding a Resource to a String*

**2. RESTful Client:**

*Generic Client:*

Mithilfe Generic-Client kann man Zugang zum Testserver bekommen und bestehende Testdaten abfragen.

Link des Testservers: <http://fhirtest.uhn.ca/> (siehe myProject/GenericClient.java)

1. **Erzeugen von Patient mit Random-daten:**

Das Erzeugen von Patient mit Random-daten mithilfe Dokumentation ist leicht nachvollziehbar.

1. https://www.ibm.com/developerworks/community/blogs/jfp/entry/Using\_Docker\_Machine\_On\_Windows?lang=en [↑](#footnote-ref-1)