Special Instructions for Deployment

CDC Post-Discharge Stroke Data Reporting App

Team Health Informaniacs

Fayaz Adam
Vinh Nguyen
Anirrudh Saddi
Gurpreet Singh

Source: https://github.gatech.edu/gt-hit-fall2016/CDC-Post-Discharge

Table of contents

I. Overview

CDC Post-Discharge Stroke Data Reporting App (aka PDS App) is deployed as two Java JARs. They can be deployed on the same server or different servers if there is accessible network path between the two. This app also works in conjunction with FHIR server.

- **triggerservice** - For querying FHIR resources asynchronously for stroke patients

- **strokeservice** - REST interface and web application for hosting survey and saving to FHIR

To deploy the sandbox environment, begin by setting up the FHIR sandbox environment using the instructions located here: https://github.com/smart-

on-fhir/installer. For the purposes of our GaTech deliverable, a link with a ready VirtualBox appliance will be provided in the shared folder below.

https://drive.google.com/drive/folders/0Bzoh1sBtLYo_TEdLTGJCRHEydUk

Once the FHIR sandbox environment is downloaded, which runs on Ubuntu 14.04, the following steps are required.

II. Set up environment and synthetic data

1. If JDK version is not 1.8 or later, install JDK1.8

```
sudo add-apt-repository ppa:webupd8team/java -y
sudo apt-get update
sudo apt-get install oracle-java8-installer
```

Set JAVA_HOME:

```
export JAVA_HOME=/usr/lib/jvm/java-8-oracle
```

2. Setup the backend PostgreSQL database for the application

   a. Create DB User Id and Password:

```
student@student:~$ sudo -u postgres psql
[sudo] password for student:
psql (9.3.14)

postgres=# ALTER USER postgres PASSWORD 'password';
ALTER ROLE
```

   b. Create StrokeApp Database:

```
student@student:~$ sudo -u postgres psql
[sudo] password for student:
psql (9.3.14)
```

```
postgres=# CREATE DATABASE strokeapp;
```

3. Turn off OAuth2 on SMART-on-FHIR and start FHIR service

   a. Edit main.yml:

```
cd ~/installer/provisioning/roles/common/defaults/
sudo vi main.yml
# Specific settings for the SMART-on-FHIR server authorization behavior
# Default: fhir_server_use_auth: true. Change true to false to turn off OAuth
fhir_server_use_auth: false
```

   b. Save changes in main.yml and restart the FHIR service:

```
sudo ansible-playbook -c local -i 'localhost,' -vvvv \ smart-on-fhir-servers.yml
```

4. Load synthetic data to PDS App. This step uses a custom script for loading sample patients who contain Encounters and/or Conditions which meet the stroke criteria (using given ICD-10 codes). See readme in "generated_data"

```
cd ~/CDC-Post-Discharge/generated_data
sudo su root ./addStrokePatients.sh
```

5. Configuration Parameters for **strokeservice**. Verify these values match database configurations from step 2.

Property file:

```
~/CDC-Post-Discharge/strokeservice/src/main/resources/application.properties
```

```
# ================================
# [ Server Configuration Section ]
# ================================

server.port = 8888

# ==================================
# [ Database Configuration Section ]
# ==================================

spring.jpa.database = POSTGRESQL
spring.jpa.hibernate.ddl-auto = update

spring.datasource.platform = postgres
spring.datasource.url                                          =
jdbc:postgresql://localhost:5432/strokeapp
spring.datasource.username = postgres
spring.datasource.password = password

# ====================================
# [ Application Configuration Section ]
# ====================================

admin.password = password
fhirService.baseUrl = http://localhost:9080/
```

6. Configuration Parameters for **triggerservice**. Verify these values match database configurations from step 2.

Property file:

```
~/CDC-Post-
Discharge/triggerservice/src/main/resources/application.properties

# ================================
# [ Server Configuration Section ]
# ================================
server.port = 8887

# ========================================
# [ Stroke Service Configuration Section ]
# ========================================
strokeService.baseUrl = http://localhost:8888/
```

```
# ======================================
# [ FHIR Service Configuration Section ]
# ======================================
fhirService.baseUrl = http://localhost:9080/
initialFetchDays = -1
```

III. Starting Application

The PDS App can be started in two ways, using maven (requires maven to be installed) or with java jars.

To start the application with maven:

1. Run **strokeservice**

```
cd ~/CDC-Post-Discharge/strokeservice
mvn spring-boot:run
```

Optionally, pass the -Dadmin.password=<password> property to specify the password for the 'admin' user to log in to the Admin UI. For example:

```
cd ~/CDC-Post-Discharge/strokeservice
mvn spring-boot:run –Dadmin.password=123456
```

2. Run **triggerservice**

Note: strokeservice is REQUIRED to be running before starting triggerservice

In a new terminal:

```
mvn spring-boot:run
```

Optionally, pass the -DinitialFetchDays=<days> property to specify the number of days back to use when querying patients who have been discharged for stroke conditions.

```
cd ~/CDC-Post-Discharge/triggerservice
mvn spring-boot:run -DinitialFetchDays=90
```

To start the application with the JAVA jars:

1. Run **strokeservice**

```
cd  ~/CDC-Post-Discharge/strokeservice
java -jar ./target/strokeservice.0.1-SNAPSHOT.jar
```

OR

```
cd  ~/CDC-Post-Discharge/strokeservice
java -jar ./target/strokeservice-0.0.1-SNAPSHOT.jar \
-Dadmin.password=123456
```

2. Run **triggerService**

Note: strokeservice is REQUIRED to be running before starting triggerservice

In a new terminal:

```
cd  ~/CDC-Post-Discharge/triggerservice
java -jar ./target/triggerservice.0.1-SNAPSHOT.jar
```

OR

```
cd  ~/CDC-Post-Discharge/triggerservice
java     -jar     ./target/triggerservice.0.1-SNAPSHOT.jar     -
DinitialFetchDays=90
```

To start the application with the JAVA jars:

1. Run **strokeservice**

```
java -jar ./target/strokeservice.0.1-SNAPSHOT.jar
```

OR

```
java -jar ./target/strokeservice-0.0.1-SNAPSHOT.jar \
-Dadmin.password=123456
```

2. Run **triggerService**

```
cd  ~/CDC-Post-Discharge/triggerservice
java -jar ./target/triggerservice.0.1-SNAPSHOT.jar
```

OR

```
cd  ~/CDC-Post-Discharge/triggerservice
java    -jar    ./target/triggerservice.0.1-SNAPSHOT.jar    -
DinitialFetchDays=90
```

IV. Verify Applications Running

1. **strokeservice**, which hosts the web application, will listen on port 8888. Verify the application is up and running:
   http://localhost:8888/

2. **triggerservice**, runs asynchronously in the background and queries the FHIR server nightly at 1 AM.

V. Running the Application

1. In a web browser (Chrome/Firefox), navigate to URL "http://localhost:8888" where our stroke service resides.

2. Login in as admin using the credentials
   user: admin

password: password (default) or the password entered while initializing the stroke service.

3. If there is a questionnaire present, logout. If not, click on the upload button, navigate to the questionnaire (CDC-Post-Discharge\strokeservice\src\test\resources\data\stroke-questionnaire.json) to be used for collecting 30 days post discharge questionnaire. Logout once the upload is complete.

4. Now, login as a survey administrator using the credentials
   username: nurse_jane
   password: password

   You will now see a list of patients assigned to Nurse Jane. The questionnaire column will display a question icon for those patients for whom the questionnaire needs to be completed or the response column displays the response ID of responses completed for the patients.

5. You can sort the table on patient first name or days since discharge by clicking on the column header.

6. To complete the questionnaire, click on the question icon next to the patient, it will route you to a questionnaire page. Enter as many applicable fields as you would like and at the top, click the Submit button to submit it, or Cancel button to return home.

7. Once submitted, the app will route you back to home with a response ID next to the patient for whom the questionnaire has just been completed.

a. To view the JSON response, you can click on the response ID. Here you can download the response either as a .json file or a .csv file by clicking either of the Download button.
b. You can click on the response ID again to hide the response.

8. Logout