

# Community Acquired Pneumonia (CAP) Clinical Stability Predictor Manual

Georgia Institute of Technology CS 6440 - Intro to Health Informatics Fall 2017

## Mentor

Dr. Lorenzo Di Francesco

## Team No Pressure

Kishan Bhoopalam - PM

Michael Fahey - QA

Byron Ruth - Developer

Raj Vansia - Developer

## GitHub Repository

<https://github.gatech.edu/gt-hit-fall2017/Community-Acquired-Pneumonia-Clinical-Stability-Indicator>

## Introduction

Community acquired pneumonia (CAP) is defined as pneumonia not acquired in a hospital or a long-term care facility and affects 5 to 6 million people a year in the United States. Symptoms include cough, fever, chills, fatigue, dyspnea, rigors and pleuritic chest pain. Influenza and pneumonia is the eight most common cause of death in the United States. As such, we set out to develop a FHIR compliant clinical decision support application in order to assist clinicians in determining when a patient is determined to be clinically stable enough to be discharged from the hospital. The results of our research and development is called CAP Clinical Stability Predictor. CAP Clinical Stability Predictor uses the FHIR API to retrieve patient information from a FHIR server and uses the vital signs of the patient to offer clinically actionable information to clinicians about the patient. From the information provided from CAP Clinical Stability Predictor, the clinician is able to make an informed decision on the treatment plan for each patient.

## Setup

The following section will guide you through the process of setting up and running the CAP Clinical Stability Predictor application.

## Prerequisites

CAP Clinical Stability Predictor runs on Docker and therefore [Docker](#) and [Docker Compose](#) are required and must be installed prior to running the application. Docker can be installed by visiting

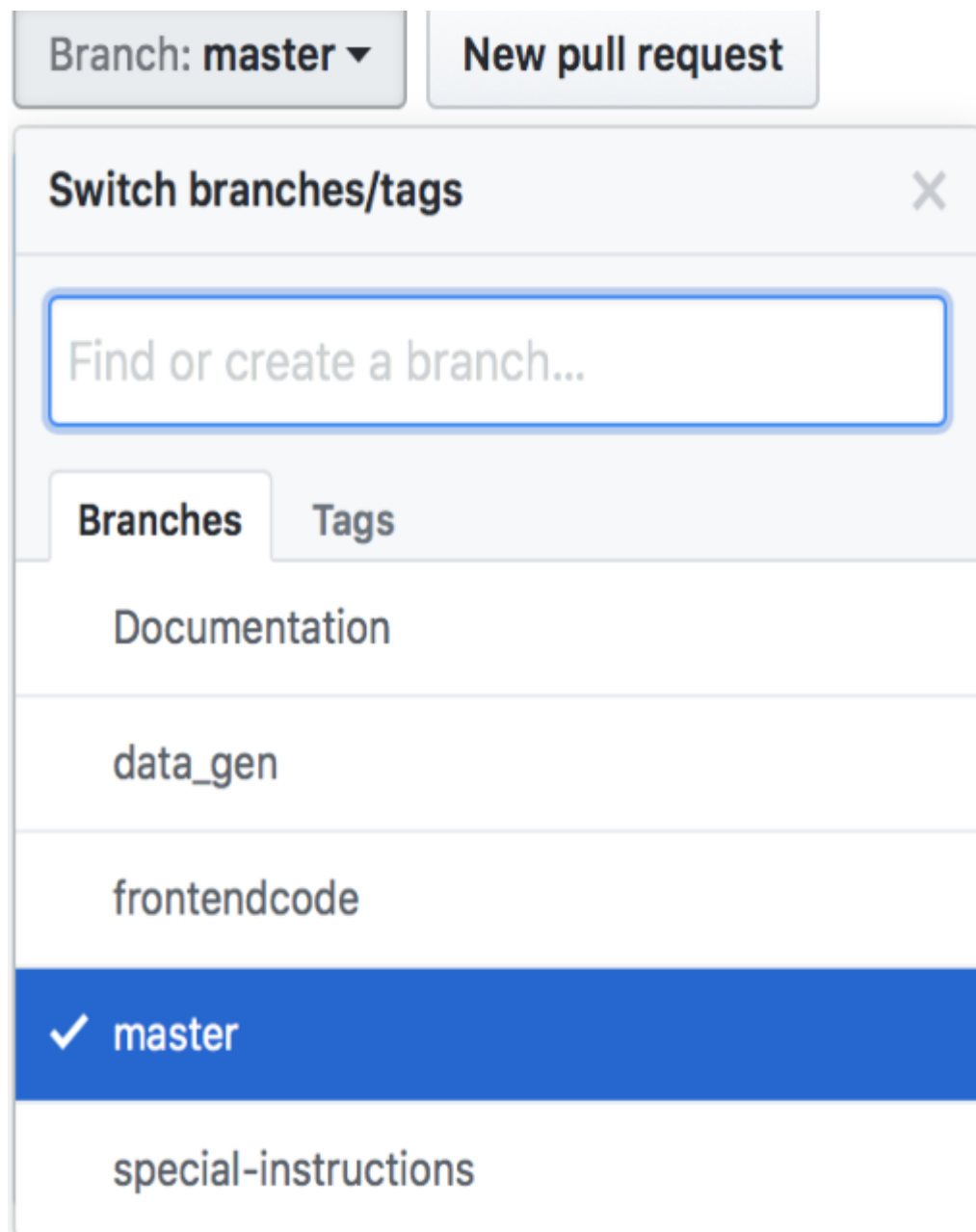
<https://docs.docker.com/engine/installation/#supported-platforms> and following the on page instructions. Additionally, Docker Compose can be installed by visiting <https://docs.docker.com/compose/install/> and following the on page instructions. For more information on Docker visit <https://www.docker.com/>. The versions tested against during development were Docker 17.09.0 and Docker Compose 1.16.1, however most recent versions should work properly. For Docker troubleshooting information visit <https://docs.docker.com/toolbox/faqs/troubleshoot/>.

## **Application Setup**

Once Docker is successfully installed on your system the next step in setting up the application is to download the Docker images. The follow section will discuss the steps necessary in order to download and run the application.

### **1. Downloading the Repository and Docker Images.**

- a. To download the Docker images along with the rest of the repository used by the application, visit <https://github.gatech.edu/gt-hit-fall2017/Community-Acquired-Pneumonia-Clinical-Stability-Indicator>. Note: you may be prompted to login during this step. Follow the on screen directions to login.
- b. On the left hand side, ensure that next to branch 'master' is selected. If it does not show master, select the drop down box and choose master from the list.



c. Next, select the 'clone or download' button located on the right hand side.



d. Select the 'Download zip' button to begin downloading the repository and Docker images. Note: this may take some time to download depending on your internet speeds.

e. Lastly, unzip the repository to be used in the next steps.

Alternatively, you can use the command line interface to clone the repository from GitHub using

```
git clone  
https://github.gatech.edu/gt-hit-fall2017/Community-Acquired-Pneumonia-Clinical-Stability-Indicator.git
```

## 2. Pulling the Latest Docker Images.

The previous step should download the latest Docker images from the repository, however, it is always a good idea to pull for the latest images.

- a. To do so, open a command line and navigate to where the repository was downloaded to in the previous step.
- b. Once you are inside of the Community-Acquired-Pneumonia-Clinical-Stability-Indicator directory, navigate to the directory called "Final Delivery"
- c. Now, run the following command from within the Final Delivery directory. This will take a few minutes to download and setup the latest images.

```
docker-compose pull
```

## 3. Running the Docker Images.

Now that the latest Docker images have been downloaded, the next step is to run the images.

- a. In order to run the Docker images you must be in the Final Delivery directory under Community-Acquired-Pneumonia-Clinical-Stability-Indicator (if you performed step 2 you should already be in this directory).
- b. Once you have navigated to the Final Delivery directory, run the following command

```
docker-compose up
```

- c. On first run, this will pull four images from Docker Hub which may take several minutes depending on your Internet connection speed.

```
Final Delivery — docker-compose ◀ docker-compose up — 80x24
0b6ff42c1d69: Pull complete
f96a0cfcc2a2: Pull complete
3b91661b4eaa: Pull complete
d86db4880b39: Pull complete
d4e82f379a6c: Pull complete
Digest: sha256:0ebf40b11bd3017ded0f947728442f13b3839c9df54ebd1c6ad94df0ced5c95c
Status: Downloaded newer image for bruth/np-frontend:latest
Pulling postgres (bruth/np-fhir-postgres:latest)...
latest: Pulling from bruth/np-fhir-postgres
9f0706ba7422: Downloading [=====>]
 27.56MB/52.61MBwnload complete
945954562465: Download complete
4fa64a58ef1a: Download complete
 6.963MB/7.114MBwnload complete
10e84c2461e4: Download complete
22e2459f9d56: Download complete
c5037dff1690: Downloading [=====>]
 20.79MB/43.29MBwnload complete
3dd32c7a376f: Download complete
3e120e164d93: Download complete
e3b494ed0617: Download complete
34cfe40f51e3: Download complete
a544950ef465: Download complete
 5.381MB/5.432MB
```

#### 4.Viewing the Application.

Once you have the Docker images successfully running, the next step is to bring up the user interface in order to view the application.

- a. Start by opening up a web browser, we recommend using either Safari or Chrome, and navigate to <http://localhost:5931>. Note: it may take a few moments to load depending on if step 3 has completed. If you see "Loading" for more than a few minutes, refresh your browser.
- b. You should now see the patient list in your browser. Refer to the section **Using the Application** for further details on how to interact with the application.

CAP Clinical Stability Predictor					
<div> <div>Amy V Shaw</div> <div>DOB: 2007-03-20 MRN: 12081332</div> <div>Clinically Not Stable</div> </div>					
110/63 mmHg	81	34	38 C	100%	
BP	HR	RR	Temp	SpO2	
Length of Stay: 4 days					
<div> <div>Robert Smith</div> <div>DOB: 1998-05-20 MRN: 21903927</div> <div>Clinically Stable</div> </div>					
113/75 mmHg	72	21	38 C	100%	
BP	HR	RR	Temp	SpO2	
Length of Stay: 5 days					
<div> <div>Jon Doe</div> <div>DOB: 2001-01-13 MRN: 50239183</div> <div>Clinically Not Stable</div> </div>					
122/69 mmHg	102	40	39 C	100%	
BP	HR	RR	Temp	SpO2	
Length of Stay: 1 day					

## Troubleshooting

If you are having difficulties running the application, there are a few troubleshooting steps you can try. Some common errors and troubleshooting steps are discussed below.

### 1.Docker Not Running

When attempting to run the application if receive the following error it may be due to Docker not running.

***ERROR: Couldn't connect to Docker daemon - you might need to run 'docker-machine start default'***

To resolve this error, ensure Docker is actively running on your system. If you continue to have problems, follow the link to Docker troubleshooting in the **Setup** section.

### 2.Application Not Running

When attempting to view the application via the web browser at <http://localhost:5931>, you may see a blank screen or a screen that continuously says "loading..". This indicates that the application has not successfully finished loading. Ensure steps 1-3 have been completed in the **Application Setup** instruction and that several minutes have passed in order to give the server enough time to properly finish setting up. If the issue persists for more than a few minutes, quit Docker and perform steps 1-3 in the **Application Setup**. If this does not resolve the issue, try clearing the cache in your current web browser or try a different web browser.

### 3.Docker Unexpectedly Quit

After successfully running the application and interacting with the frontend via the web browser,

may experience the application is no longer is able to connect to the server. This error may be due to Docker quitting unexpectedly. If the console contains "LOG: Shutting down", "Unexpected API error" or "Bad response from Docker engine" this indicates that Docker has quit while the application was still running. To resolve this issue, attempt to restart Docker on your system and try again. If you are unable to restart Docker, follow the link to Docker troubleshooting in the **Setup** section.

## User Accounts

CAP Clinical Stability Predictor does not require any user accounts to use and is accessible to anyone with the Docker images. Future enhancement to the application could be added to implement a user sign-in and display patient's specific to the clinician who is currently signed in.

## Using the application

At this point, you should have successfully started the Docker images and see the main screen of the application. If you have not performed these steps, please refer to the **Application Setup** section under **Setup** for further information on how to setup the application.

## Patient List

The initial view presented in the application is the patient list. The patient list displays a comprehensive list of patients who are currently diagnosed with CAP. The patient list is shown in the image below where we see three patients who have been diagnosed with CAP. From the patient list clinicians are able to monitor the status of their patients. The patient list is updated once every 24 hours where the patient's vital signs, length of stay and clinical stability will be updated.

CAP Clinical Stability Predictor					
Amy V Shaw DOB: 2007-03-20 MRN: 12081332 <b>Clinically Not Stable</b>					
110/63 mmHg BP	81 HR	34 RR	38 C Temp	100% SpO2	
Length of Stay: 4 days					
Robert Smith DOB: 1998-05-20 MRN: 21903927 <b>Clinically Stable</b>					
113/75 mmHg BP	72 HR	21 RR	38 C Temp	100% SpO2	
Length of Stay: 5 days					
Jon Doe DOB: 2001-01-13 MRN: 50239183 <b>Clinically Not Stable</b>					
122/69 mmHg BP	102 HR	40 RR	39 C Temp	100% SpO2	
Length of Stay: 1 day					

## Individual Patient View

Within the patient list view, are individual patients. For each patient, the application presents to the

clinician information about the patient including their name, vital signs and stability status as shown in the image below. From this information, the clinician is able to quickly determine the required actions to take for an individual patient.

<b>Amy V Shaw</b> <b>DOB: 2007-03-20</b> <b>MRN: 12081332</b> <b>Clinically Not Stable</b>				
<b>110/63 mmHg</b> <b>BP</b>	<b>81</b> <b>HR</b>	<b>34</b> <b>RR</b>	<b>38 c</b> <b>Temp</b>	<b>100%</b> <b>SpO2</b>
<b>Length of Stay: 4 days</b>				

#### Individual Patient View Key

##### Name

The patient's name.

##### DOB

The patient's date of birth.

##### MRN

The patient's medical record number.

##### Clinical stability

The status of the patient. This field can be either 'Clinically Stable' or 'Clinically Not Stable'.

Clinically Stable indicates that based upon the patient's vital signs, the patient is exhibiting normal vital signs and clinicians can choose to switch therapy early in favor of intravenous antibiotics to oral antibiotics and possible early hospital discharge.

Clinically Not Stable indicates that based upon the patient's vital signs, the patient is not well enough to be discharged and further observation is required.

##### BP

The patient's systolic blood pressure measured in Millimetres of mercury (mm Hg).

##### HR

The patient's heart rate measured in beats per minute (bpm).

##### RR

The patient's respiratory rate measured in breath rate per minute (brpm).



## Temp

The patient's temperature measured in degrees Celsius.

## SpO2

The patient's room air pulse oximetry (RA pulse ox) measured in percent.

## Length of Stay

The amount of time the patient has been admitted measured in days.

## Deleting Patients

Once the clinician has taken action for an individual patient or once a patient has been discharged, the clinician can remove the patient from the patient list. To do so, the clinician can select the "X" button in the top right corner of the individual patient they wish to remove. This will remove the patient from the list of patients and the clinician can focus on their remaining patients. Note, deleting a patient does not persist on the server. If the page is refreshed, all matching patients will be in view again. This is to prevent needing to implement a recovery mechanism for "deleted" patients

## Developer Instructions

To begin development on CAP Clinical Stability Predictor there are a few necessary steps.

The public ports are in the docker-compose.yml including 5839 for the backend and 5931 for the frontend. If the backend port needs to change, an additional file needs to change in the frontend prior to building the containers (since it depends on the URL from the browser). Edit the port listed in frontend/src/

## Test Patients

- Amy Shaw - <http://localhost:5839/patients/np-2>
  - Condition: SNOMED-CT 233604007
  - Status: active
  - Length of stay: 4 days
  - Clinically stable: No
    - respiratory rate > 24
- Jon Doe - <http://localhost:5839/patients/np-2>
  - Condition: SNOMED-CT 385093006
  - Status: active
  - Length of stay: 1 day
  - Clinically stable: No
    - temperature > 37.7
    - heart rate > 100
    - respiratory rate > 24

- Robert Smith - <http://localhost:5839/patients/np-3>
  - Condition: ICD-10 J18.9
  - Status: active
  - Length of stay: 5 days
  - Clinically stable: Yes
- Dana Lee - <http://localhost:5839/patients/np-4>
  - Condition: ICD-10 J18.0
  - Status: resolved
  - Clinically stable: Yes
- Harper Miller - <http://localhost:5839/patients/np-5>
  - Condition: No match (control)

## Usage

- Go to <http://localhost:5931> in your browser to view the frontend.
- View an API endpoint by going to <http://localhost:5839/patients>.
- View the FHIR server frontend by going to <http://localhost:8080>

## Architecture

### Backend

The *backend* talks to the FHIR server and encapsulates the logic for identifying relevant patients and resources. For each patient, the data are passed into the *model* which applies the expert rules to determine if a CAP patient is "clinically stable". The result of this is augmented onto the data payload and then sent to *frontend* which renders the information.

### Install

Tested on Python 3.4+

```
pip install -r requirements.txt
```

### Run

```
python main.py
```

Environment variables:

- HOST - Host to listen on. Use 0.0.0.0 for all interfaces. (default 127.0.0.1)
- PORT - Port to bind to. (default 5000)
- DEBUG - Set to 1 for debug mode. (default 0)
- FHIR\_API\_URL - URL to a FHIR server.
- FHIR\_REF\_DATE - Fixed reference time for queries. If not set, every request will query relative to the current server time. In other words, this should *only* be set if the FHIR server does not contain

live data.

Variables can be set inline to executing the program:

```
DEBUG=1 PORT=6000 python main.py
```

## Endpoints

- /patients - Get a list of all patients with Pneumonia.
- /patients/<id> - Get details about a patient in the system and their CAP-related info.

## Model

The model has been implemented on the backend. Accessing a specific patient's detail page will compute the stability of the patient based on the last 24 hours of vital sign data.

## Frontend

The frontend uses React to create reusable UI components. The frontend will retrieve patient data such as vital signs and CAP recommendation from the backend by calling the /patients endpoint from the backend to update the state of the patients object.

We are using Redux Saga to manage state of our application.

With this data it will pass into React UI components Card this component dynamically generates a patient card for every patient that is in the patients object.

Deleting patients updates the state of the current view by calling the PATIENT\_DELETE action. This removes the patient object from the patients list by id from the state.

## FHIR Data

To import the FHIR data, clone the tag-uploader repository in the `data/` directory and install the Node dependencies.

```
git clone https://github.com/smart-on-fhir/tag-uploader\ cd tag-uploader\ npm install
```

Ensure the FHIR server is running (using `docker-compose up fhir`), then run the command in the tag-uploader directory to import the test patient FHIR bundles in the data/bundles directory from the root of the repository.

```
node . -v -d ../bundles -S http://localhost:8080/baseDstu3
```