



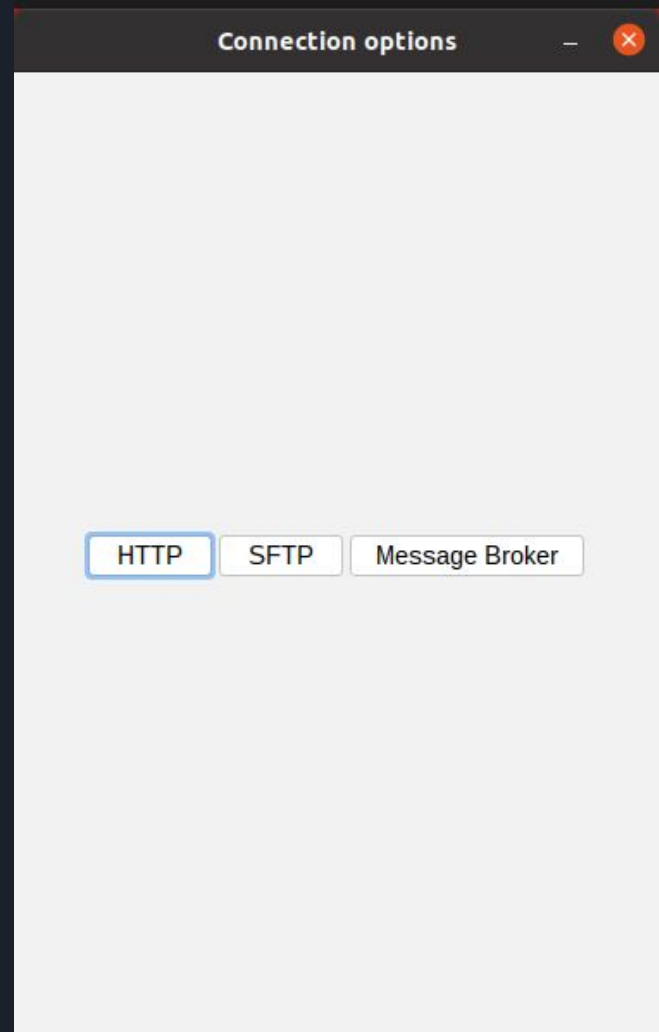
# Healthcare Data Simulators

Summary for the client - Beta Release



# Short walkthrough

When you run the app you can choose the protocol that you want to use



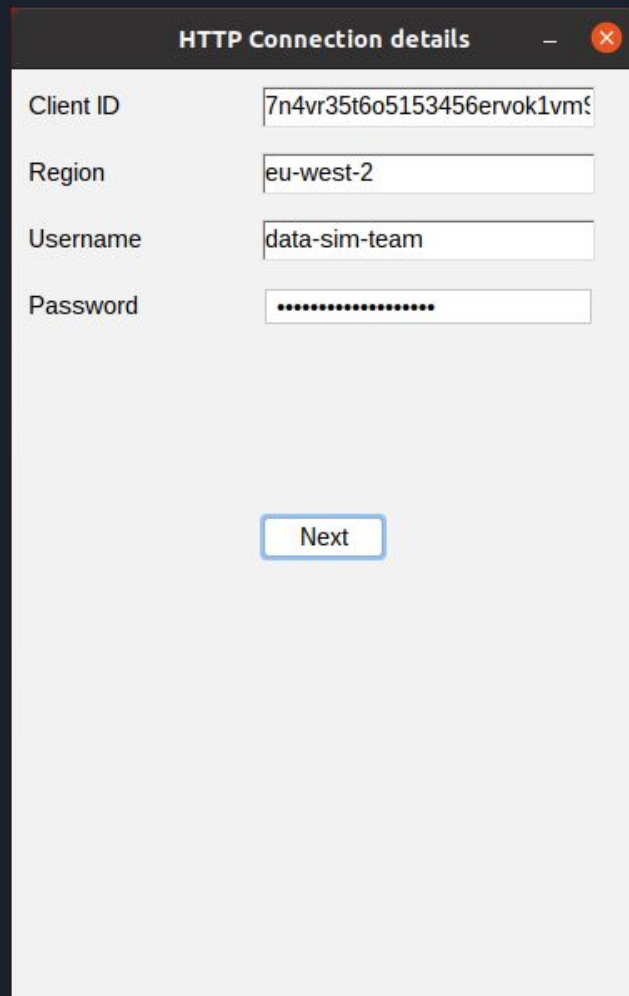


# Short walkthrough

If HTTP protocol is chosen you will have to enter the connection details.

By default here are the details are the connection details provided from the Data Lake Team.

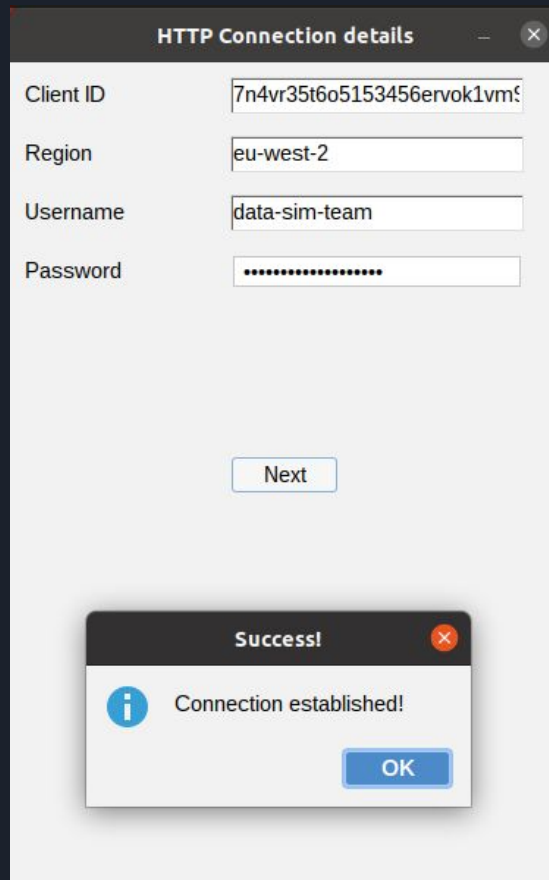
By clicking next it will request a token from AWS Cognito in order to establish the connection.



HTTP Connection details	
Client ID	7n4vr35t6o5153456ervok1vm5
Region	eu-west-2
Username	data-sim-team
Password	.....
<a href="#">Next</a>	

## Short walkthrough

Token has been received  
so we can start sending  
POST requests to the  
server



The image shows a software interface with two windows. The top window, titled 'HTTP Connection details', contains four input fields: 'Client ID' with the value '7n4vr35t6o5153456ervok1vm...', 'Region' with 'eu-west-2', 'Username' with 'data-sim-team', and 'Password' with masked characters. A 'Next' button is located below these fields. The bottom window, titled 'Success!', features an information icon and the text 'Connection established!', with an 'OK' button at the bottom right.

HTTP Connection details	
Client ID	7n4vr35t6o5153456ervok1vm...
Region	eu-west-2
Username	data-sim-team
Password	.....

Next

**Success!**

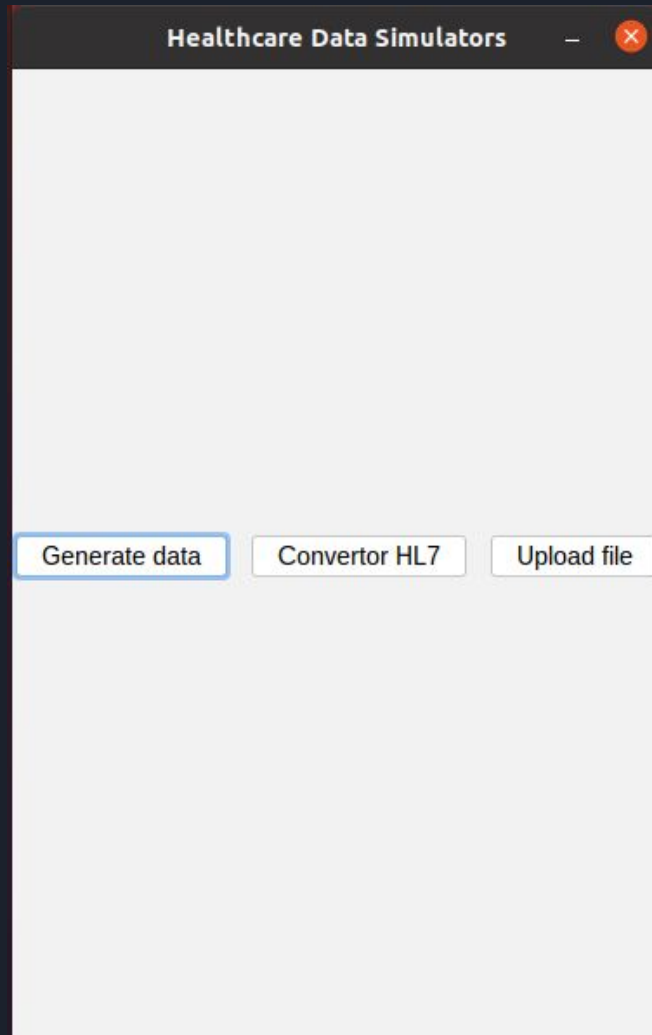
*i* Connection established!

OK

## Short walkthrough

We can choose now between the following options:

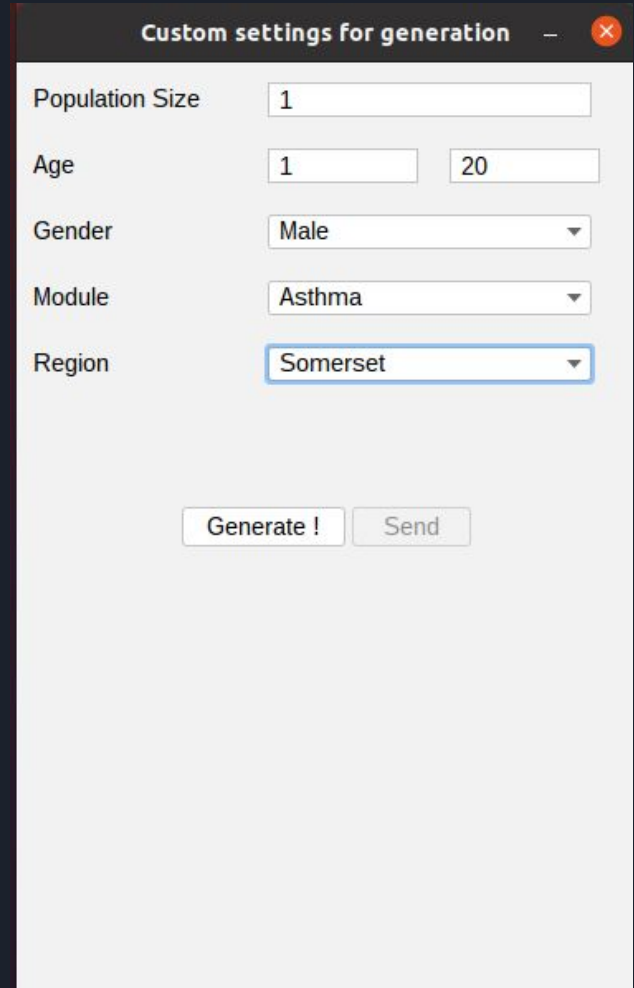
- Generate Data = use Synthea Patient Generator
- Convertor HL7 = convert local HL7 V2.x files to FHIR
- Upload File = send any kind of data (e.g. pdf, png, any local stored file)



## Short walkthrough

In this example we opted for “Generate Data”.

You can input your own configuration.



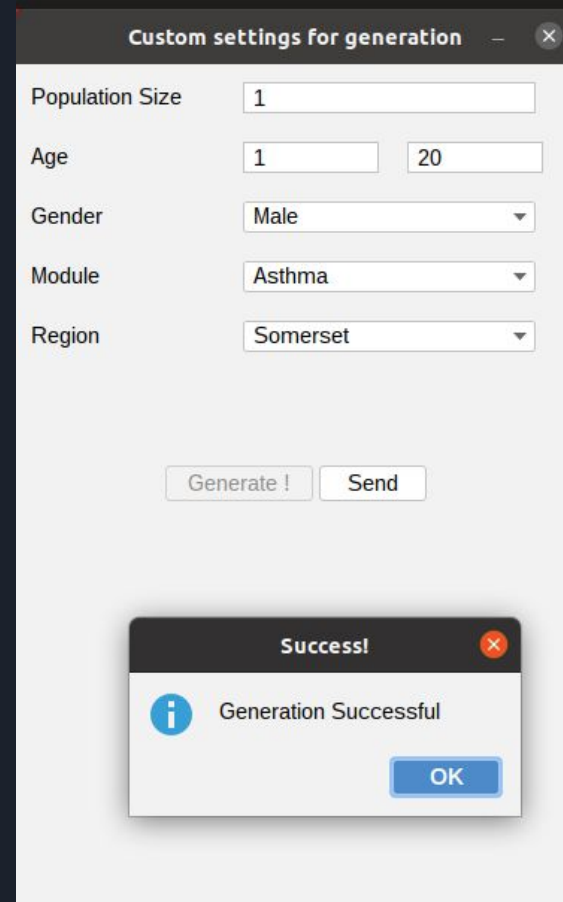
The screenshot shows a dialog box titled "Custom settings for generation" with a close button (X) in the top right corner. The dialog contains the following settings:

- Population Size: 1
- Age: 1 (min) and 20 (max)
- Gender: Male (dropdown menu)
- Module: Asthma (dropdown menu)
- Region: Somerset (dropdown menu)

At the bottom of the dialog, there are two buttons: "Generate !" and "Send".

## Short walkthrough

“Under the hood”  
the data has been  
generated and is  
ready to be sent  
using the chosen  
protocol.



The image shows a software interface for generating data. The main window is titled "Custom settings for generation" and contains several input fields: "Population Size" (set to 1), "Age" (set to 1 and 20), "Gender" (set to Male), "Module" (set to Asthma), and "Region" (set to Somerset). Below these fields are two buttons: "Generate !" and "Send". A smaller "Success!" dialog box is overlaid on the bottom right, displaying an information icon and the text "Generation Successful", with an "OK" button.

Setting	Value
Population Size	1
Age	1, 20
Gender	Male
Module	Asthma
Region	Somerset

Buttons: Generate !, Send

Success! Generation Successful

OK



# Short walkthrough

Here is a picture of my terminal where the application printed out the response messages which confirms that the data has been successfully delivered.

```
ecute-api.eu-west-2.amazonaws.com/dev/Claim}
Response{protocol=http/1.1, code=201, message=Created, url=https://e81uscwufb.ex
ecute-api.eu-west-2.amazonaws.com/dev/ExplanationOfBenefit}
Response{protocol=http/1.1, code=201, message=Created, url=https://e81uscwufb.ex
ecute-api.eu-west-2.amazonaws.com/dev/ExplanationOfBenefit}
Response{protocol=http/1.1, code=201, message=Created, url=https://e81uscwufb.ex
ecute-api.eu-west-2.amazonaws.com/dev/Claim}
Response{protocol=http/1.1, code=201, message=Created, url=https://e81uscwufb.ex
ecute-api.eu-west-2.amazonaws.com/dev/Encounter}
Response{protocol=http/1.1, code=201, message=Created, url=https://e81uscwufb.ex
ecute-api.eu-west-2.amazonaws.com/dev/Encounter}
Response{protocol=http/1.1, code=201, message=Created, url=https://e81uscwufb.ex
ecute-api.eu-west-2.amazonaws.com/dev/Encounter}
Response{protocol=http/1.1, code=201, message=Created, url=https://e81uscwufb.ex
ecute-api.eu-west-2.amazonaws.com/dev/Immunization}
Response{protocol=http/1.1, code=201, message=Created, url=https://e81uscwufb.ex
ecute-api.eu-west-2.amazonaws.com/dev/Immunization}
Response{protocol=http/1.1, code=201, message=Created, url=https://e81uscwufb.ex
ecute-api.eu-west-2.amazonaws.com/dev/Claim}
Response{protocol=http/1.1, code=201, message=Created, url=https://e81uscwufb.ex
ecute-api.eu-west-2.amazonaws.com/dev/Practitioner}
Response{protocol=http/1.1, code=201, message=Created, url=https://e81uscwufb.ex
ecute-api.eu-west-2.amazonaws.com/dev/ExplanationOfBenefit}
```

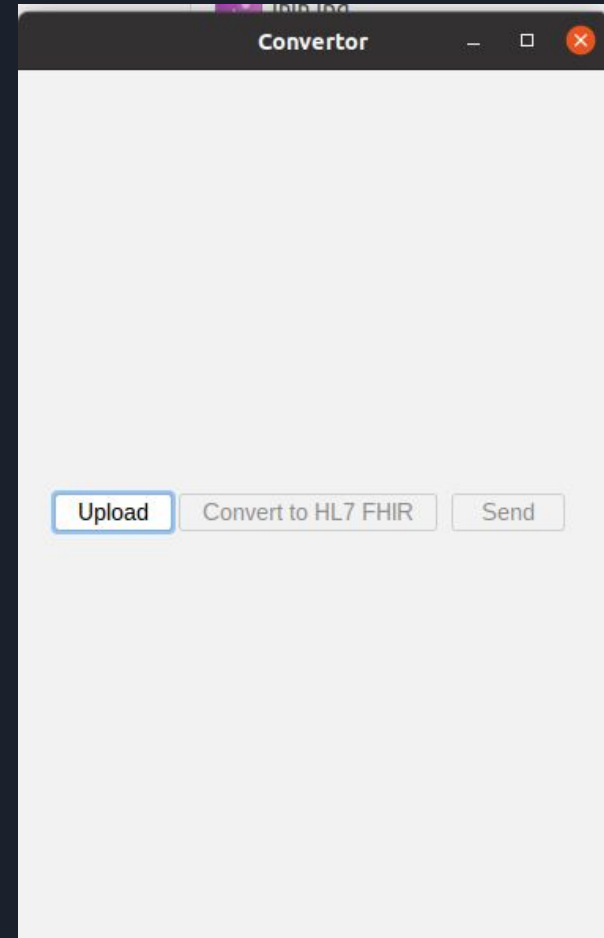




## Short walkthrough

Here is a screenshot from the GUI of the Convertor.

You need to upload the data, then you can convert it to FHIR and send it.





# Installation requirements

You need have to install on your local machine the following:

1. Java 11
2. Python 3
3. boto3 module for Python `pip install boto3`

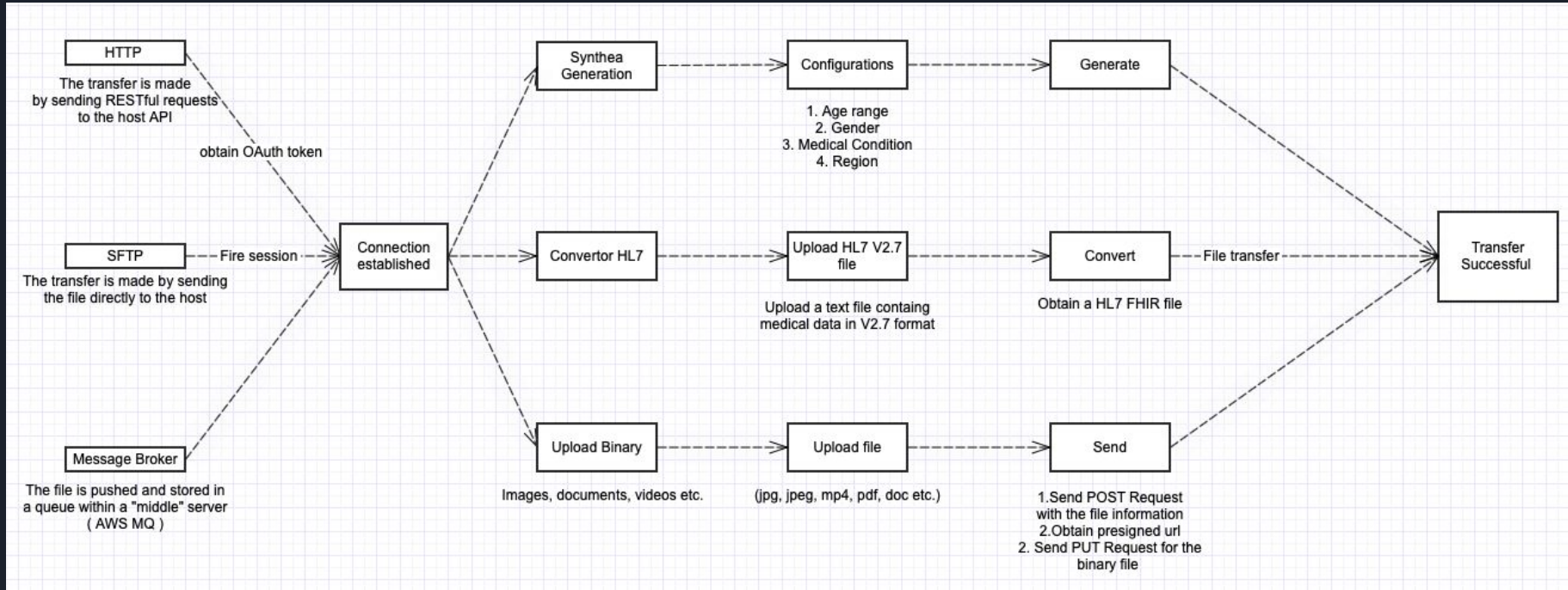


# Installation guide

The installation guide can be found on our repository page:

[Click here to access our github repository](#)

# Architecture - WorkFlow + Process Details





# About the \*.jar files dependencies

We made use of [Synthea™ Patient Generator](#) and [synthea-international](#) to make a synthetic population generator which includes UK population. We have deployed an executable jar files which download link can be found on our repository page.

We modified an open-source project which converts HL7 V2.7 into FHIR : [github repository](#). The project was modified and redeployed in order to make it as portable and as efficient as possible to be easily integrated in our project. The deployed jar can be found on our repository page.