
Article

[Phillip Booth](#) · Jan 30, 2020 3m read

[Open Exchange](#)

Using Synthea and Docker for Consistent, Realistic Synthetic Patient Generation.

Over the last couple of weeks the Solution Architecture team has been working to finish off our 2019 workload: this included open-sourcing the Readmission Demo that was brought to HIMSS last year, so we could make it available to anyone looking for an interactive-way of exploring the tooling provided by IRIS.

While in the process of open sourcing the demo we were immediately hit hard with a showstopper. The underlying patient data that was used to build the demo could not be used as part of an open-source project because it was not owned by InterSystems; it was owned by our Partner Baystate Health.

Our group was in a bit of a jam and needed to come up with a way of replacing the original data with usable synthetic data while still keeping the demos “ story ” , or its underlying functionality, consistent. Since the demo showcases how IRIS supports a data scientist ’ s machine learning workflow, there was an added level of complexity because any data we ended up using needed to be realistic enough that it could support our investigative modeling. After a brief bit of digging, Synthea came to our rescue.

Synthea is an open-source, synthetic patient generator that models the medical histories of synthetic patients. Synthea provides high-quality, realistic but not real, patient data in a variety of formats (FHIR include) with varying levels of complexity, covering every aspect of healthcare. The resulting data is free from cost, privacy, and security restrictions, enabling research with Health IT data that is otherwise legally or practically unavailable.

After some initial exploration, Synthea was chosen as the tool to fix our data problem. Synthea is an incredible tool, however one issue that we found was that in order to run the software and get patients we needed to install multiple dependencies on our machines.

- Java JDK 1.8
- Gradle Build Tool

When working by yourself this generally isn ’ t an issue, but since our team consists of multiple

individuals, it ' s important that everyone can get up to speed with a new software quickly and installing dependencies can be a nightmare. We believe in making as few people as possible suffer through installation processes when integrating a new software into our workflow.

Since we needed anyone on our team making updates to the Readmission Demo to be able to easily generate patients, and we didn ' t want everyone to have to install Gradle on their machines; we leaned on Docker and injected the Synthea software inside of a Docker image, allowing the image to take care of the underlying environmental dependencies.

This ended up working great for our team; we figured that being able to generate synthetic patient data on the fly is probably a very common use case that our fellow Sales Engineers face, so our team wanted to share it with the Developer Community.

Anyone can use the line of code below to quickly generate 5 synthetic-patient medical histories in FHIR format, and have the resulting patients left in an output folder in the current working directory.

```
docker run --rm -v $PWD/output:/output --name synthea-docker  
intersystemsdc/irisdemo-base-synthea:version-1.3.4 -p 5
```

The code for this Docker Image has its own repository on Github and can be found here for anyone looking to take a look, make custom changes, or contribute:

<https://github.com/intersystems-community/irisdemo-base-synthea>

We are currently in the process of making updates so that the project supports custom modules, so anyone looking to add an illness to their generated patients that Synthea doesn ' t currently provide out of the box can do so, and it will be automatically built into their image.

Where is it Currently Being Used?

The current build process of the Readmission Demo uses the irisdemo-base-synthea image to generate 5000 synthetic patients on the fly and load them into our IRIS normalized, relational DataLake. For anyone interested in checking out how to parse this generated, synthetic patient data (In FHIR format) please check out the recently open sourced Readmission Demo. The class you are looking for is: IRISDemo.DataLake.Utils. starting at line 613.

The Readmission Demo can be found here: <https://github.com/intersystems-community/irisdemo-demo-readmission>

[#Analytics](#) [#Best Practices](#) [#Big Data](#) [#Containerization](#) [#Docker](#) [#Tools](#) [#InterSystems IRIS for Health](#)
[Check the related application on InterSystems Open Exchange](#)

Source

URL:<https://community.intersystems.com/post/using-synthea-and-docker-consistent-realistic-synthetic-patient-generation>