

# FHIR Workshop for Ready 2025

For developers interested in reproducing the FHIR workshop exercises for FHIRPath and FHIRAnalytics on their laptop, follow the steps below.

## Initial Setup

The most important step is setting up a FHIR development environment which includes the IRIS for Health Community Edition 2025.1 with a pre-configured FHIR server and sample patients, Docker, VS Code, Python, and Jupyter Notebooks.

I strongly recommend using the FHIR environment setup guide which you can find on the developer community at <https://community.intersystems.com/post/fhir-environment-setup-guide>.

That article include two videos explaining the development environment setup in detail. One of these videos is for setting up the FHIR development environment on Windows: <https://youtu.be/lyvuHbxCwCY> and the other is setting up the FHIR development on macOS <https://youtu.be/Ss7vU0l3JNU>.

Once these steps are complete, in the same folder or directory that you created your development environment, open a terminal window, and download or clone the FHIR workshop exercise material at: <https://github.com/intersystems/Ready2025.git>.

Make sure your docker container is running which contains IRIS for Health and the FHIR Server with sample patients. You can validate your FHIR server is running by opening the file search.http and executing the metadata request: (GET) <http://127.0.0.1:8080/csp/healthshare/demo/fhir/r4/metadata>

If the capability statement comes back, you are ready go.

## FHIRPath

The entire FHIRPath exercise is described in the Jupyter notebook, FHIRPath/Fhirpath1.ipynb. If you have set up your environment correctly, you should have no trouble running all the cells. There are some cells in the exercise that make a

call to the local FHIR server, which was in the development environment setup. Make sure this server is running, by calling `docker ps` and creating a the metadata request to the FHIR server at <http://127.0.0.1:8080/csp/healthshare/demo/fhir/r4/metadata>.

We will not be using FHIRpath2.ipynb, but you are welcome to examine it.

## FHIR SQL Builder

Again, make sure your FHIR server is running, by calling `docker ps` and making a FHIR metadata request: <http://127.0.0.1:8080/csp/healthshare/demo/fhir/r4/metadata>. If the server is not running, you will not be able to use the FHIR SQL Builder to perform any of the steps in the exercise.

You can access the FHIR SQL Builder at <http://127.0.0.1:8080/csp/fhirsql/index.html>. You must first login. The demo environment has been configured with Username = \_SYSTEM, Password = ISCDemo (all caps).

There are three steps to creating a projection: analysis, transformation, and projection. I have made a video that walk you through all these steps at: <https://bcove.video/4iRIDN8>

To create your first analysis, you must configure the FHIR SQL Builder for the first time. Click the new button above analysis.

Name\*

The name for this FHIR Repository Configuration – choose a name such as fhirserver

Host\*

The DNS name or IP address of the host of the repository to analyze. For the local host the IP address is 127.0.0.1

Port\*

The port that should be used to access the FHIR repository on our local. In our case 52773

URL Prefix

Optional URL prefix. A valid prefix starts with a forward slash (/) – just leave blank.

SSL Configuration

The SSL connection to use

Leave blank for this exercise.

### Credentials\*

The credentials to use to log into Host

You should create a new username and password and give it a label. For this exercise, I am using admin as the label, with the username= \_SYSTEM and password = ISCDemo

### FHIR Repository URL\*

The URL of the FHIR repository. Please fill in Host, Port and Credentials first. The dropdown should show the FHIR endpoint path, which for us is at:

/csp/healthshare/demo/fhir/r4

You can now launch a new analysis task. Since this is a small repository of roughly 70K FHIR resources, select 100 percent of the records to be analyzed and click "Launch Analysis Task". This will take a few minutes but provide a detailed picture of all the Resources (including their elements) in your FHIR server.

The next step is to create a transform. Click on the button "new" above the transform pane and select the analysis you just created. Give the transform a simple name and description for future reference.

For the exercise we need to create two transforms with the following columns:

For Patient Table (pt\_transform) , here are the columns that are needed:

"ID"  
"BirthDate"  
"FirstName"  
"Gender"  
"LastName",  
"LastUpdated",  
"MaritalStatus",  
"PatientBirthPlaceCity"

For the Allergy Intolerance Table (allergy\_transform), here are the columns that are needed:

"ID"  
"AllergyCode" (this code is the is under clinical status and indicates whether the allergy is active or inactive)  
"AllergyDisplay"  
"LastUpdated"  
"PatientReference"

It is easier to show rather than describe how to make these transformations, so please watch the video at <https://bcove.video/4iRIDN8>

Once the transformations are built, we can create a SQL projection. Click the “new” button over the projection panel and fill in the text boxes. The drop down selects the FHIR repository we are projecting over. Since this was already set up in the analysis phase, just select the name you gave, in my case “fhirserver”. Now select the transform you created. We can have more than one table in a transformation, but in our case, since we created only one table in each transform, pt\_transform and allergy\_transform, we will create a separate project for each one. Let’s start with the pt\_transform and place it in SQL Package sql1 (the name is up to you) and the second transform, allergy\_transform and place it in SQL Package sql2. Click “launch projection” to start the projection which starts very rapidly.

Back on the main page you will see a symbol that looks like a chain link in the projection panel. If you click on this link, you will be taken to the management portal. Unfortunately, even though our FHIR server is running at port 52773 in the container, we are running the docker container at port 8080. So change the url, [http://127.0.0.1:52773/csp/sys/exp/%25CSP.UI.Portal.SQL.Home.zen?\\$NAMESPACE=DEMO](http://127.0.0.1:52773/csp/sys/exp/%25CSP.UI.Portal.SQL.Home.zen?$NAMESPACE=DEMO) to port 8080 like so:

[http://127.0.0.1:8080/csp/sys/exp/%25CSP.UI.Portal.SQL.Home.zen?\\$NAMESPACE=DEMO](http://127.0.0.1:8080/csp/sys/exp/%25CSP.UI.Portal.SQL.Home.zen?$NAMESPACE=DEMO)

Now you can access your projected tables by filtering the SQL tables by the package name(s) you supplied. Again, look at the video to see how this is done <https://bcove.video/4iRIDN8> .

The analysis of this data is done using Python and Jupyter notebooks. Open the Jupyter notebook, FHIRAnalytics/AllergyIntolerance.ipynb and follow the directions in the notebook. The notebook also covers how to use the InterSystems Python db-api driver to get the projected data from our FHIR repository.