FHIR® DevDays Virtual
November 17-20, 2020

Let’s build a FHIR app - .Net

Day 1 - Build a FHIR Data Mapper

Rationale: When adopting FHIR, a common scenario is needing to convert your existing data into the FHIR model. This can be a challenging first step, but if you approach it systematically it can be easy.

Exercise: For this exercise, we will be building a mapper that converts existing data a CSV file into FHIR Patient and Observation resources. We will use the Firely .NET SDK for the FHIR models.

We'll be using a sample data file from the CDC NHANES (National Health and Nutrition Examination Study) publicly available sample data set. The format of the data set is described at this link but James Agnew has reworked the format a bit to add fake patient identities and timestamps to the data.

The input CSV file can be found here: https://bit.ly/MirjamsTraining - /sample-data.csv

Approach

The input data looks like the following:

```
SEQN   ,TIMESTAMP               ,PATIENT_ID,PATIENT_FAMILYNAME,PATIENT_GIVENNAME,PATIENT_GENDER,WBC,RBC,HB
93704.0,2020-11-13T07:47:35.964Z,PT00002   ,Simpson           ,Marge            ,F             ,7.4,0.1,13.1
```

Note the columns:

- SEQN: This is a unique identifier for the test
- TIMESTAMP: This is the timestamp for the test
- Patient detail columns (note that the patients repeat so you will want to):
  - PATIENT_ID: This is the ID of the patient
  - PATIENT_FAMILYNAME: This is the family (last) name of the patient
  - PATIENT_GIVENNAME: This is the given (first) name of the patient
  - PATIENT_GENDER: This is the gender of the patient
- Test result columns (each of these will be a separate Observation resource):
  - WBC: "White Blood Cells": This a count of the number
  - RBC: "Red Blood Cells"
  - HB: "Hemoglobin"

Information for mapping the Observations:

White blood cell count - This corresponds to LOINC code:

Code:	6690-2
Display:	Leukocytes [#/volume] in Blood by Automated count
Unit System:	http://unitsofmeasure.org
Unit Code:	10*3/uL

Red blood cell count - This corresponds to LOINC code:

Code:	789-8
Display:	Erythrocytes [#/volume] in Blood by Automated count
Unit System:	http://unitsofmeasure.org
Unit Code	10*6/uL

Hemoglobin:

Code:	718-7
Display:	Hemoglobin [Mass/volume] in Blood
Unit System:	http://unitsofmeasure.org
Unit Code:	g/dL

Exercise steps

- Create a new solution. In the example project, we have used a simple Console app, but you might want to display your data in a nicer way.

- Add the Hl7.Fhir.R4 library through NuGet

- Optionally add a library with methods that can help you read in the CSV, like CSVHelper

- Create a model class that contains the structure of the CSV

- Add code to read in the CSV data and put it into that model
    - You can add the file to your project, or make sure it is in a location accessible to your app
    - In the example, we have read in the complete file resulting in a collection of the data, but it could also be done one line at a time

- Now create a Mapper class, which will contain the code to transform the custom data structure to FHIR resources

- For each line:
    - Map the patient data to a FHIR Patient object
        - You need to convert the gender in the CSV (‘M’ and ‘F’) to the corresponding allowed value in FHIR (‘male’ and ‘female’) by using the AdministrativeGender enum that is in the FHIR library
        - You can add a new HumanName object with the given and family names to the name field
        - You can set the technical id for the Patient to the patient id in the CSV – although sending the Patient to a server with a ‘create’ interaction will overwrite this, it is useful for now, so you can link Observations to the Patient
    - Map the white blood cell count, red blood cell count and hemoglobin to three separate Observation objects using the mapping information provided
        - Set the code field by creating a new CodeableConcept with the LOINC code mentioned, and use ‘http://loinc.org’ for the system value
        - Fill the subject field with a new Reference to ‘Patient/<patient_id>’. This links the Observation to the Patient it is for.
        - Look at http://hl7.org/fhir/observation.html to see if you have data for any of the other fields, and if so, map them.
        - Optionally, you can set the category to ‘laboratory’ by creating a new CodeableConcept with that string as code, and ‘http://terminology.hl7.org/CodeSystem/observation-category’ as system

- After reading all lines, you should end up with a set of FHIR Patient objects, and a set of FHIR Observation objects
    - Bonus if you have added each Patient to your set only once!

- With your set of Observations and Patients, display the values nicely
    - For example, per Patient show their blood values in a table or create a chart
        - Bonus points if you create a nicer display than just the plain text output in a Console app!
    - Make sure you use the FHIR resources, not the CSV data itself. In a next exercise we will retrieve resources from a FHIR server and display that data the same way

Have fun, and remember to ask for help if you get stuck!