Module 4 Exercise 1 - Setting up an HL7 feed in IRIS for Health

Task:

Instructions:

The first step is connecting to the AWS EC2 instance. Navigate to the following link: http://ec2-18-219-162-44.us-east-2.compute.amazonaws.com/irishealth/csp/sys/UtilHome.csp

You will be assigned a username and password to use in the format of: train# with the same password, train# i.e. train1/train1

On your assigned FHIRTRAIN namespace of your HealthShare instance, navigate to the EdgeGatewayProduction by clicking on Interoperability -> Configure -> Production.

Creating an HL7 to FHIR Integration

Learning Objectives:

- Build an end-to-end HL7 to FHIR pipeline
- Learn about standard IRIS FHIR processes, operations, and transformations
- Configure the IRIS FHIR Server
- Read a FHIR message trace
- Validate a FHIR Bundle using FHIR Validator

Ensure your container is running or refer back to Module 5: Exercise 1.

Load Necessary Code: From the IRIS System Management Portal, navigate to System Explorer -> Classes. Select Import and choose **My Local Machine** and click on Browse to locate all files found under the /irisdata/module4-exercise1-code/ folder. Make sure File of type is set to All Files (*) so you can see all the files available to select. Once all files are selected, choose Next and then Import. The code should all compile without errors.

From the IRIS System Management Portal, open the FHIRDEMO Production by navigating to Interoperability -> Configure -> Production from the **FHIRDEMO** namespace. Execute the following steps:

1. **Add a Business Service:** Click on the plus + next to the **Services** header.

Configure these **Business Service** settings in the wizard:

Configuration Name	Value
Service Class	EnsLib.HL7.Service.FileService
Service Name	HL7InboundFileService1
Display Category	module4-Exercise1

Configuration Name	Value
Enable Now	Not Selected

The service should be added to the production now. The service should remain disabled for now.

2. **Add a Business Process:** Click on the + symbol next to the **Processes** header.

Configure these **Business Process** settings in the wizard:

Configuration Name	Value
Business Process Class	HS.Local.FHIR.HL7toSDAProcess
Business Process Name	FHIR.HL7toSDA1
Display Category	module4-Exercise1
Enable Now	Selected

3. **Add the Standard FHIR Business Process:** Click on the + symbol next to the **Processes** header. (We will have two business processes).

Configure these **Business Process** settings in the wizard:

Configuration Name	Value
Business Process Class	HS.FHIR.DTL.Util.HC.SDA3.FHIR.Process

Configuration Name	Value
Business Process Name	HS.FHIR.DTL.Util.HC.SDA3.FHIR.Process1
Display Category	module4-Exercise1
Enable Now	Selected

4. **Add a Standard FHIR Server Operation:** Click on the + symbol next to the **Operations** header.

Configure these **Business Operation** settings in the wizard:

Configuration Name	Value
Operation Class	HS.FHIRServer.Interop.Operation
Operation Name	HS.FHIRServer.Interop.Operation
Display Category	module4-Exercise1
Enable Now	Selected

5. **Build End-to-End**: Now you have four business components. In order to hook them together to run and end-to-end, we'll cover the **Properties** for each of the components.

5-1. Start on the left by clicking on the icon/name for the **HL7InboundFileService1** service:

Then click on the **Settings** tab on the right panel to configure the service properties:

Property Name	Value
File Path	/tmp/fhir-trainingXX/hl7
TargetConfigNames	FHIR.HL7toSDA1

> Replace fhir-trainingXX with your assigned number.

Make sure to click **Apply** to save your Settings.

5-2. Click on the icon/name for the **FHIR.HL7toSDA1** Process.

This is a custom process that is identifying what HL7 field will be set as the `PatientResourceId`, which is required in the Patient Resource. Configure the following Settings.

Property Name	Value
PatientIdLocation	PID:3.1
TargetConfigNames	HS.FHIR.DTL.Util.HC.SDA3.FHIR.Process1

Make sure to click **Apply** to save your Settings.

5-3. Click on the icon/name for the **HS.FHIR.DTL.Util.HC.SDA3.FHIR.Process1** Process.

Configure the following Settings.

Property Name	Value
TargetConfigNames	HS.FHIRServer.Interop.Operation
TransmissionMode	transaction
FHIRMetadataSet	HL7v40 / FHIR R4 Core Specification
FHIREndpoint*	/csp/healthshare/fhirdemo/fhir/r4/
LogTraceEvents	*checked*
TraceOperations	*FULL*

- The **FHIREndpoint** path doesn't yet exist, but you will configure this in a few steps.

Make sure to click **Apply** to save your Settings.

6. **Add the Trace Logging Operation**

Configuration Name	Value
Operation Class	HS.Util.Trace.Operations
Operation Name	HS.Util.Trace.Operations
Display Category	module4-Exercise1
Enable Now	Selected

Make sure to click **Apply** to save your Settings.

7. **Configure the FHIR Server**

With the **FHIRDEMO** namespace selected, click on **Home** and then select **HEALTH** either on the panel on the left or at the top of the System Management portal.

Select the **FHIR Configuration** section:

7-1. Log in with the same username and password for IRIS. User:`_system` Password: `SYS`

7-2. Select **Server Configuration** and then click the **Add Endpoint** button.

Enter these configurations: Configure the following Settings.

Property Name	Value
CORE FHIR package	hl7.fhir.r4.core@4.0.1
URL	/csp/healthshare/fhirtrain#/fhir/r4
Additional packages	hl7.fhir.us.core@3.1.0
Interactions Strategy Class	HS.FHIRServer.Storage.Json.Interactions.Strategy
Storage	*Keep all default values*

> Replace fhirtrain# with your assigned number.

Click **Add**. It will take a few minutes to build the endpoint. You can leave this screen and return to VSCode while the endpoint builds.

8. **Run the Input File**

Starting the HL7 Inbound service will run an HL7 file through.

9. **Check the Production:** Return to your System Management Portal. If you are looking at the "FHIR Server" screen, you can click on the profile for the `_System` user in the right corner. Once you click on the icon, select **Management Portal** to return **Home**.

Go to **Home -> Interoperability -> Select FHIRDEMO -> Configure -> Production**.

Click on the **Messages** tab. You should see the available message traces. Click on the link under **Header** to trace the activity.

If you see errors, read the error messages, double check settings, and try to fix things so you get a complete message trace (see below). If you ever need to re-run the message, you can re-send from the **Message Viewer** or drop the file again like you did in **Step 8-2**.

10. **Review the Message Trace:**

If everything has gone well, Step 3 of the trace will show the actual FHIR bundle that was sent to the local FHIR Server.

11. Review the steps in the message trace. Can you confirm whether or not the FHIR bundle posted successfully? What was the response status code from the FHIR Server?

12. You turned on `*FULL*` Trace so you could see detailed information like the actual FHIR bundle.

> This is a good technique for testing, but should be turned off once the interface is in production as FHIR bundles can get very big.

Click on Step 3 and on `View Raw =Contents`. Copy the JSON string in the tags, but do not include the XML tag.

13. Go to the FHIR validator at FHIR Validator.

13-1. Select the **Options** menu at the top. Then select the **FHIR Version (4.0.1)** and **Implementation Guides (hl7.fhir.us.core version 3.1.0)** and select **Add**.

This is the version of the FHIR Server we are checking against (remember when we configured those packages on the endpoint?)

13-2. Click back on **Validate** and paste the JSON into the **Enter Resource** window.

Select the **Validate** button at the bottom of the screen and review the errors and warnings.

> 14. Even though the FHIR bundle was accepted by the IRIS FHIR Server, there are many errors still when the message is validated against the official specifications. FHIR Servers will vary in how strict their validations are and whether they are configured to reject messages or accept them when there are non-conformance issues that are not deemed fatal.

** Note that SDA does include extensions which are beyond the US Core extensions so the validator doesn't recognize these. While these are conformant with R4, they are beyond what is typically expected and thus cause validation errors.

> 15. Now that you have completed an end-to-end interface, go back to the **FHIRDEMO** and try to figure out how to do each of these:

15-1. What happens when you send another HL7 message through? What about an ORU?

15-2. How can you configure the components to send individual resources rather than an entire bundle at once?

15-3. How can you configure the feed to use the **Message Control ID (MSH:10)** as the `PatientResourceId`? (You wouldn't necessarily want to do this, but it is good to see how the message changes when that's done)

Checking Results

You can query the FHIR Server with Postman or any other REST client. If none are available, curl is a command line utility that is available on most systems.

To query for Patient id 1009456

From a command line prompt:

```
curl http://ec2-18-219-162-44.us-east-
2.compute.amazonaws.com/irishealth/csp/healthshare/fhirdemo/fhir/r4/Patient/1009456/$everything -
u "train1:train1"
```

This sends the FHIR bundle response to the terminal, which is hard to read. If you want to redirect to a file so you can view and validate, use the following command to write the response to a file called **fhir_response.json**.

```
curl http://ec2-18-219-162-44.us-east-
2.compute.amazonaws.com/irishealth/csp/healthshare/fhirdemo/fhir/r4/Patient/1009456/$everything -
u "train1:train1" > fhir_response.json
```