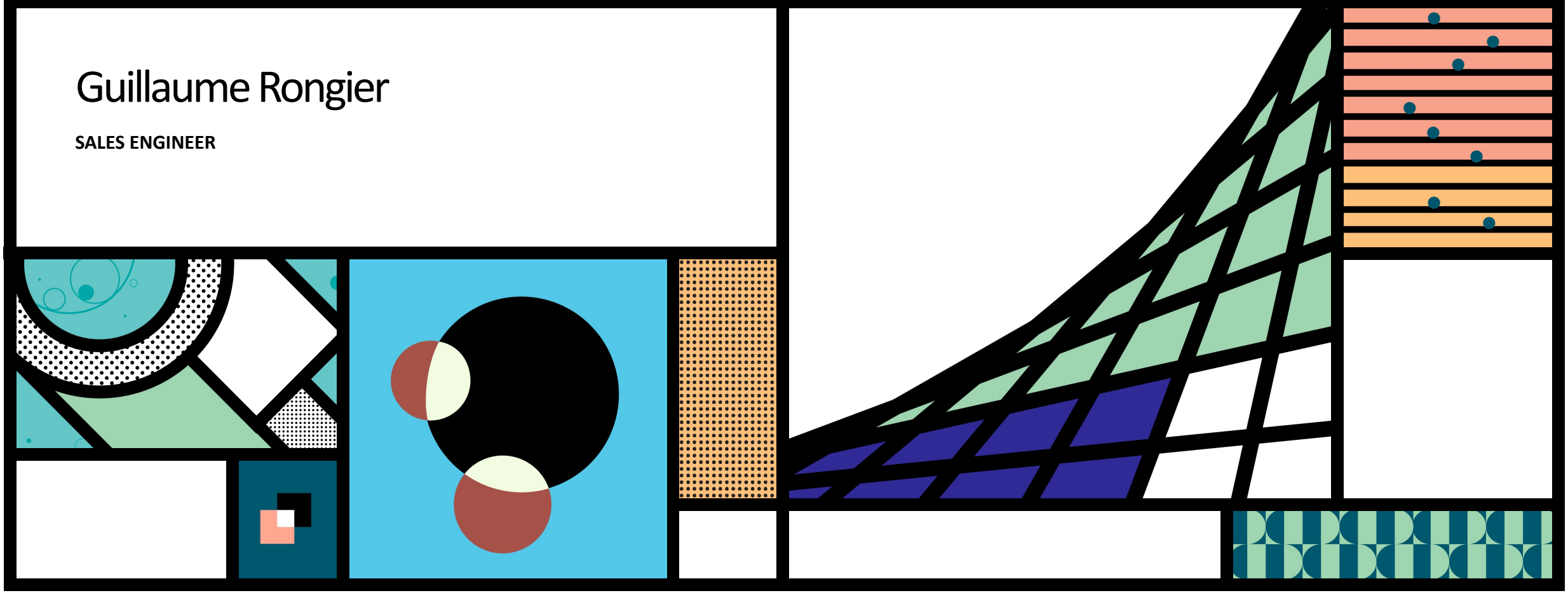
IRIS For Health Training

InterSystems®
Health | Business | Government

Guillaume Rongier

SALES ENGINEER

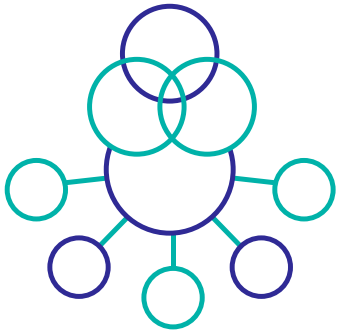


Index

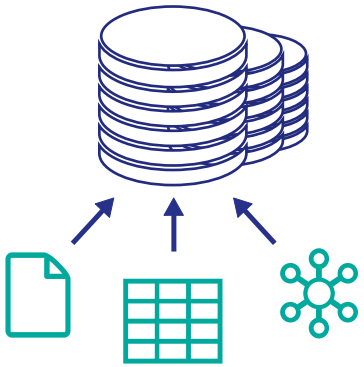
Introduction

IRIS For Health

Unifier pour Simplifier



InterSystems IRIS
Interoperability



InterSystems IRIS
Multi Model Database

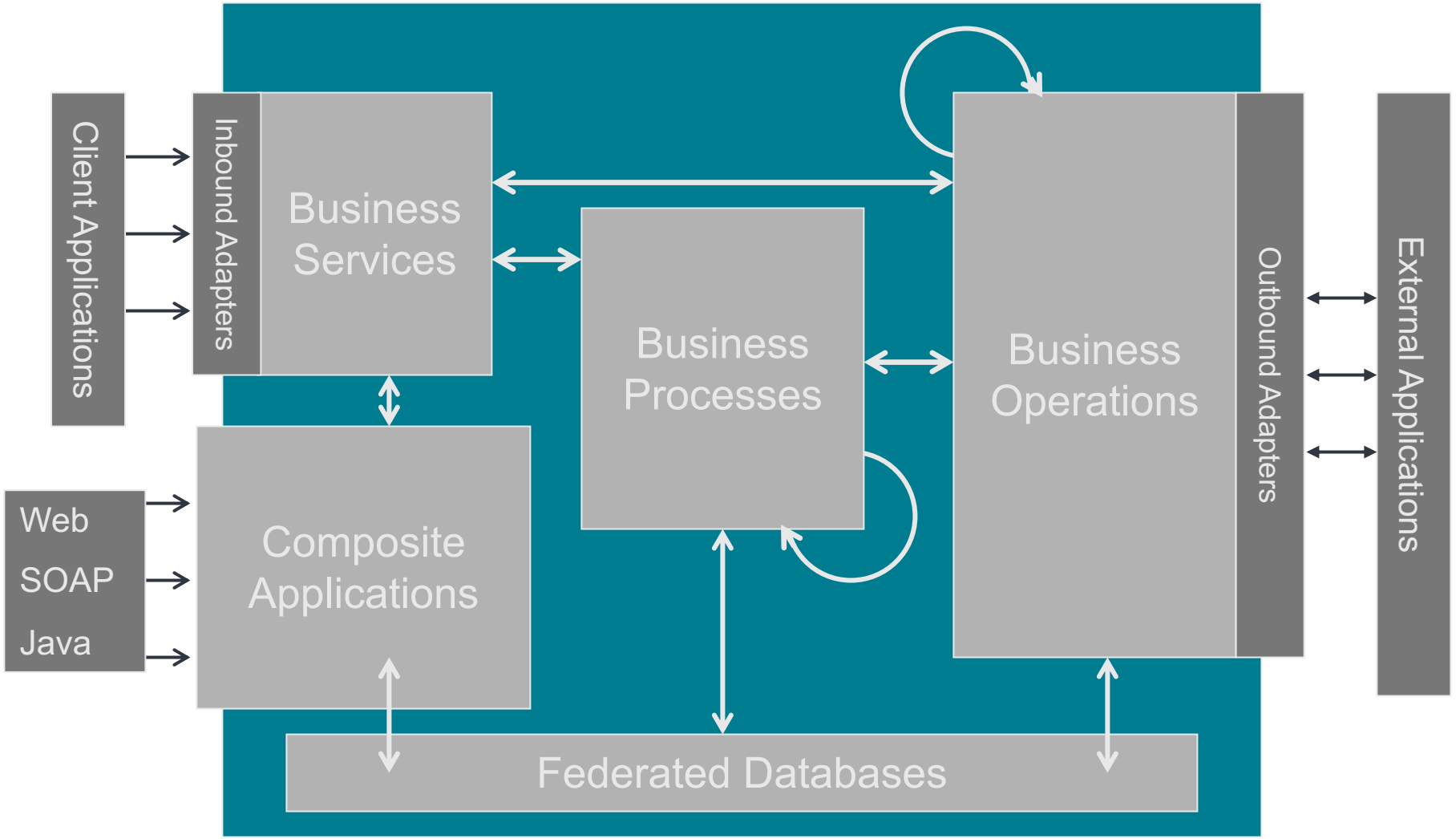
InterSystems IRIS
Analytics

IRIS Mantra

Bring code to data

Not data to code

Interoperability Framework



 | |

Installation

Dockerfile

```
ARG IMAGE=intersystemsdc/irishealth-community:2020.4.0.547.0-zpm
FROM $IMAGE

ARG IRIS_PASSWORD

USER root
WORKDIR /opt/irisapp
RUN chown ${ISC_PACKAGE_MGRUSER}:${ISC_PACKAGE_IRISGROUP} /opt/irisapp
USER ${ISC_PACKAGE_MGRUSER}

COPY . /opt/irisapp
COPY iris.script /tmp/iris.script


# run iris and initial
RUN iris start IRIS \
&& iris session IRIS < /tmp/iris.script \
&& iris stop IRIS quietly
```


Iris.script

```
zn "%SYS"
Do ##class(Security.Users).UnExpireUserPasswords("*")
zn "HSLIB"
// Install a Foundation namespace and change to it
Do ##class(HS.HC.Util.Installer).InstallFoundation("TRAINING")
zn "TRAINING"

// Install FHIR repo
Set appKey = "/api/fhir"
Set strategyClass = "HS.FHIRServer.Storage.Json.InteractionsStrategy"
Set metadataConfigKey = "HL7v40"
```


Docker-compose

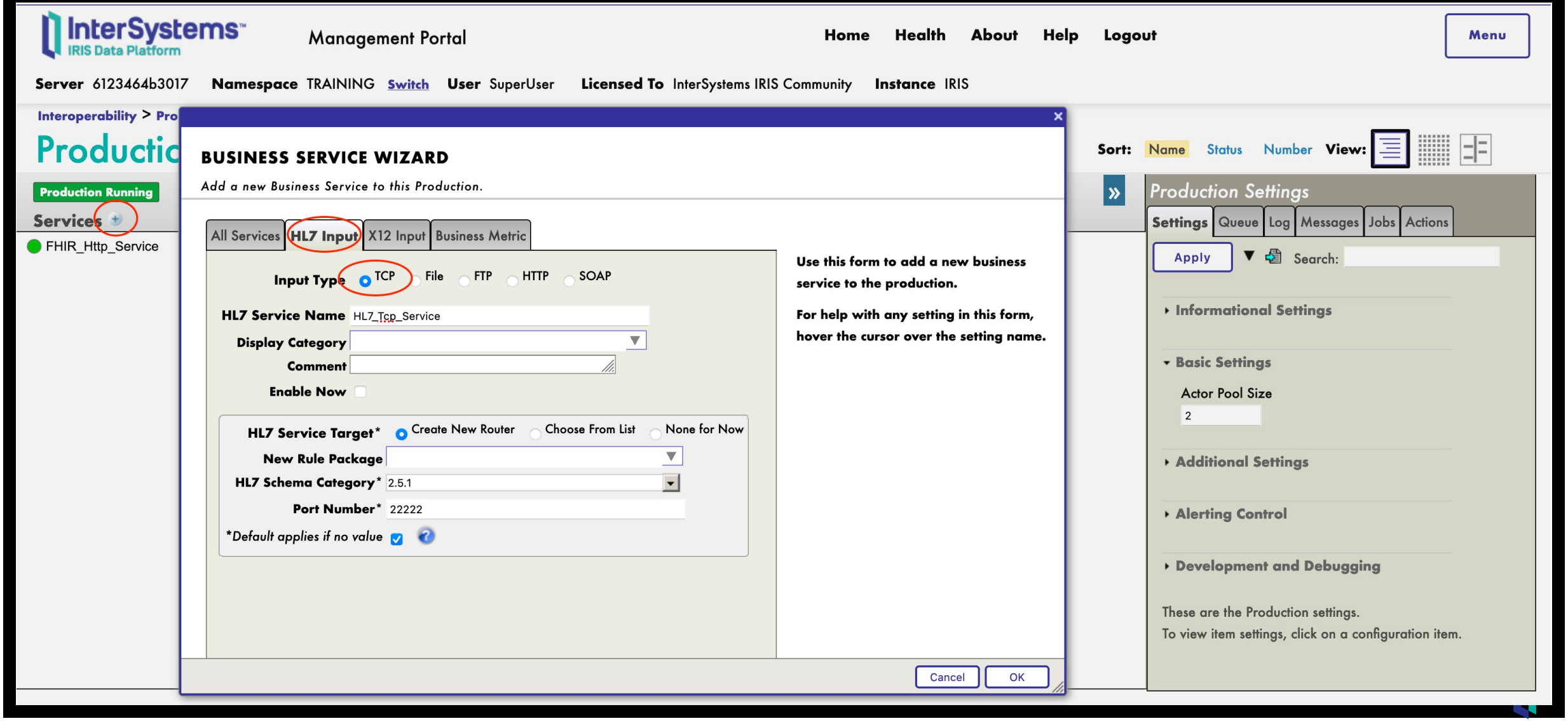
```
version: '3.6'

services:
 iris:
   build:
     context: .
     dockerfile: dockerfile
    args:
      - IRIS_PASSWORD=$IRIS_PASSWORD

  restart: always

  env_file:

   - .env

  ports:
    - 32782:1972
    - 32783:52773
    - 32784:53773
  volumes:
    - ./:/irisdev/app
```


Launch it !

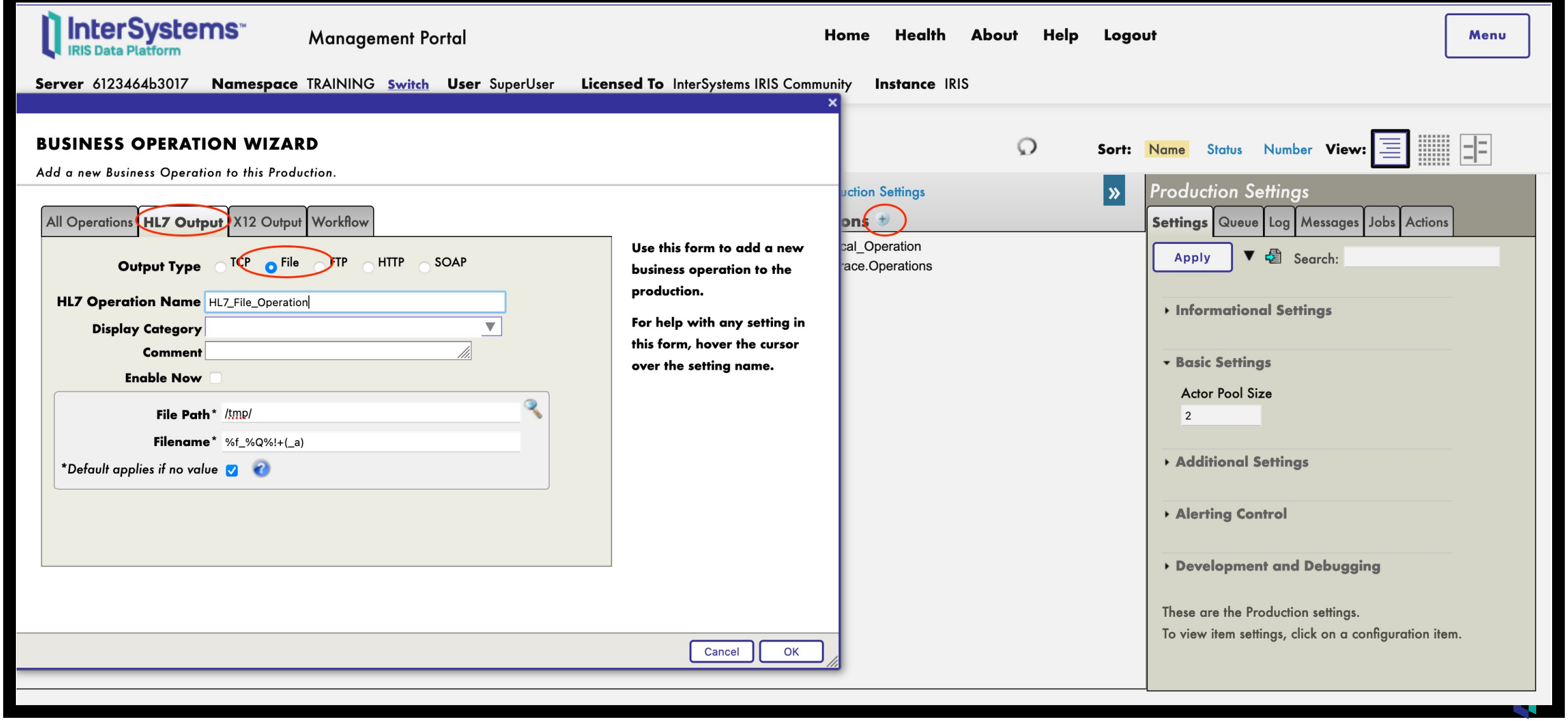
Docker compose up

Premier flux HL7v2

First HL7 TCP Service

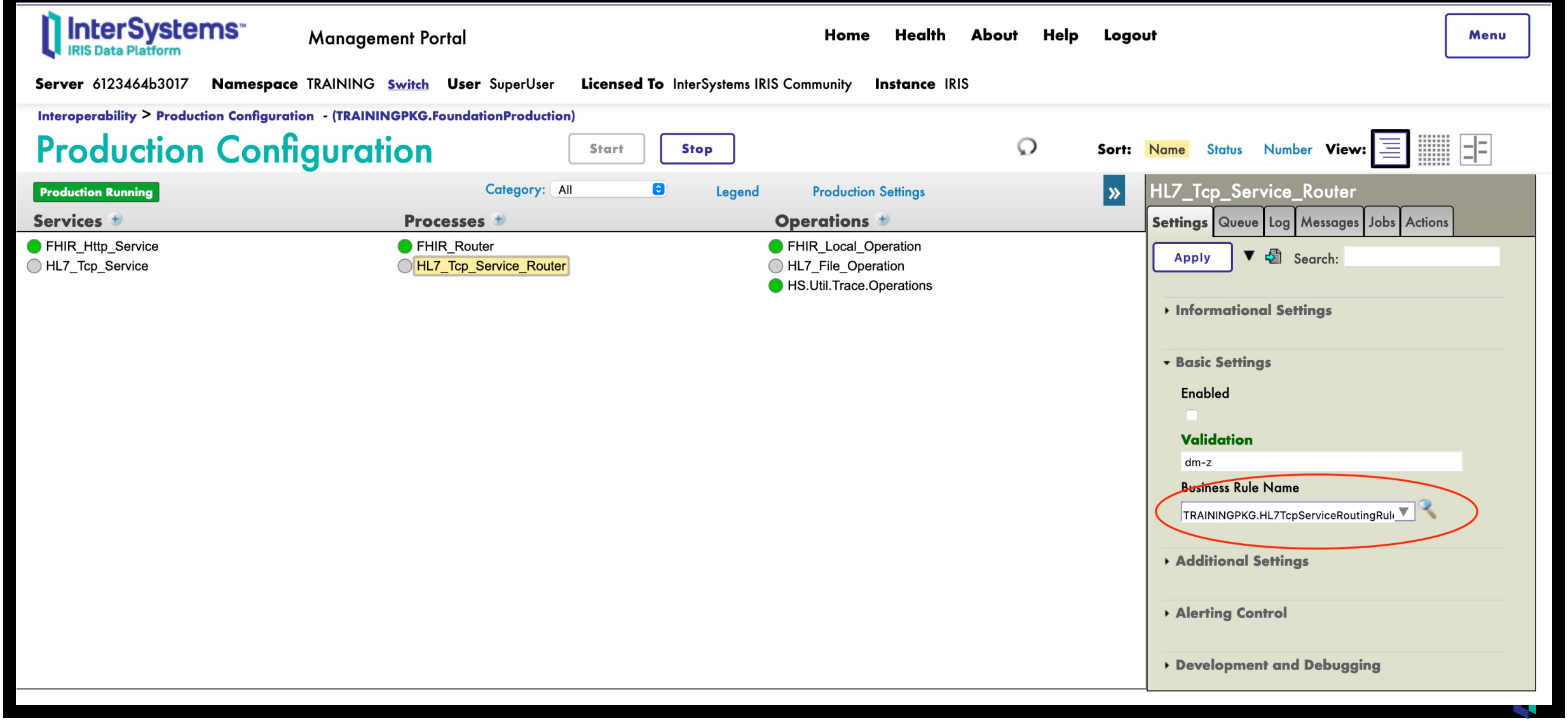


HL7 Operation

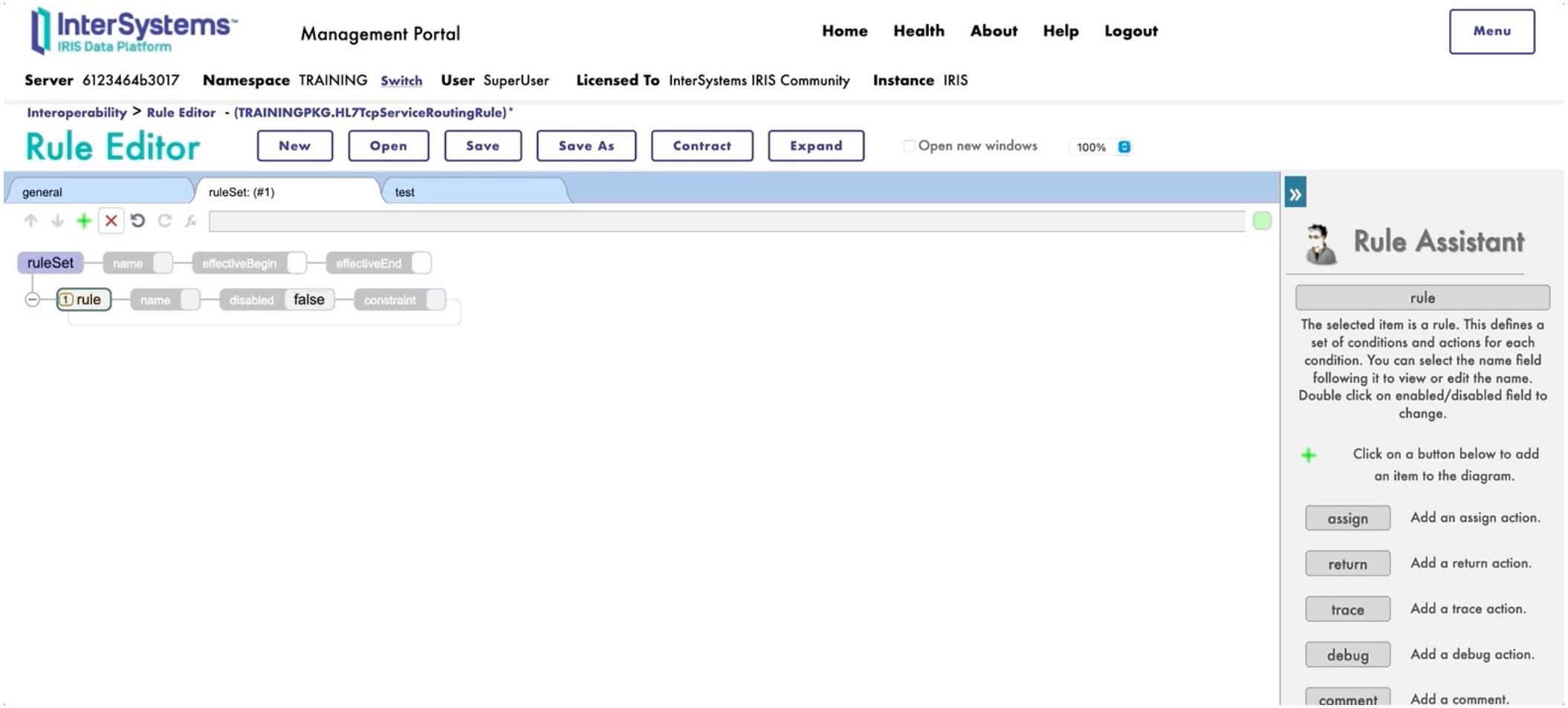


 | |

HL7 router



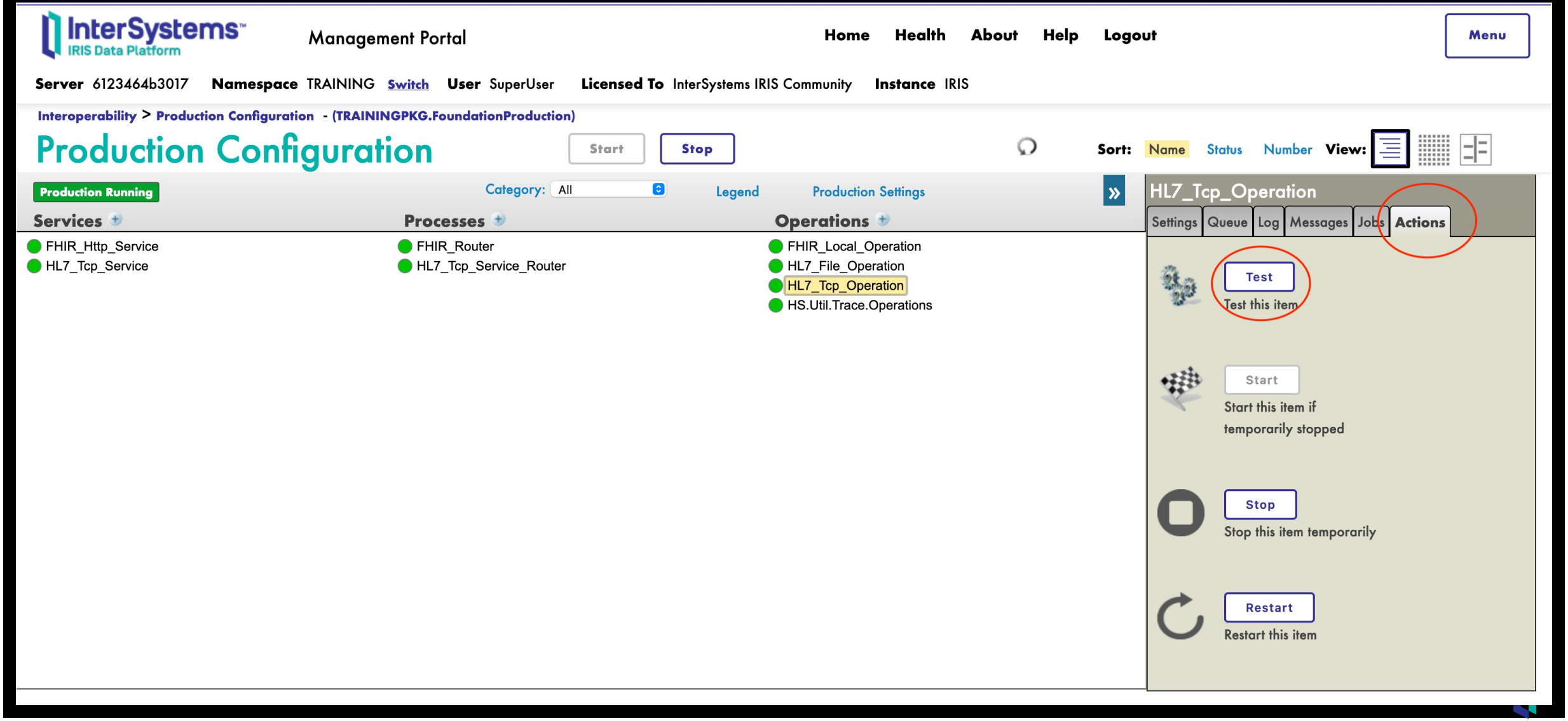
HL7 router



Test It !

1. Create an Operation
    1. Name : HL7_Tcp_Operation
    2. Port : 2222
    3. Host : localhost

Test It !



Client/Server

Client / Server

Save you work in git.

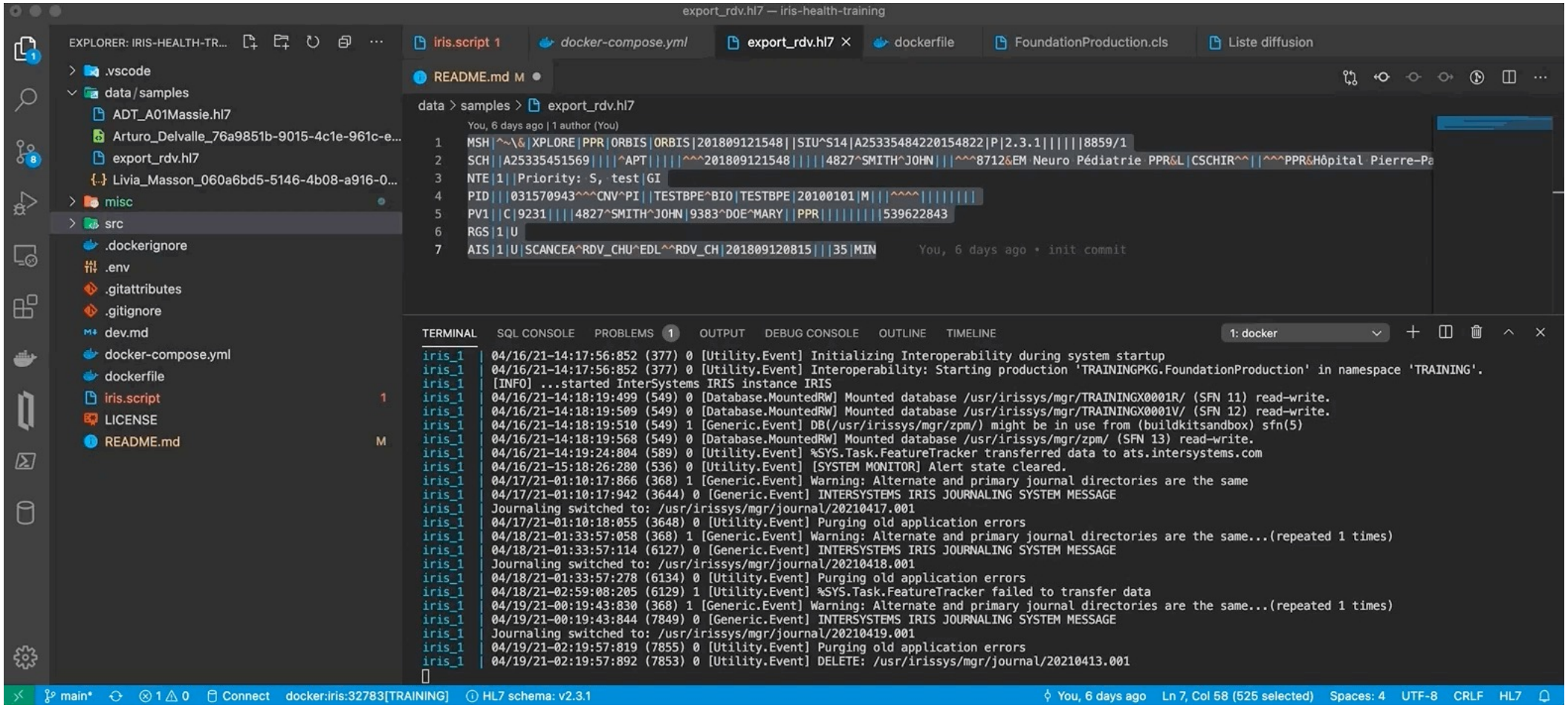
First remember the mantra :

- Bring code to data
- Not data to code

Second :

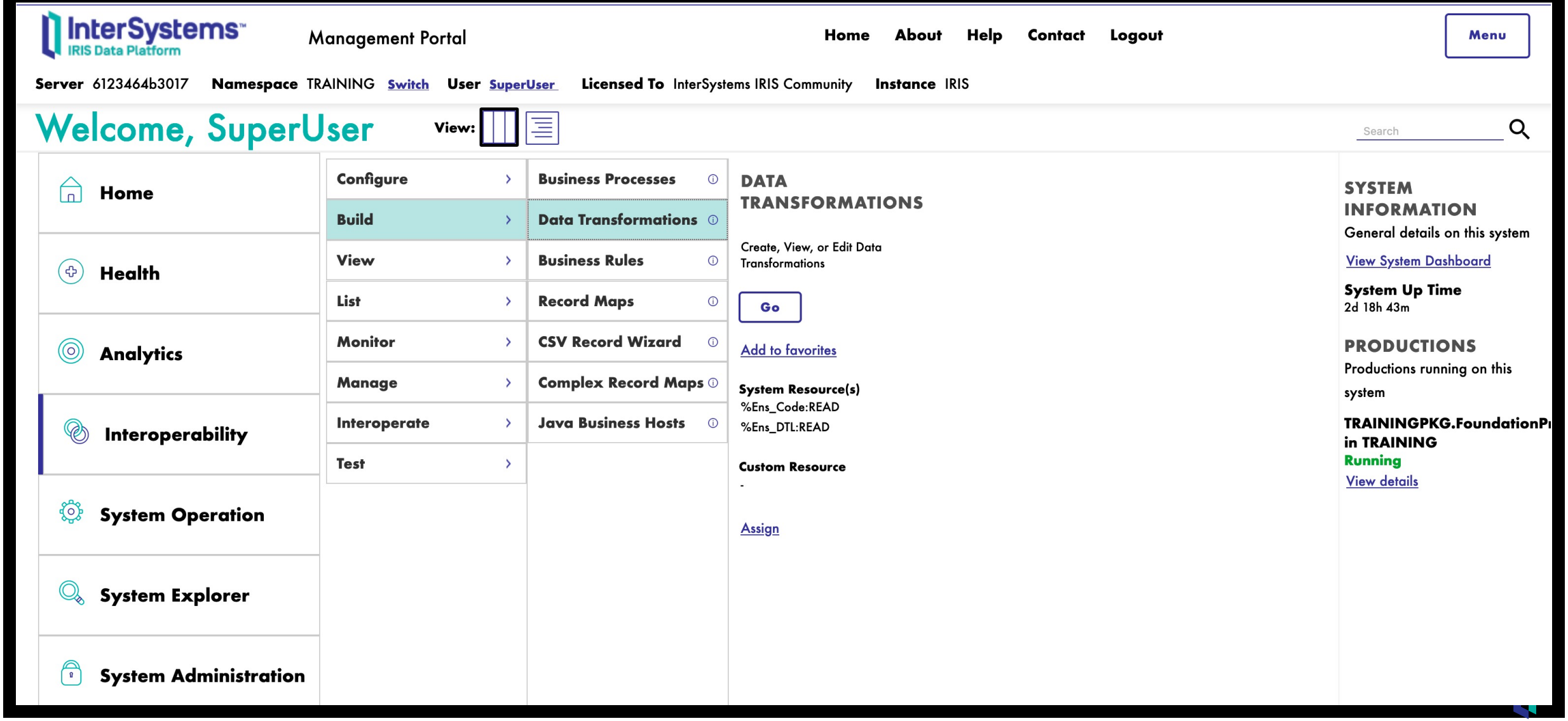
- Its a database

Client / Server



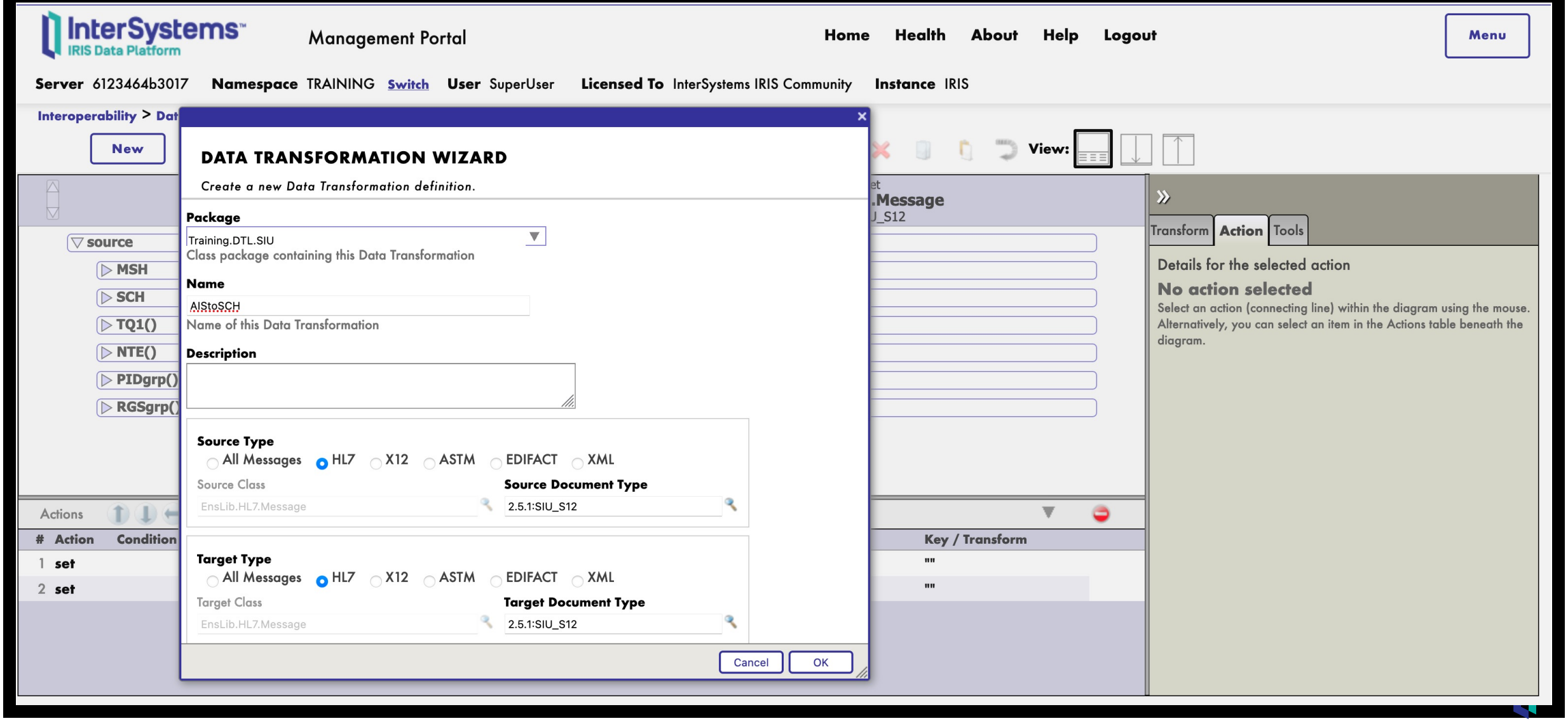
DTL

DTL

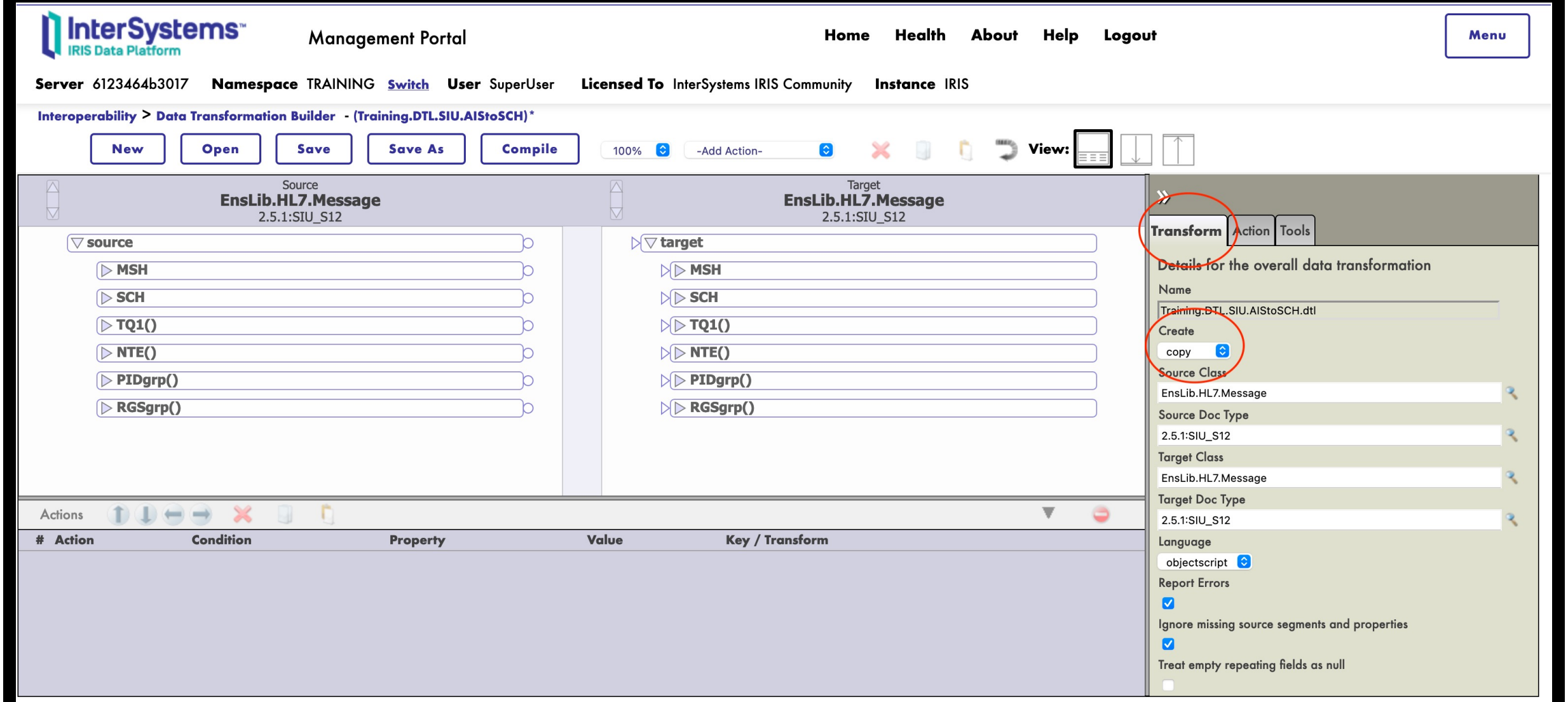


 | |

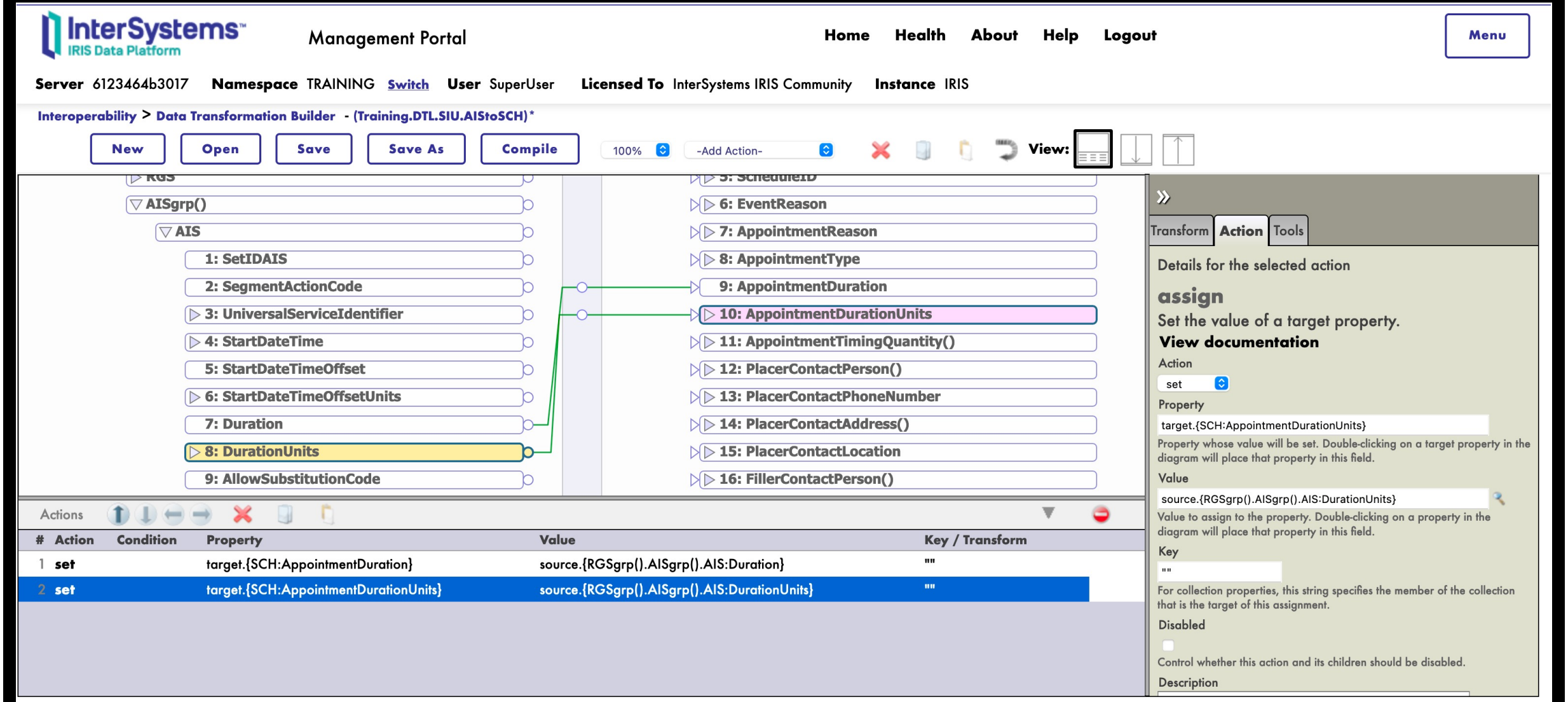
DTL



DTL



DTL



 | |

Test It !

1. Compile
2. Test It !
    1. Tools Tab
        1. Test
3. Add DTL to router
    1. Test It !

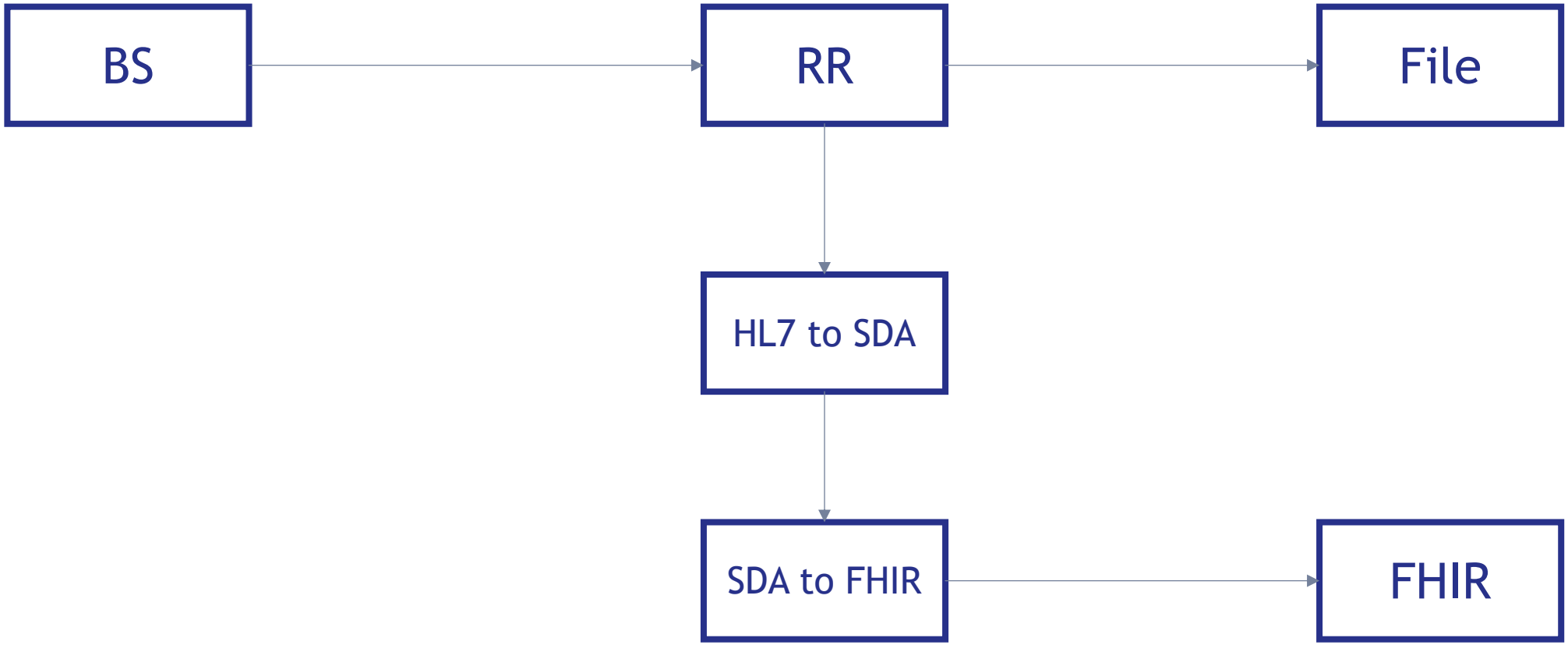
 | |

HL7v2 To FHIR

Demographics Only

Overview, what we have done

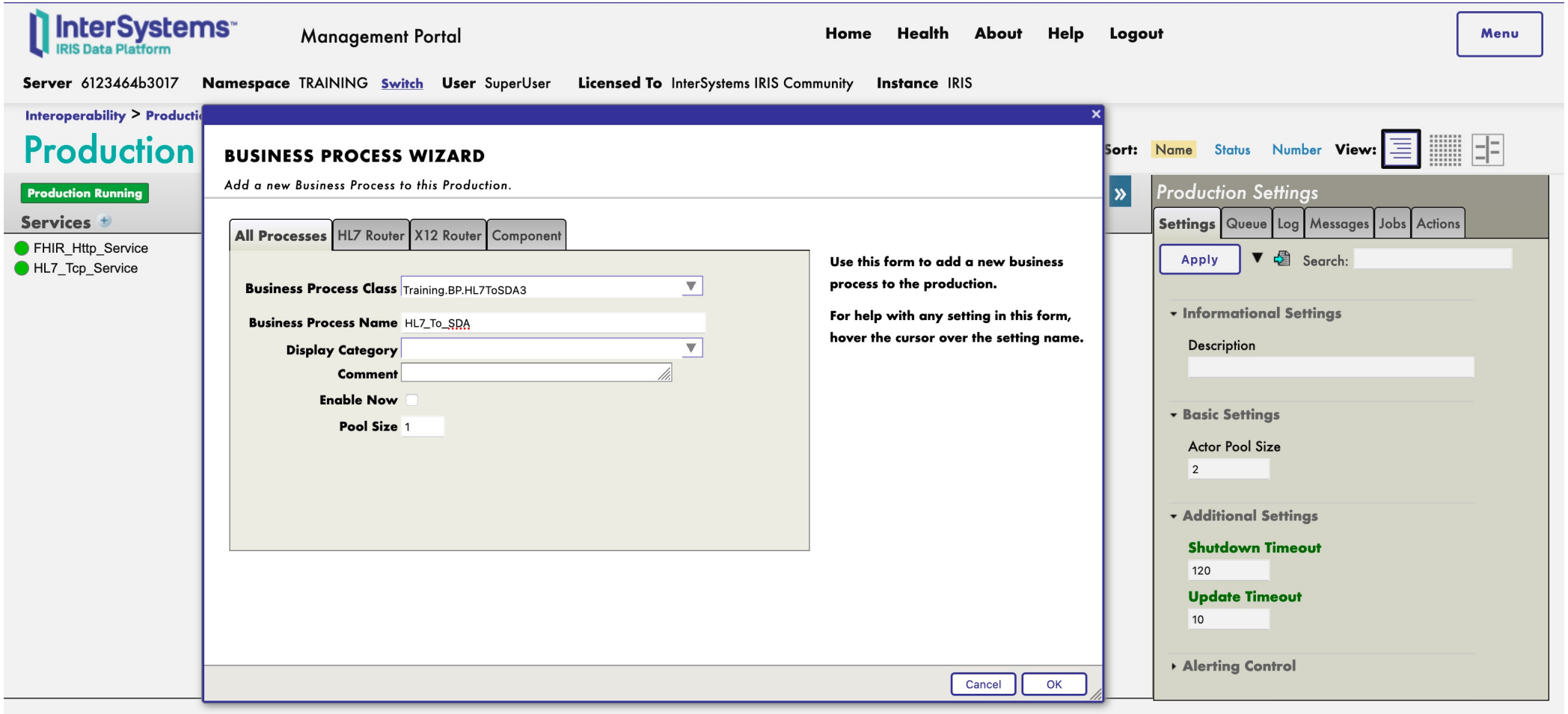
Overview, what we plan to do



 | |

Instantiate the new BPs

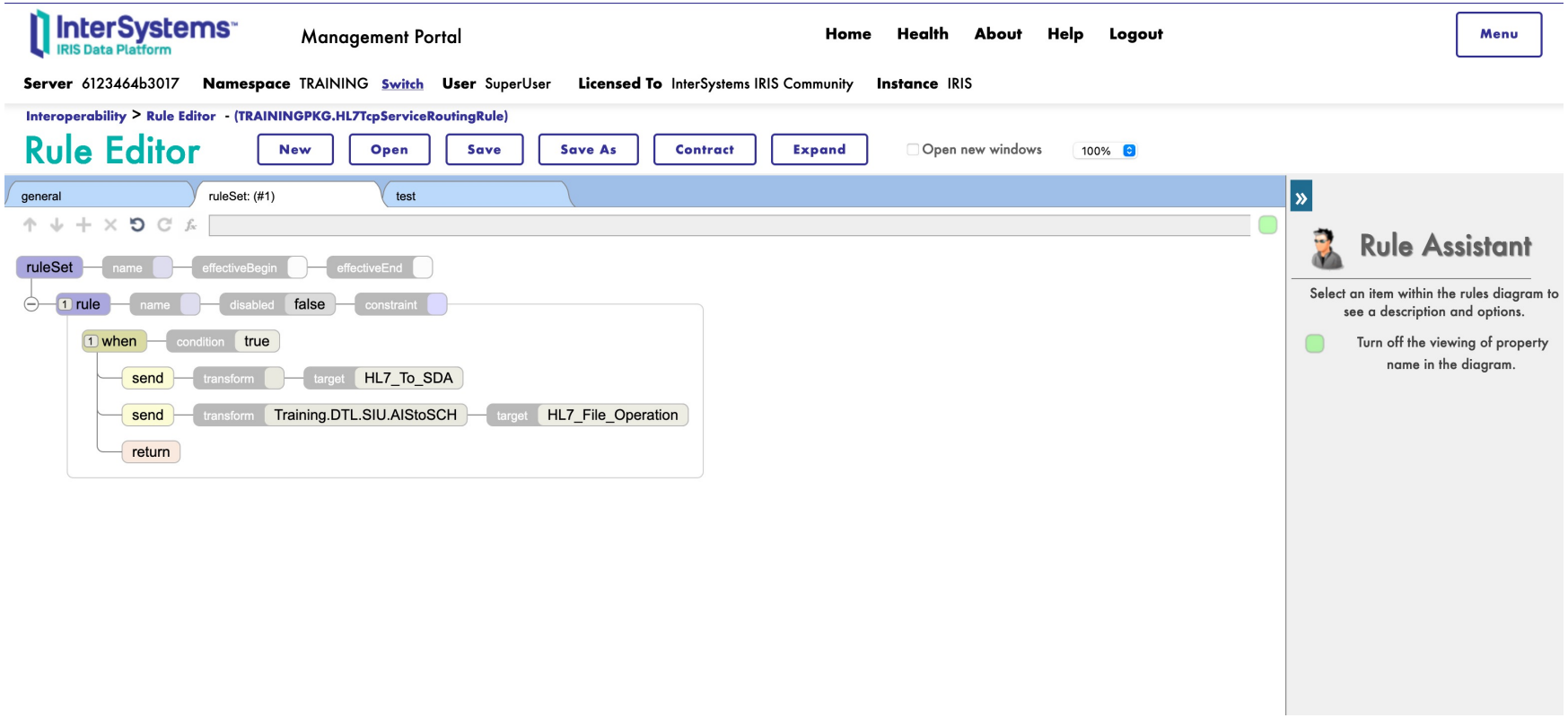
Instantiate **Training.BP.HL7ToSDA3** as **HL7_To_SDA**



Instantiate the new BPs

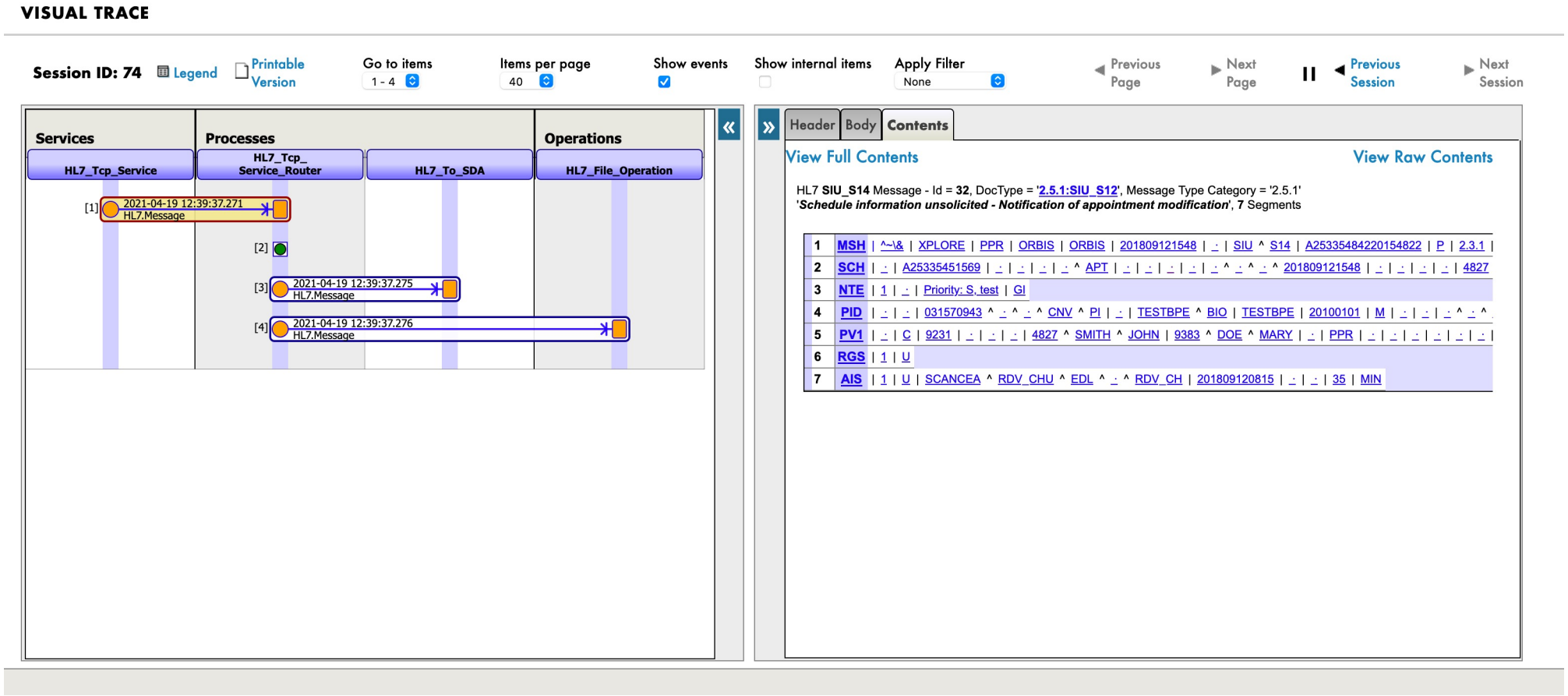
Instantiate **HS.FHIR.DTL.Util.HC.SDA3.FHIR.Process** as **SDA_To_FHIR**

Add route to **HL7_To_SDA to TRAININGPKG.HL7TcpServiceRoutingRule**



Test It !

What happen ?



Use the helper tools in Training.BP.HL7ToSDA3

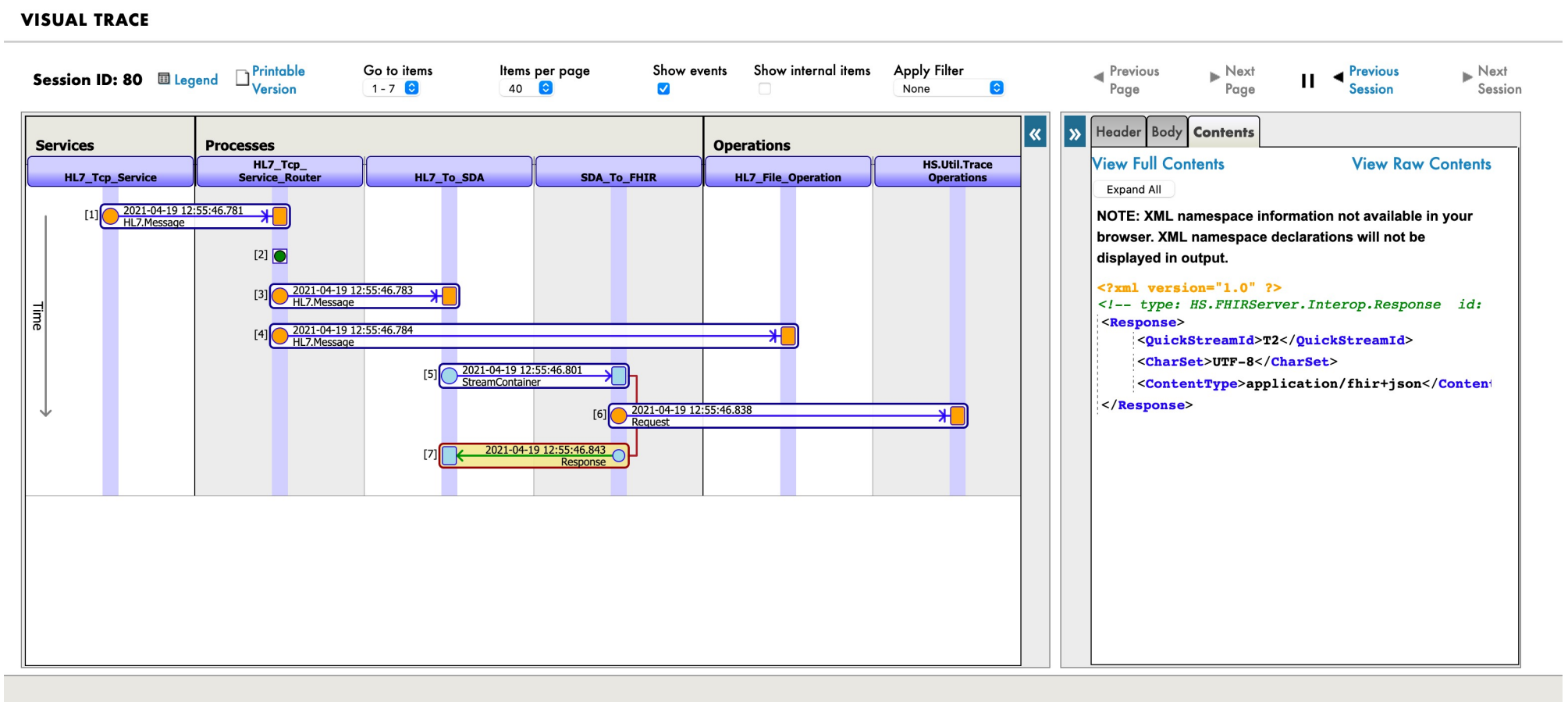
The helper is **Training.Tools.HL7toSDA3Demographics** who extends **HS.Hub.Standalone.HL7.HL7ToSDA3**

```
/// Transforms an HL7 message to SDA, an internal healthcare format for
InterSystems IRIS for Health.
Method OnRequest(pRequest As EnsLib.HL7.Message, Output pResponse As
Ens.Response) As %Status
{
set tSC = $$$OK
try {
//set tSC = ##class(Training.Gateway.HL7.HL7ToSDA3).GetSDA(pRequest,.tSDA)
//set tSC=
##class(Training.Hub.Standalone.HL7.HL7ToSDA3).GetSDAContainer(pRequest,.t
Container)
//set tSC = tContainer.ToQuickXMLStream(.tSDA)
//set tSC =
..SendRequestSync(..TargetConfigName,##class(Ens.StreamContainer).%New(tSD
A),.pResponse)
} catch ex {
set tSC = ex.AsStatus()
}
quit tSC
}
```

```
/// Transforms an HL7 message to SDA, an internal healthcare format for
InterSystems IRIS for Health.
Method OnRequest(pRequest As EnsLib.HL7.Message, Output pResponse As
Ens.Response) As %Status
{
set tSC = $$$OK
try {
//set tSC =
##class(Training.Gateway.HL7.HL7ToSDA3).GetSDA(pRequest,.tSDA)
set tSC=
##class(Training.Tools.HL7toSDA3Demographics).GetSDAContainer(pRequest,.t
Container)
set tSC = tContainer.ToQuickXMLStream(.tSDA)
set tSC =
..SendRequestSync(..TargetConfigName,##class(Ens.StreamContainer).%New(tS
DA),.pResponse)
} catch ex {
set tSC = ex.AsStatus()
}
quit tSC
}
```


Test It !

What happen ?



Use custom DTLs

1. Overload method **GetTransformClass** from **Training.Tools.HL7toSDA3Demographics**

Objective :

1. Definie new package of custom DTL
2. If no custom
    1. Use stand DTLs

 | |

Example

```
set tTransformPackage =
"HS.Hub.Standalone.HL7.DTL"

set pTransformClass =
tTransformPackage_"."_pTransformClass
```

```
set tTransformPackage =
"Training.Tools.DTL"

set tTransformClass =
tTransformPackage_"."_pTransformClass


if '$$$comClassDefined(tTransformClass)

{

set tTransformPackage =
"HS.Hub.Standalone.HL7.DTL"

set tTransformClass =
tTransformPackage_"."_pTransformClass

}


set pTransformClass = tTransformClass
```


Import DTLs to our custom package

Go to InterSystems explorer -> **HS.Hub.Standalone.HL7.DTL**

Export classes and save the one you want overload in **Training.Tools.DTL**

 | |

HL7v2 To FHIR

Full Experience

Before starting

ObjectScript Commands

- Set (a variable)
- Do (do an action, execute a static method)
- Write (standard output, like a print(“”) in python)
- Kill (unload a variable)
- Quit (exit a block, method, loop, …)
- Merge (merge two globals, like set but for multi dimensional variable)

(https://docs.intersystems.com/irisforhealthlatest/csp/docbook/Doc.View.cls?KEY=RCOS_COMMANDS)

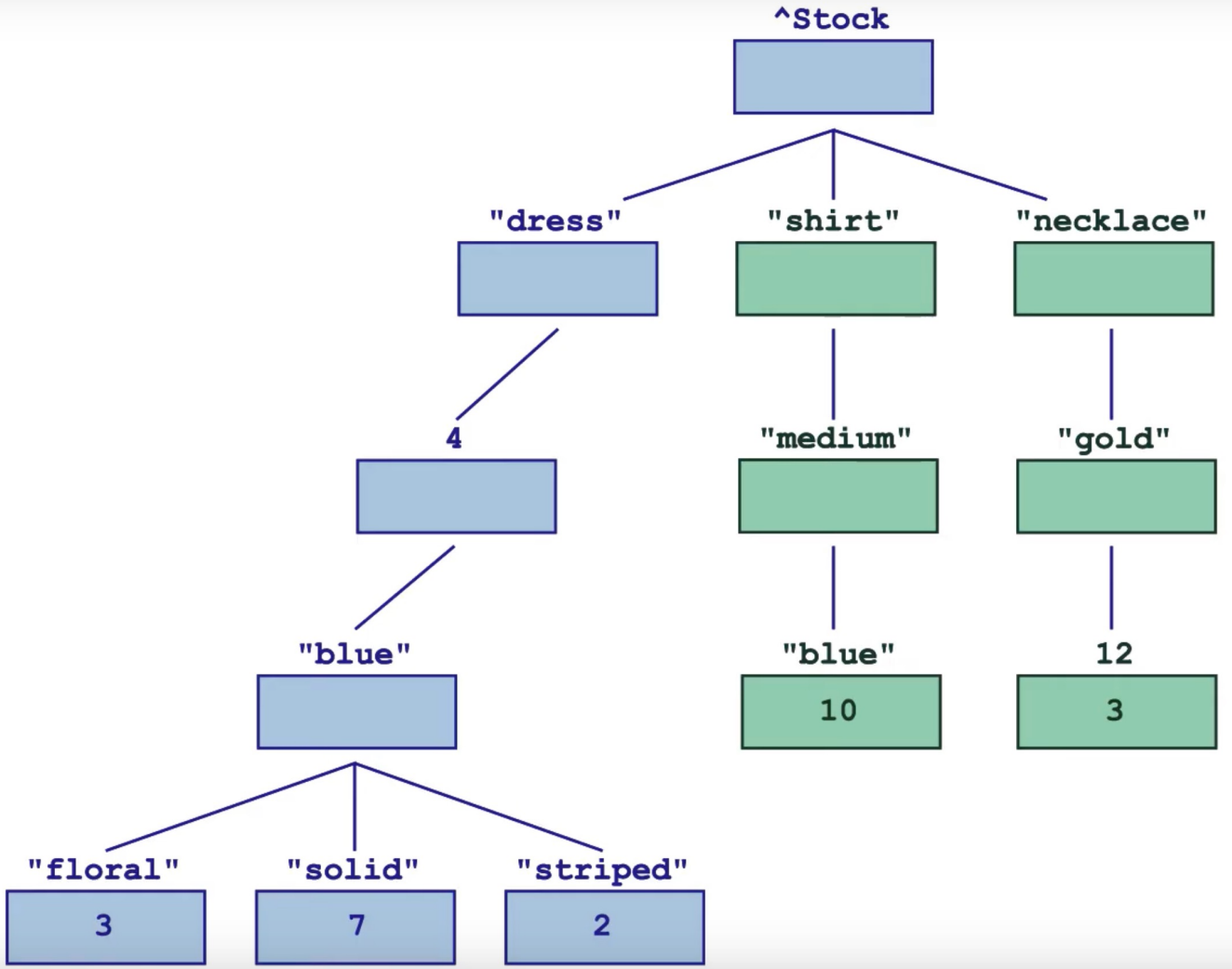
Before starting

Globals

```
^Stock("dress", 4, "blue", "floral") = 3
^Stock("dress", 4, "blue", "solid") = 7
^Stock("dress", 4, "blue", "striped") = 2
^Stock("shirt", "M", "blue") = 10
^Stock("necklace", "gold", 12) = 3
```

Before starting

Globals



 | |

Let’s Go !

Have a look at **HS.Gateway.HL7.HL7ToSDA3**

Objective of this helper class :

- Be fast
- Write the SDA XML Object
- Parse HL7 Message as a Global

Docs :

http://localhost:32783/csp/healthshare/TRAINING/HS.UI.HL7.cls

Let’s Debug It !

Edit **Training.BP.HL7ToSDA3**

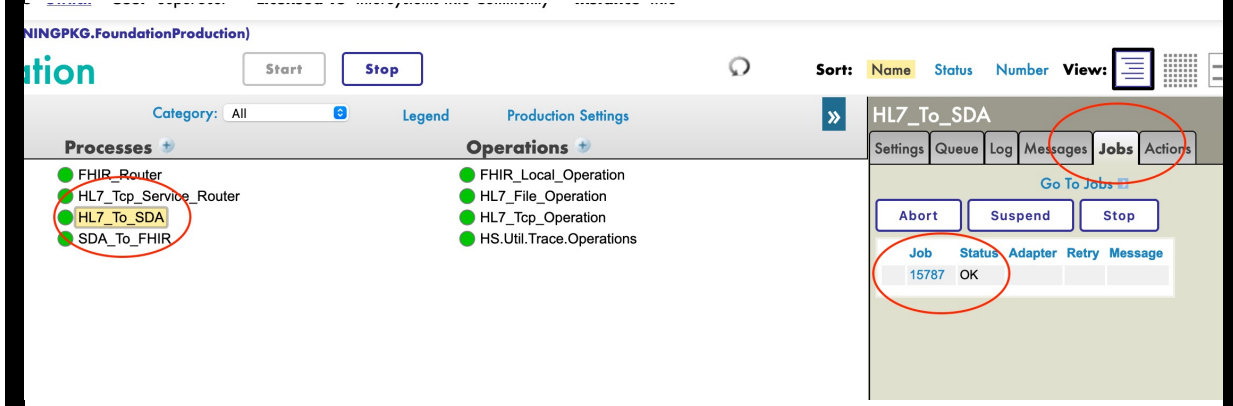
Add a switch for selecting the right helper (don’t forget the property and setting)

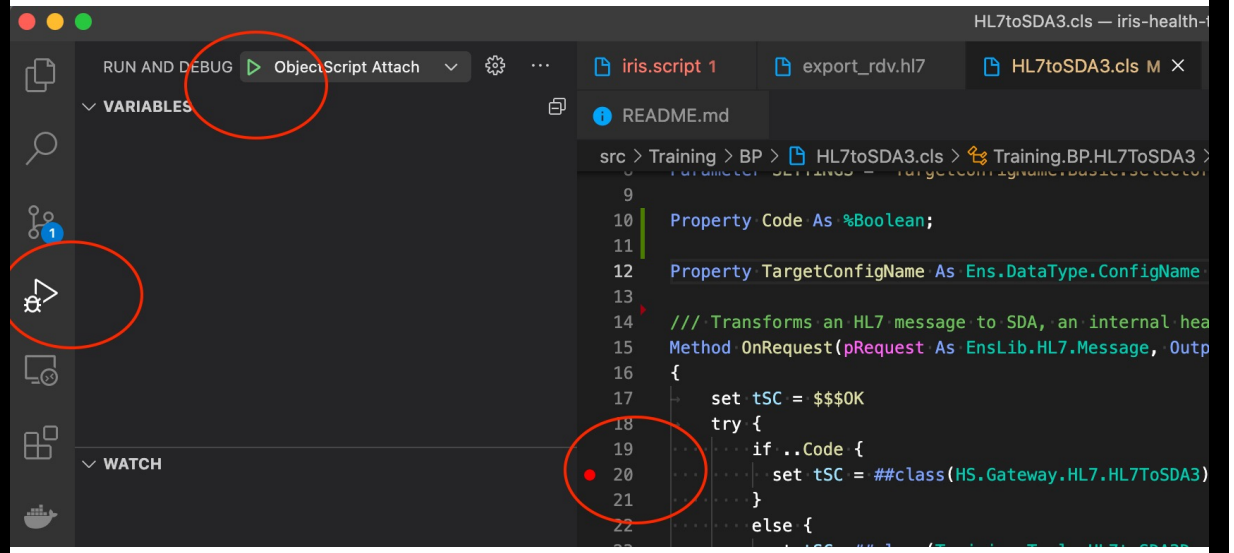
```
/// Transforms an HL7 message to SDA, an internal healthcare format for
InterSystems IRIS for Health.
Method OnRequest(pRequest As EnsLib.HL7.Message, Output pResponse As
Ens.Response) As %Status
{
set tSC = $$$OK
try {
  //set tSC =
##class(Training.Gateway.HL7.HL7ToSDA3).GetSDA(pRequest,.tSDA)
  set tSC=
##class(Training.Tools.HL7toSDA3Demographics).GetSDAContainer(pRequest,.t
Container)
  set tSC = tContainer.ToQuickXMLStream(.tSDA)
  set tSC =
..SendRequestSync(..TargetConfigName,##class(Ens.StreamContainer).%New(tS
DA),.pResponse)
} catch ex {
  set tSC = ex.AsStatus()
}
quit tSC
}
```

```
/// Transforms an HL7 message to SDA, an internal healthcare format for
InterSystems IRIS for Health.
Method OnRequest(pRequest As EnsLib.HL7.Message, Output pResponse As
Ens.Response) As %Status
{
set tSC = $$$OK
try {
  if ..Code {
    set tSC = ##class(HS.Gateway.HL7.HL7ToSDA3).GetSDA(pRequest,.tSDA)
  }
  else {
    set tSC=
##class(Training.Tools.HL7toSDA3Demographics).GetSDAContainer(pRequest,.tC
ontainer)
    set tSC = tContainer.ToQuickXMLStream(.tSDA)
}

set tSC =
..SendRequestSync(..TargetConfigName,##class(Ens.StreamContainer).%New(tSD
A),.pResponse)

} catch ex {
  set tSC = ex.AsStatus()
}
quit tSC
}
```

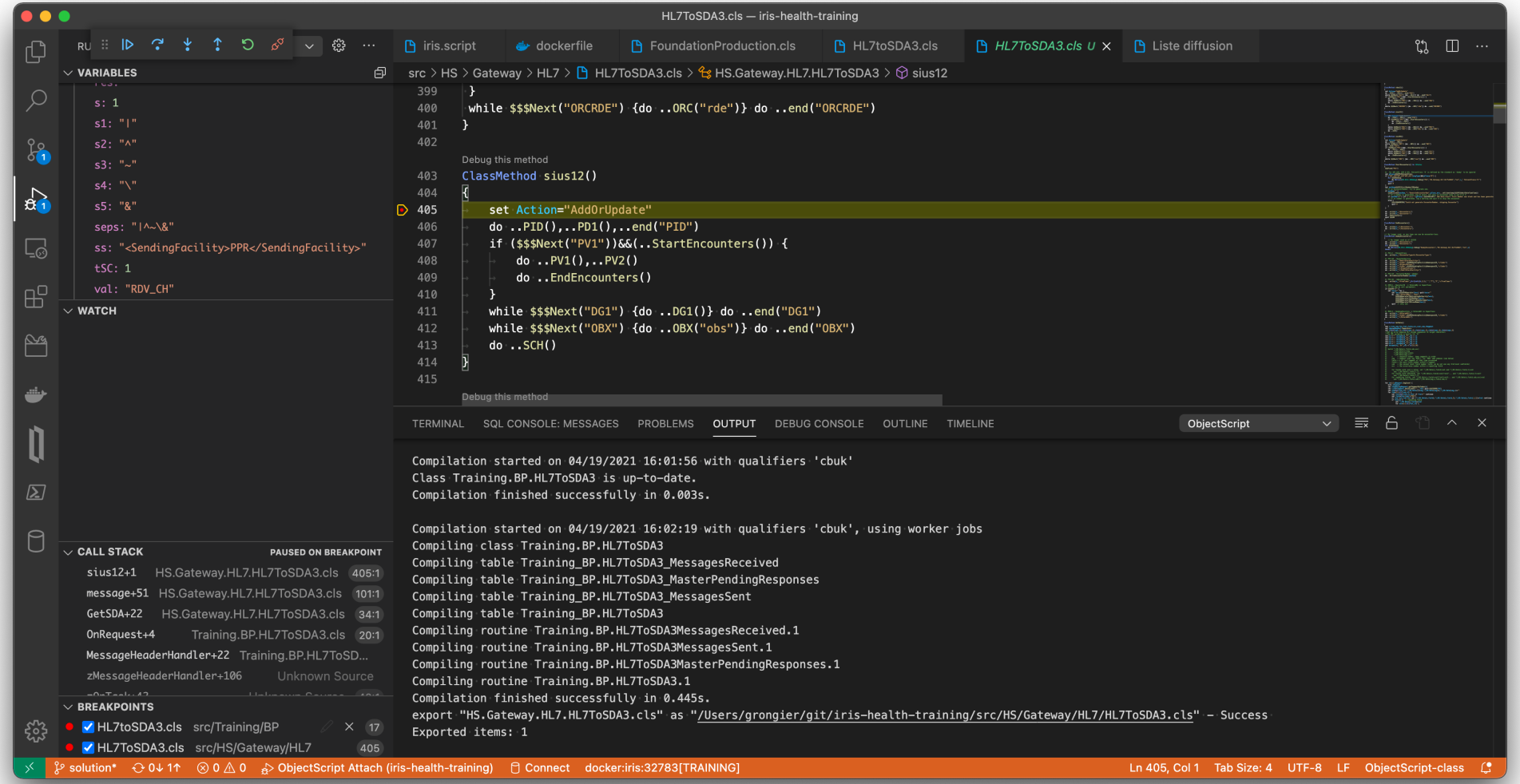
 | |

Attach process to VsCode





Attach process to VsCode



Now overload this class

Let’s create a class **Training.Tools.HL7toSDA3Full** who extends **HS.Gateway.HL7.HL7ToSDA3**

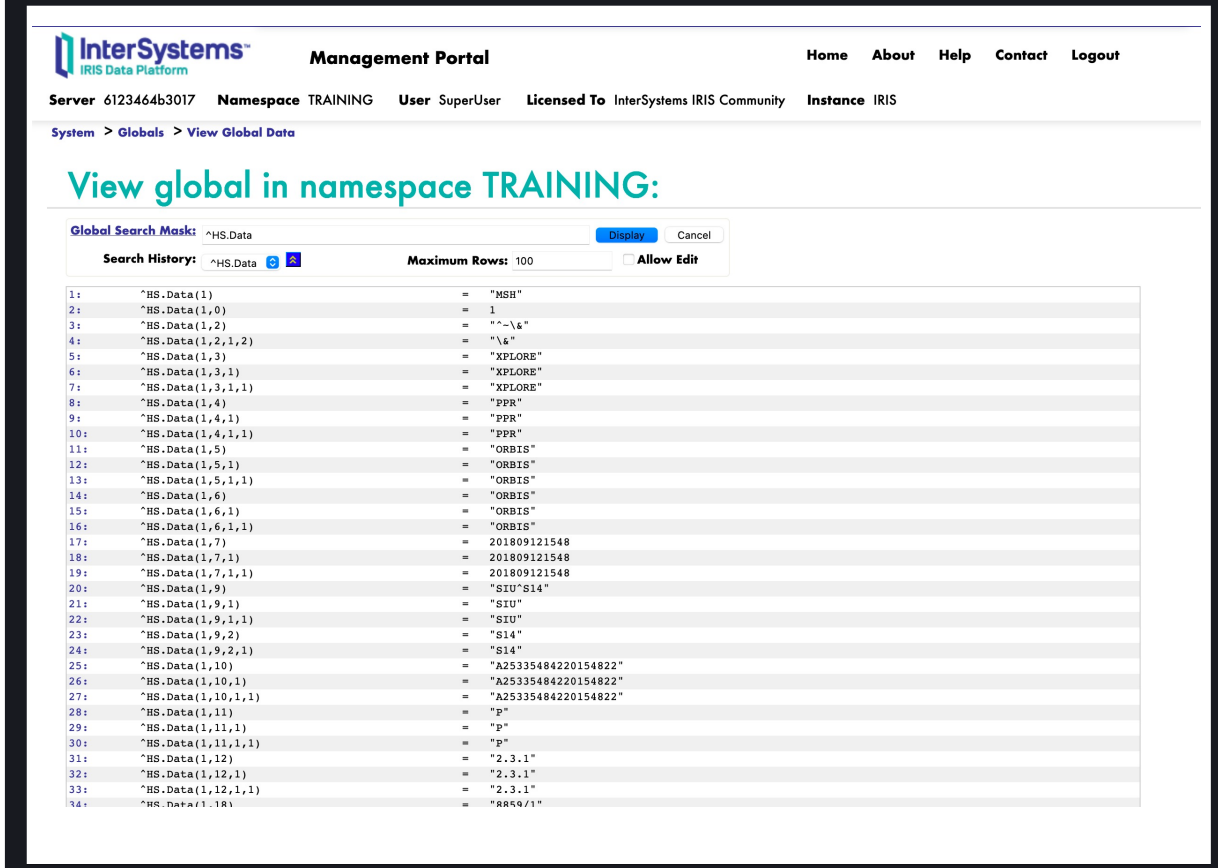
Then overload **GetSDA**

```
Class Training.Tools.HL7toSDA3Full Extends HS.Gateway.HL7.HL7ToSDA3
{

}
```

```
ClassMethod GetSDA(pRequest As EnsLib.HL7.Message, Output pXML As
%Stream.GlobalCharacter, pLogAlerts As %Boolean = 1, pObservationMode As
%Boolean = 0, pMultiLineOBXCompatibilityMode As %Boolean = 0,
pKeepDuplicateOBXIdentifiers As %Boolean = 0) As %Status
{
  set sc =
##super(pRequest,.pXML,pLogAlerts,pObservationMode,pMultiLineOBXCompatibil
ityMode,pKeepDuplicateOBXIdentifiers)
  kill ^HS.Data
  merge ^HS.Data=^||HS.Data
  quit sc
}
```


See the global ^HS.Data

http://localhost:32783/csp/sys/exp/UtilExpGlobalView.csp?$ID2=HS.Data&$NAMESPACE=TRAININ G&$NAMESPACE=TRAINING



What do we see here ?

We see the HL7 message put in a global.

Every field is now in a key value pair.

Example :

```
^HS.Data(2)      =      "SCH"
^HS.Data(2,0)    =      1
^HS.Data(2,2)    =      "A25335451569"
```

Segment 2 is an SCH

In field 0 we have 1

In field 2 we have "A25335451569"

 | |

Exercice

We have an SIU^S14

In SCH we don’t have the duration, it’s in the segment AIS.

Overload **HS.Gateway.HL7.HL7ToSDA3** to take in account the duration from AIS segment even if it’s not present in SCH.

Solution :

https://github.com/grongierisc/iris-health-training/commit/016bc05b5ab60904504f66c0e50279136ac6a492

https://github.com/grongierisc/iris-health-training/commit/cad88ce78e2aa4a713cfd02ee346372d2fee1507

SDA to FHIR

Documentation

http://localhost:32783/csp/healthshare/TRAINING/SchemaMap.Tool.UI.View.zen

Mapping or Information SDA3 → FHIR4 | List SDA3 → FHIR4 Transformed | View SDA3 → FHIR4 Lookup Tables | Help | FAQ | Syntax | SDA3 Code Tables

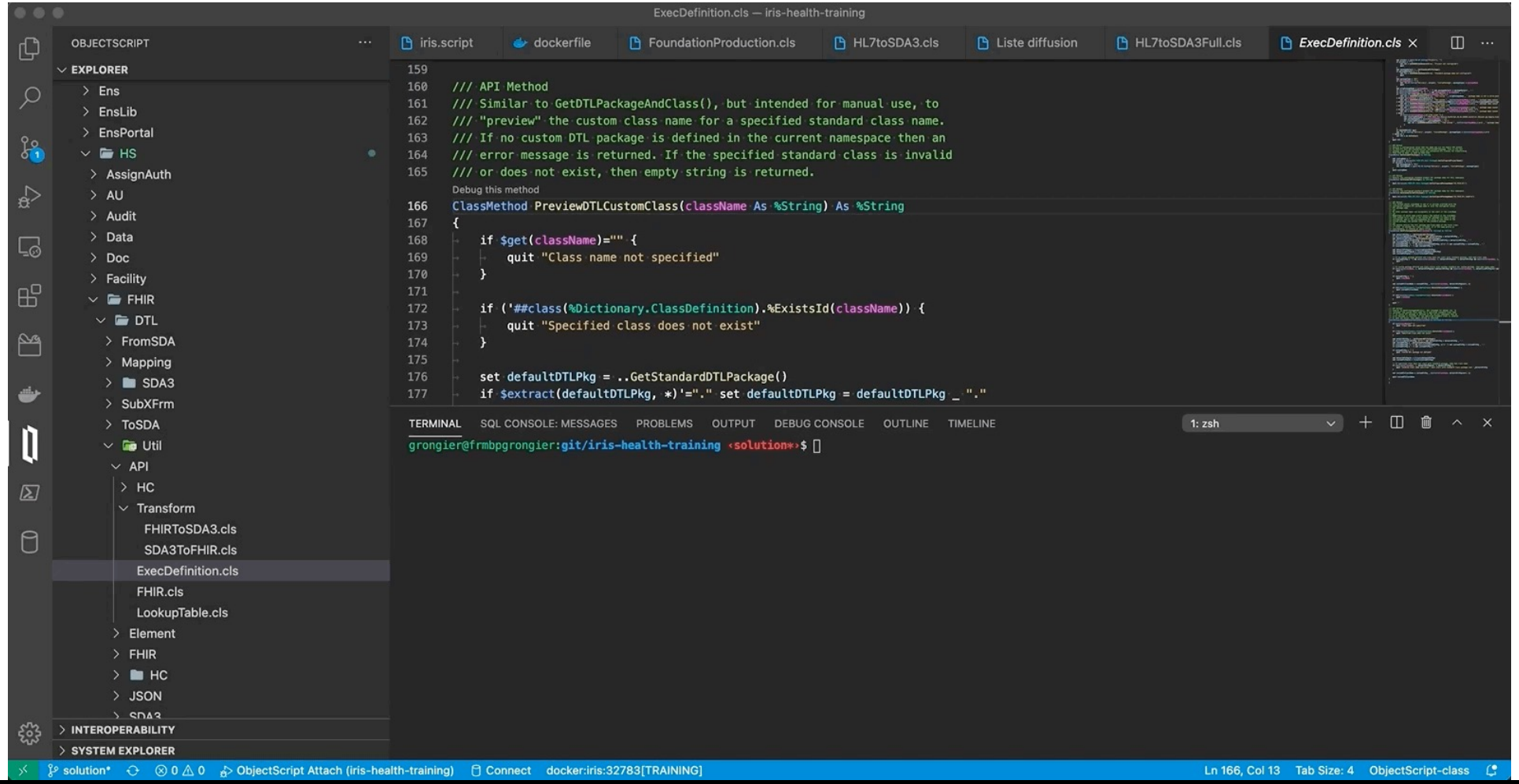
FHIR4 by Name **FHIR4 by Category** **HS.SDA3 Classes** Appointment

This page allows you to view mappings that generate transformations from SDA to FHIR R4.

SDA3 → FHIR4 SDA3 Source	SDA3 Data Type	SDA3 Card.	Actions	Map Issues	FHIR4 Target	Notes
HS.SDA3.Appointment	HS.SDA3.SuperClass				Appointment:extension	SDA3 Specification
					Appointment:participant	
HS.SDA3.Appointment:CareProvider	HS.SDA3.CodeTableDetail.CareProvider	0..1			Appointment:participant	default participant status will be "accepted"
HS.SDA3.Appointment:CustomPairs	HS.SDA3.NVPair	0..*				
HS.SDA3.Appointment:EncounterNumber	%String (MAXLEN=220)	0..1			Appointment:extension	
HS.SDA3.Appointment:EnteredAt	HS.SDA3.CodeTableDetail.Organization	0..1			Appointment:extension	
HS.SDA3.Appointment:EnteredBy	HS.SDA3.CodeTableDetail.User	0..1			Appointment:extension	
HS.SDA3.Appointment:EnteredOn	HS.SDA3.TimeStamp	0..1			Appointment:created	
HS.SDA3.Appointment:ExternalId	%String (MAXLEN=220)	0..1			Appointment:identifier	
HS.SDA3.Appointment:FillerApptId	%String	0..1			Appointment:identifier	
HS.SDA3.Appointment:FillerOrderId	%String	0..1			Appointment:extension	
HS.SDA3.Appointment:FromTime	HS.SDA3.TimeStamp	0..1			Appointment:start	
					Appointment:end	
HS.SDA3.Appointment:Location	HS.SDA3.CodeTableDetail.HealthCareFacility	0..1			Appointment:participant	
HS.SDA3.Appointment:NoShow	HS.SDA3.Boolean	0..1			Appointment:extension	
HS.SDA3.Appointment:Notes	%String (MAXLEN="")	0..1			Appointment:comment	

Implementing Custom DTLs

1. Go in terminal mode :



Implementing Custom DTLs

1. https://docs.intersystems.com/irisforhealthlatest/csp/docbook/Doc.View.cls?KEY=HXFHIR_transforms#HXFHIR_transforms_customize_dtl_package

2. check if a custom DTL package already exists, enter:

Write ##class(HS.FHIR.DTL.Util.API.ExecDefinition).GetCustomDTLPackage()

3. If the custom DTL package does not already exist, enter the following command, replacing HS.Local.FHIR.DTL with the name of your custom DTL package:

zw ##class(HS.FHIR.DTL.Util.API.ExecDefinition).SetCustomDTLPackage("Training.Tools.DTL")

4. Get the new name of HS.FHIR.DTL.SDA3.vR4.Appointment.Appointment

Write ##class(HS.FHIR.DTL.Util.API.ExecDefinition).PreviewDTLCustomClass("HS.FHIR.DTL.SDA3.vR4.Appointment.Appointment")

Implementing Custom DTLs

Open DTL **HS.FHIR.DTL.SDA3.vR4.Appointment.Appointment** and save as **Training.Tools.DTL.SDA3.vR4.Appointment.Appointment**

Test a DTL

Add this code at the beginning of the transformation :

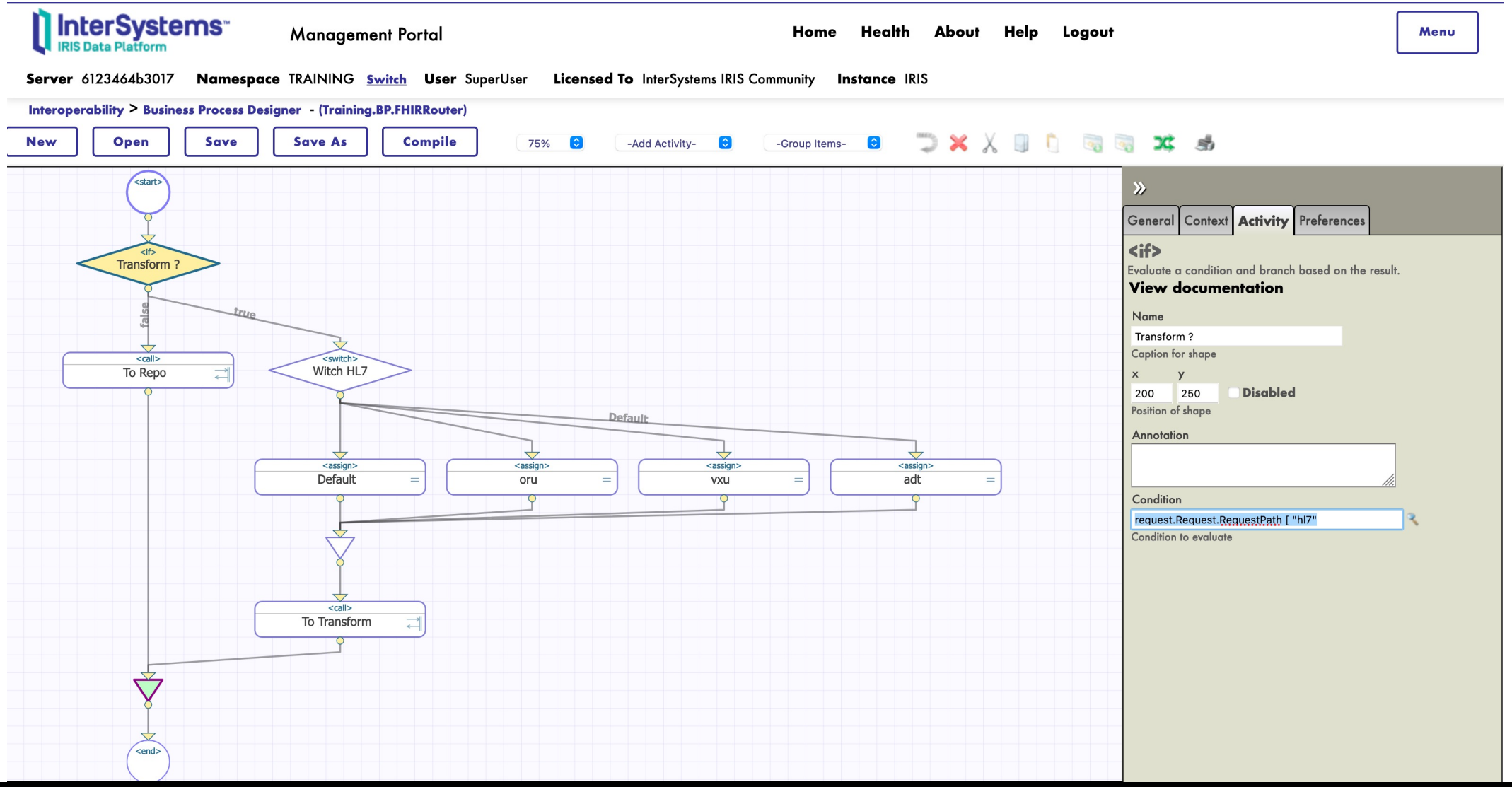
```
set schema = ##class(HS.FHIRServer.Schema).LoadSchema("R4")

set aux("transformer") = ##class(HS.FHIR.DTL.Util.API.Transform.SDA3ToFHIR).%New(schema)
```

 | |

BONUS !

FHIR -> SDA

Have a look at FHIR_Router



 | |

Instantiate the new BPs

Instantiate **HS.FHIR.DTL.Util.HC.FHIR.SDA3.Process** as **FHIR_To_SDA**

Create a new Business process to transform SDA to HL7

Training.BP.SDA3toHL7 Extends Ens.BusinessProcess

Overload Method OnRequest(pRequest As Ens.Request, Output pResponse As Ens.Response) As %Status

Training.BP.SDA3toHL7

```
Class Training.BP.SDA3toHL7 Extends Ens.BusinessProcess
{

Method OnRequest(pRequest As Ens.Request, Output pResponse As
Ens.Response) As %Status
{
  set sc = $$$OK

  Quit sc
}
```


Exercice

Complete it, base on :

https://github.com/grongierisc/iris-healthtoolkit-service

 | |

Training.BP.SDA3toHL7

```
Method OnRequest(pRequest As Ens.Request, Output pResponse As Ens.Response) As %Status

{

set sc = $$$OK


Try {

//Cast the SDA Stream as an XML Document

set tVDoc= ##class(EnsLib.EDI.XML.Document).ImportFromString(pRequest.ContentStream, .sc)

//Transform it as un ADT

do ##class(HS.Gateway.SDA3.SDA3ToHL7.ADTA0N).Transform(tVDoc,.tHL7)

//Send the tHL7 Message

set sc = ..SendRequestSync("HL7_File_Operation",tHL7,.pResponse)

if '$ISOBJECT(pResponse){

set pResponse = ##class(HS.FHIRServer.Interop.Response).%New()

}

}

Catch ex {

Set sc=ex.AsStatus()

}


Quit sc

}
```


The power behind what matters.

InterSystems®
Health | Business | Government

Thank you.