CGI Supply Chain Demonstrator

Installation

- Install InterSystems IRIS 2025.1 with the appropriate license (SC key with Vector Search enabled)

- Create SC namespace (create a a new required resource for the DB)

- Install the SCO zpm package (Documentation)

- Install the AI chatbot backend:
  zpm
  repo -delete-all
  ```
  repo -r -n registry -url http://sc-repo.iscinternal.com:52773/registry
  install isc-supply-chain-ai-copilot
  exit
  ```

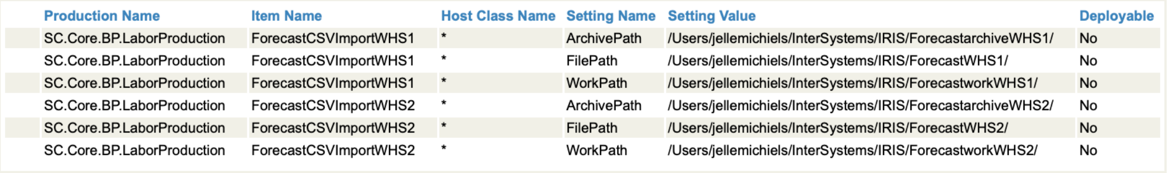
- Create a namespace per warehouse in the simulation (e.g. WHS1 and WHS2)

- Generate or copy the CSVs with product and sales information from here

- Create 3 folders per WHS (e.g. on Windows for WHS1:
  C:\InterSystems\IRIS\Mgr\ForecastWHS1\,
  C:\InterSystems\IRIS\Mgr\ForecastarchiveWHS1\,
  C:\InterSystems\IRIS\Mgr\ForecastworkWHS1\)

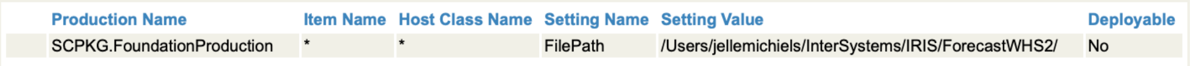
  In a 2 WHS scenario, 6 folders should be created

  Make sure you have the following system default settings in the SC namespace



Production Name	Item Name	Host Class Name	Setting Name	Setting Value	Deployable
SC.Core.BP.LaborProduction	ForecastCSVImportWHS1	*	ArchivePath	/Users/jellemichiels/InterSystems/IRIS/ForecastarchiveWHS1/	No
SC.Core.BP.LaborProduction	ForecastCSVImportWHS1	*	FilePath	/Users/jellemichiels/InterSystems/IRIS/ForecastWHS1/	No
SC.Core.BP.LaborProduction	ForecastCSVImportWHS1	*	WorkPath	/Users/jellemichiels/InterSystems/IRIS/ForecastworkWHS1/	No
SC.Core.BP.LaborProduction	ForecastCSVImportWHS2	*	ArchivePath	/Users/jellemichiels/InterSystems/IRIS/ForecastarchiveWHS2/	No
SC.Core.BP.LaborProduction	ForecastCSVImportWHS2	*	FilePath	/Users/jellemichiels/InterSystems/IRIS/ForecastWHS2/	No
SC.Core.BP.LaborProduction	ForecastCSVImportWHS2	*	WorkPath	/Users/jellemichiels/InterSystems/IRIS/ForecastworkWHS2/	No

  And the following in the namespaces for WHS1 and WHS2:

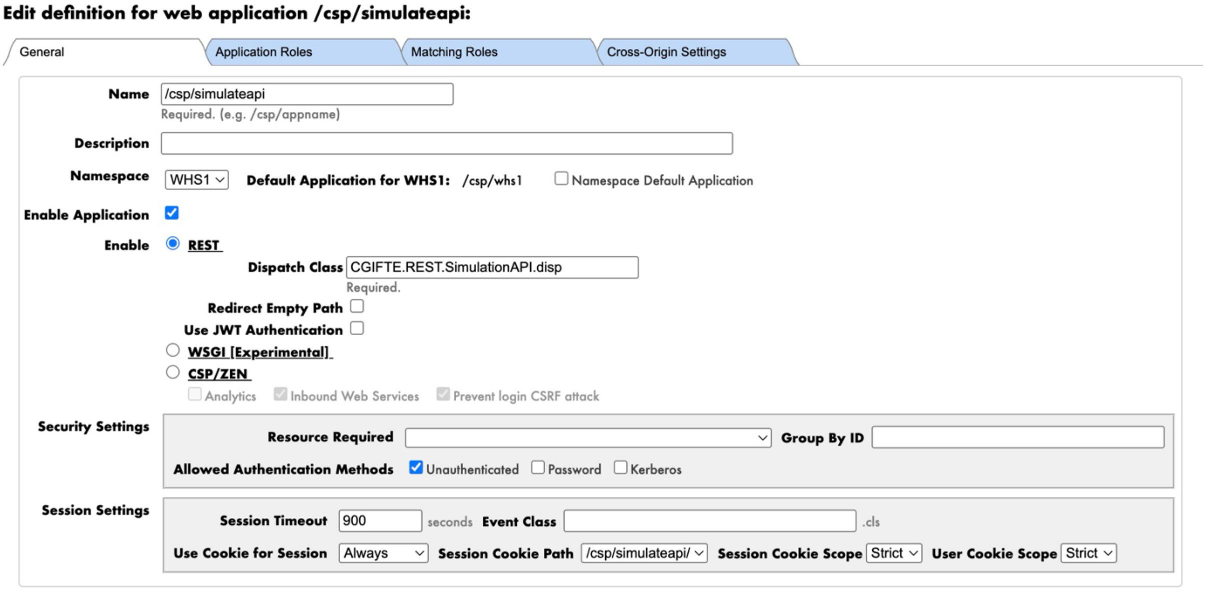


Production Name	Item Name	Host Class Name	Setting Name	Setting Value	Deployable
SCPKG.FoundationProduction	*	*	FilePath	/Users/jellemichiels/InterSystems/IRIS/ForecastWHS2/	No

- Import and compile the code in the src folder from here in the WHS namespaces (change the reference in the CGIFTE.Utils class

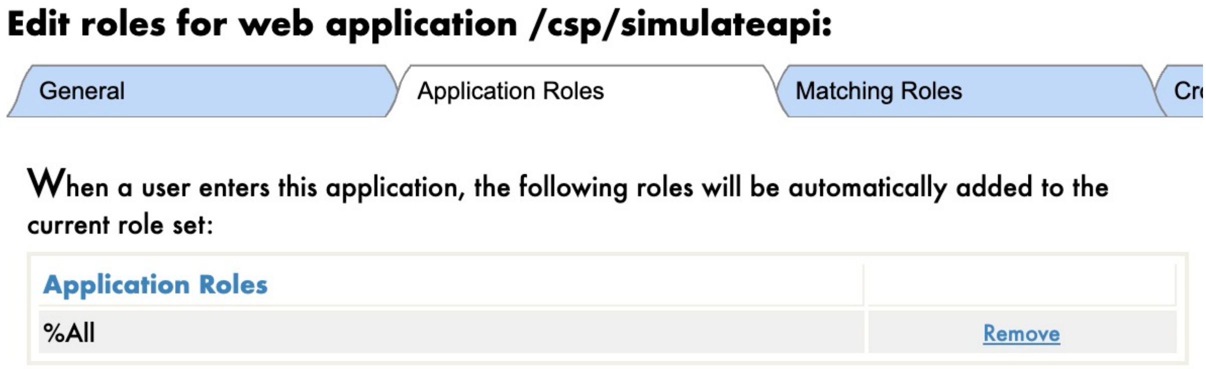
- Install the required python packages (numpy, datetime, pandas) as described here.

- Create web applications /csp/simulateapi with dispatch class CGIFTE.REST.SimulationAPI (in WHS1 namespace), as well ass /csp/warehouse1 and /csp/warehouse2 (in their resp. namespaces) with dispatch class CGIFTE.REST.EmployeeAPI



Edit definition for web application /csp/simulateapi:

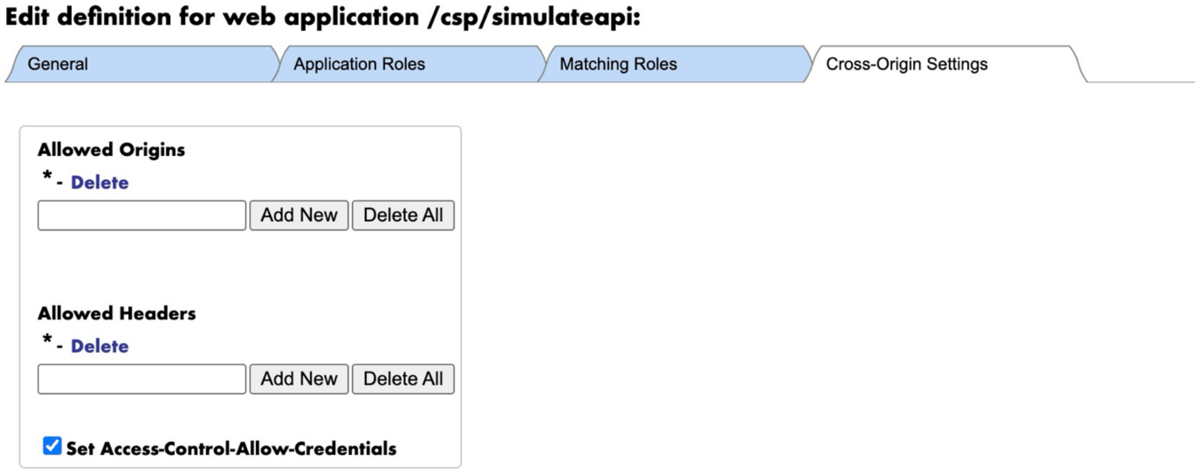
Name /csp/simulateapi
Required. (e.g. /csp/appname)
Description
Namespace WHS1 **Default Application for WHS1:** /csp/whs1 □ Namespace Default Application
Enable Application ☑
Enable ● **REST**
Dispatch Class CGIFTE.REST.SimulationAPI.disp
Required.
Redirect Empty Path □
Use JWT Authentication □
○ **WSGI [Experimental]**
○ **CSP/ZEN**
□ Analytics ☑ Inbound Web Services ☑ Prevent login CSRF attack
Security Settings
Resource Required **Group By ID**
Allowed Authentication Methods ☑ Unauthenticated □ Password □ Kerberos
Session Settings
Session Timeout 900 seconds **Event Class** .cls
Use Cookie for Session Always **Session Cookie Path** /csp/simulateapi/ **Session Cookie Scope** Strict **User Cookie Scope** Strict



Edit roles for web application /csp/simulateapi:

When a user enters this application, the following roles will be automatically added to the current role set:

Application Roles	
%All	Remove



Edit definition for web application /csp/simulateapi:

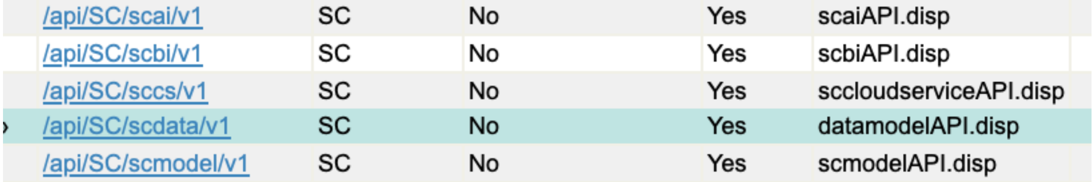
Allowed Origins
* - **Delete**
Add New Delete All

Allowed Headers
* - **Delete**
Add New Delete All

☑ **Set Access-Control-Allow-Credentials**

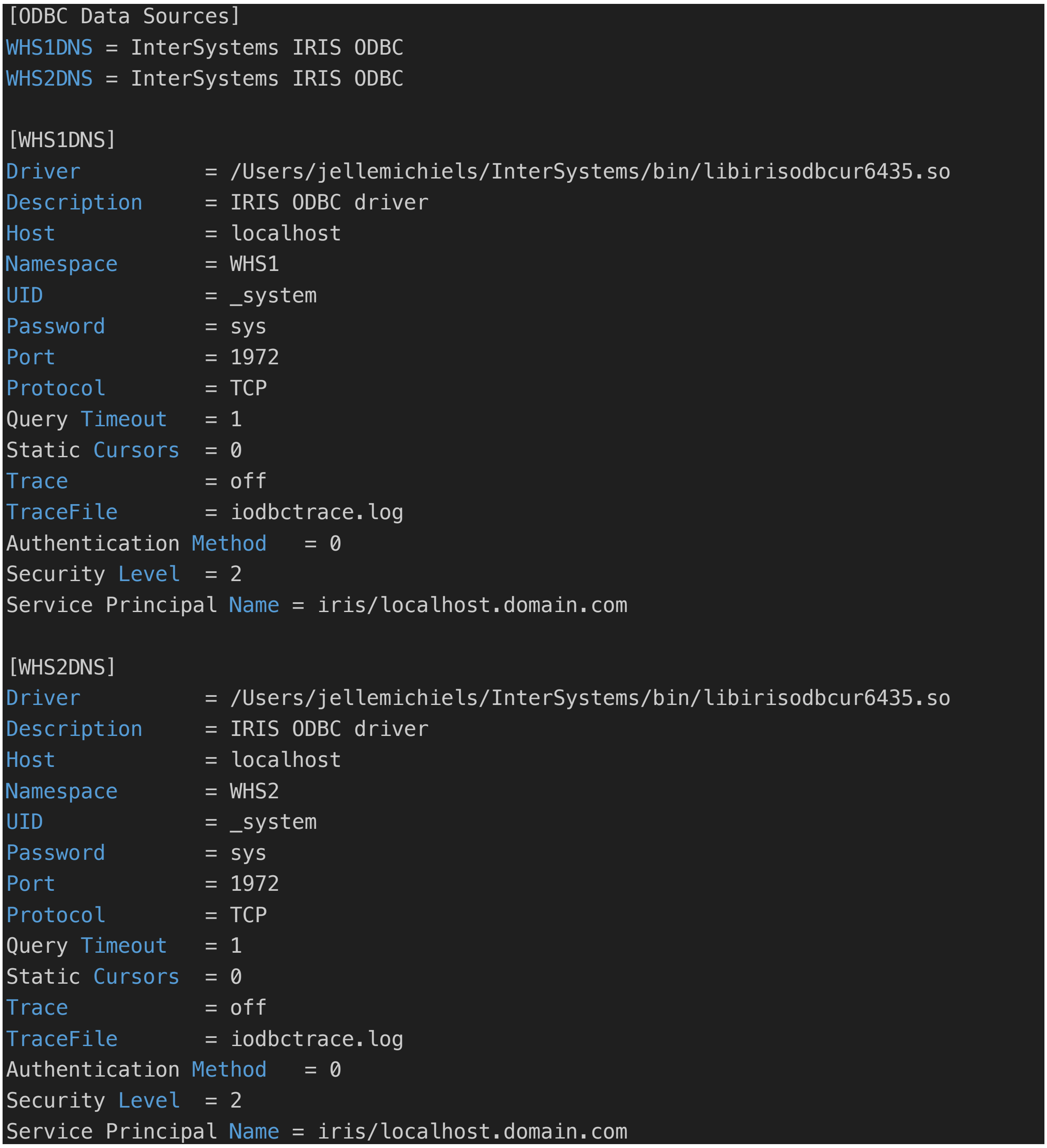
- Import and compile the code in the src folder from here in the SC namespace

- Make the SC apis unauthenticated and give them the appropriate application roles and enable cors as in the image above



/api/SC/scai/v1	SC	No	Yes	scaiAPI.disp
/api/SC/scbi/v1	SC	No	Yes	scbiAPI.disp
/api/SC/sccs/v1	SC	No	Yes	sccloudserviceAPI.disp
/api/SC/scdata/v1	SC	No	Yes	datamodelAPI.disp
/api/SC/scmodel/v1	SC	No	Yes	scmodelAPI.disp

- Add DNS for namespace WHS1 and WHS2 called WHS1DNS and WHS2DNS) E.g. on macos run ./ODBCinstall in IRIS installation directory and use following .ini file:



```
[ODBC Data Sources]
WHS1DNS = InterSystems IRIS ODBC
WHS2DNS = InterSystems IRIS ODBC

[WHS1DNS]
Driver          = /Users/jellemichiels/InterSystems/bin/libirisodbcur6435.so
Description     = IRIS ODBC driver
Host            = localhost
Namespace       = WHS1
UID             = _system
Password        = sys
Port            = 1972
Protocol        = TCP
Query Timeout   = 1
Static Cursors  = 0
Trace           = off
TraceFile       = iodbctrace.log
Authentication Method   = 0
Security Level  = 2
Service Principal Name = iris/localhost.domain.com

[WHS2DNS]
Driver          = /Users/jellemichiels/InterSystems/bin/libirisodbcur6435.so
Description     = IRIS ODBC driver
Host            = localhost
Namespace       = WHS2
UID             = _system
Password        = sys
Port            = 1972
Protocol        = TCP
Query Timeout   = 1
Static Cursors  = 0
Trace           = off
TraceFile       = iodbctrace.log
Authentication Method   = 0
Security Level  = 2
Service Principal Name = iris/localhost.domain.com
```

- In the SC namespace, start the SC.Core.BP.LaborProduction. Check if everything is green.

- In WHS1 and WHS2, start the SCPKG.FoundationProduction. Check if everything is green.

- Load the data in WHS1 and WHS2 using (replace the filename as needed):

```
LOAD DATA FROM FILE '/external/csv/Product referential.csv' INTO
CGIFTE_Persistent.Products USING {"from":{"file":{"header":true}}}
```

```
LOAD DATA FROM FILE
'/external/csv/Sales_receipts/Sales_receipts_STORE_WHS1.csv' INTO
CGIFTE_Persistent.SalesData USING {"from":{"file":{"header":true}}}
```

```
LOAD DATA FROM FILE
'/external/csv/Sales_receipts/Sales_receipts_WEB_WHS1.csv' INTO
CGIFTE_Persistent.SalesData USING {"from":{"file":{"header":true}}}
```

in WHS1 and

```
LOAD DATA FROM FILE '/external/csv/Product referential.csv' INTO
CGIFTE_Persistent.Products USING {"from":{"file":{"header":true}}}
```

```
LOAD DATA FROM FILE
'/external/csv/Sales_receipts/Sales_receipts_STORE_WHS1.csv' INTO
CGIFTE_Persistent.SalesData USING {"from":{"file":{"header":true}}}
```

```
LOAD DATA FROM FILE
'/external/csv/Sales_receipts/Sales_receipts_WEB_WHS1.csv' INTO
CGIFTE_Persistent.SalesData USING {"from":{"file":{"header":true}}}
```

in WHS2.

- In namespace SC, run:

```
SC>set sc = ##class(CGISC.Util).FullReset()
```

If you want to reset the demo, use this as well

- Load the LaborKPIs by executing the following code in the SC namespace:

```
set sc =  ##class(SC.Core.Util.LaborKPIUtil).loadLaborKPIs()
```

- Populate the employee table in WHS1 and WHS2. Execute the following in the sql shells in both namespaces:

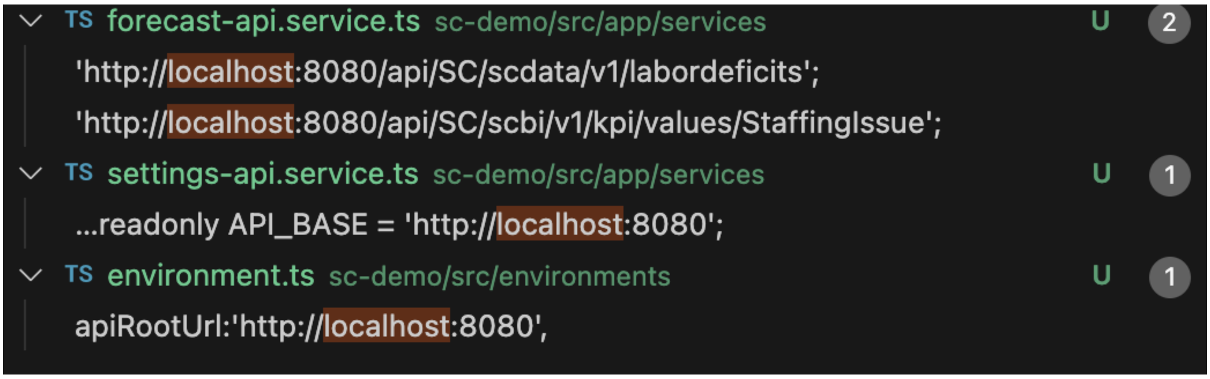
```
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP1','Storage,Pick,Prepare');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP5','Storage,Pick,Prepare');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP4','Storage,Pick,Prepare');
```

```
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP2','Storage,Pick,Prepare');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP10','Storage,Pick,Prepare');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP12','Storage,Pick,Prepare');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP17','Receive,Load');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP7','Storage,Pick,Prepare');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP11','Storage,Pick,Prepare');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP20','Receive,Load');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP14','Storage,Pick,Prepare');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP19','Receive,Load');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP21','Control');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP18','Receive,Load');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP16','Receive,Load');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP22','Control');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP24','Control');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP25','Control');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP23','Control');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP3','Storage,Pick,Prepare');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP13','Storage,Pick,Prepare');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP8','Storage,Pick,Prepare');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP6','Storage,Pick,Prepare');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP15','Storage,Pick,Prepare');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP9','Storage,Pick,Prepare');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP26','Storage,Pick,Prepare');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP27','Storage,Pick,Prepare');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP28','Storage,Pick,Prepare');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP29','Storage,Pick,Prepare');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP30','Storage,Pick,Prepare');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP31','Storage,Pick,Prepare');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP32','Storage,Pick,Prepare');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP33','Storage,Pick,Prepare');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP34','Storage,Pick,Prepare');
INSERT INTO CGIFTE_Persistent.Employee (Available, BusyUntil, EmployeeId, TaskType)
VALUES (0,NULL,'EMP35','Storage,Pick,Prepare');
```

- Populate the tables in the SC namespace in the same way using the following insert statements:

```
INSERT INTO SC_Data.Product (defaultUom, description, lastUpdatedTime, name,
recordCreatedTime, "type", uid) VALUES ('hours','Labor related to receiving and
loading tasks in the warehouse','2025-04-03 15:59:08.493','Forklift
operator','2025-04-03 15:59:08.493','LABOR','laborproduct-01');
INSERT INTO SC_Data.Product (defaultUom, description, lastUpdatedTime, name,
recordCreatedTime, "type", uid) VALUES ('hours','Labor related to quality control
task in the warehouse','2025-04-03 15:59:18.912','Quality control','2025-04-03
15:59:18.912','LABOR','laborproduct-02');
INSERT INTO SC_Data.Product (defaultUom, description, lastUpdatedTime, name,
recordCreatedTime, "type", uid) VALUES ('hours','Labor related to picking, packing
and storage task in the warehouse','2025-04-03 15:59:27.986','General warehouse
operations','2025-04-03 15:59:27.986','LABOR','laborproduct-03');
INSERT INTO SC_Data.Location (city, coordinates, country, lastUpdatedTime, name,
recordCreatedTime, "type", uid) VALUES ('Paris','48.85341, 2.3488','France','2025-
04-03 16:05:21.207','Pompéa warehouse','2025-04-03
16:05:21.207','WAREHOUSE','warehouse-01');
INSERT INTO SC_Data.Location (city, coordinates, country, lastUpdatedTime, name,
recordCreatedTime, "type", uid) VALUES ('Lille','50.6330, 3.0586','France','2025-
04-30 14:33:09.815','Pompéa warehouse Lille','2025-04-30
14:33:09.815','WAREHOUSE','warehouse-02');
```

- Download the angular UI from here, change the ip address in the following places so it points to the correct IRIS instance and run with ng serve (TO DO: fix so it uses the apiRootUrl everywhere)



```
forecast-api.service.ts sc-demo/src/app/services
    'http://localhost:8080/api/SC/scdata/v1/labordeficits';
    'http://localhost:8080/api/SC/scbi/v1/kpi/values/StaffingIssue';
settings-api.service.ts sc-demo/src/app/services
    ...readonly API_BASE = 'http://localhost:8080';
environment.ts sc-demo/src/environments
    apiRootUrl:'http://localhost:8080',
```