Integration with a Master Patient Index

The reference application can be configured to connect to a Master Patient Index(MPI), allowing patient information to be shared. This feature is still in active development and testing, and will be released soon.

OpenEMPI Integration Demo

The registration app is configured with default settings to allow it to connect to OpenEMPI. OpenEMPI is an Enterprise Master Patient Index developed by Sysnet International. It provides both a REST API and PIX/PDQ Server. Patient Identifier Cross-referencing(PIX) and Patient Demographics Query(PDQ) is done using the HL7 v2 Standard.

This system uses the OpenHIE framework, with OpenHIM acting as the Interoperability Layer and OpenEMPI acting as the Client Registry.

OpenHIM Setup

OpenHIM has to be installed to act as a reverse proxy for OpenHIE.

Step 1: Install Nodejs
Install the latest stable Nodejs LTS release(Nodejs v4 or later): https://nodejs.org/en/download/package-manager/

Step 2: Install Mongodb
Install Mongodb(compatible with v2.6 to v3.4): https://docs.mongodb.com/v3.4/administration/install-community/

Ensure that Mongodb is running after installation.

Step 3: Install openhim-core
Assuming npm is installed with Nodejs, run the npm command for installing openhim-core globally.

```
$ sudo npm install openhim-core -g
```

Before starting openhim-core, the api port has to be changed from its default of Port 8080, to prevent conflicts with OpenMRS and OpenEMPI. The config files can be found in node_modules/openhim-core/config.md and node_modules/openhim-core/default.json. Your node_modules folder is in the node install directory by default, and can be found based on this npm documentation.

In both files, change the port under api httpsPort from port 8080 to another port(e.g. Port 8085).

After the port has been configured, openhim-core can be run from anywhere using the following command.

```
$ openhim-core
```

More information regarding OpenHIM setup can be found at https://openhim.readthedocs.io/en/latest/getting-started.html

Step 4: Install OpenHIM Console

OpenHIM Console has to be run on a web server. Here are guides for installing apache2 and nginx.

Follow the OpenHIM guide on “Installing the OpenHIM Console”: https://openhim.readthedocs.io/en/latest/getting-started.html#installing-the-openhim-console

Releases can be found on the OpenHIM Console Github Page: https://github.com/jembi/openhim-console

OpenHIM Console has to be configured to connect to the openhim-core API Port. The config can be found in the directory openhim-console was extracted to, under config/default.json. The port has to match the API port configured in openhim-core in Step 3.

Ensure your web server is running before accessing the web console.

Step 5: Run and configure OpenHIM

Once installed, the OpenHIM console can be accessed at your web server's default port (Ubuntu apache2 and nginx: Port 80). Login with the default details:

Login: root@openhim.org
Password: openhim-password

You will be prompted to change the password upon successful login. If running for the first time with the default configuration, there may be an error logging in due to the use of a self signed certificate. Follow the instructions in the error or the instructions under “Getting Started with OpenHIM Console” on the Github Page if needed.

After logging in, go to Export/Import, and Import the following configuration: openhim-openempi-default.json

This will setup channels for PIX and PDQ, forwarding traffic from OpenMRS to the necessary ports for OpenEMPI. It also converts the HL7 over HTTP to HL7 over MLLP. Lastly, it creates a client with the credentials admin:admin, to allow OpenMRS to connect.

More information on OpenHIM Configuration can be found in the OpenHIM Documentation.

OpenEMPI Setup

Download OpenEMPI from the OpenEMPI Website: https://www.openempi.org/openempi-downloads/index.html

Install OpenEMPI using OpenEMPI Wiki Instructions: http://www.openempi.org/confluence/display/openempi30/Installation+Instructions

OpenEMPI Configuration

After OpenEMPI is installed, it has to be configured to work with OpenMRS.

Step 1: Change the OpenEMPI Admin Port

By default, OpenEMPI uses Port 8080 as the port for the administration webpage. This will cause a conflict as OpenMRS and OpenHIM also have Port 8080 as default ports. In order to prevent this conflict, the config file can be modified to change the port to another port(e.g. port 8090). OpenEMPI installs to a user's home directory, and the config file can be found in openempi/conf/server.xml.

```xml
<Connector port="8080" protocol="HTTP/1.1"
    connectionTimeout="200000"
    redirectPort="8443" />
```

OpenEMPI API Ports

The API ports that OpenEMPI uses for PIX/PDQ Communication can also be changed if needed. By default, PIX uses Port 3600 and PDQ uses Port 3700. The PDQ port can be changed in openempi/openempi-entity-x.x.xc/conf/PdSupplierConnections.xml(where x.x.x is the version number) and the PIX port can be changed in openempi/openempi-entity-x.x.xc/conf/PixConsumerConnections.xml and openempi/openempi-entity-x.x.xc/conf/PixManagerConnections.xml

Step 2: Run OpenEMPI and Edit Identifier Domains

Once the Admin Port has been changed, start openempi by running the appropriate script in the openempi/bin folder.

Windows: startup.bat

Unix based systems: startup.sh

The openempi admin can then be accessed through your web browser at localhost:8080/openempi-admin, where 8080 is replaced by the port you have configured in Step 1.
Login to OpenEMPI. Default:

Login: admin
Password: admin

Go to EditEdit Identifier Domains

Here we can add and change the identifier domains the OpenEMPI uses. The OpenEMPI GID is the default global identifier domain used in the MPI. The OpenMRS ID has to be added to OpenEMPI through “Add Identifier Domain”.

Here the OpenMRS ID can be added with the following values:

1. Name: OpenMRS ID
2. Description: OpenMRS Patient Identifier
3. Namespace Identifier: 2.25.71280592878078638113873461180761116318
4. Universal Identifier: 2.25.71280592878078638113873461180761116318
5. Universal Identifier Type: PI

Once this has been added, OpenEMPI has been configured for use with OpenMRS.
Further information on configuring OpenEMPI can be found in the OpenEMPI Documentation. Hapi TestPanel is an HL7v2 message editor, transmitter and receiver that can be used to test that OpenEMPI is configured correctly and working.

OpenMRS Registration Core Master Patient Index Global Properties

In order to use an MPI, the following Global Properties must be configured:

1. registrationcore.mpi.hl7implementation: specifies the version of HL7 used and the protocol used. Example: registrationcore.mpiHl7v2HttpSender
2. registrationcore.mpi.implementation: specifies which MPI implementation you are using. Since registrationcore v1.4 you can use OpenEMPI implementation.
3. registrationcore.mpi.url: specifies url to MPI server Property type: string. Example: 128.100.10.10:8080/openempi-admin
4. registrationcore.mpi.personIdentifierTypeUuid: specifies local patient identifier type id. Through this identifier imported MPI patient will be linked with patient on remote MPI server. You have to create new patient identifier type in Managing patient identifiers console.
5. registrationcore.mpi.pixEndpoint: url of the endpoint for pix communication. Example: localhost:5001/pix
6. registrationcore.mpi.pixPort: port used for pix communication. Default: 3600
7. registrationcore.mpi.pixIdentifierTypeUuidList: List of patient type identifiers included in pix messages
8. registrationcore.mpi.pdqPort: port used for pdq communication. Default: 3700
9. registrationcore.mpi.pdqIdentifierTypeUuidList: List of patient type identifiers included in pdq messages
10. registrationcore.mpi.receivingApplication: Application receiving pix/pdq messages. Default: Demo MPI
11. registrationcore.mpi.receivingFacility: Facility Receiving pix/pdq messages. Default: OpenEMPI Demo
12. registrationcore.mpi.sendingApplication: Application sending pix/pdq messages. Default: OpenMRS
13. registrationcore.mpi.sendingFacility: Facility Sending pix/pdq messages. Default: Demo

MPI Identifier TypeMap

In order to allow interoperability between various MPIs, a TypeMap global property is created such that the Uuid of each Patient Identifier in OpenMRS can be converted to the ID format used by the MPI. OpenEMPI uses an Object Identifier(OID) based system with HL7 OIDs. The default global properties thus do a mapping from an OpenMRS Uuid to the OpenEMPI ID. The map has three sections:

1. OpenMRS Identifier Type Uuid (Example: 05a29f94-c0ed-11e2-94be-8c13b969e334)
2. MPI Identifier Type ID (Example: 2.25.7128059287807638113873461180761116318)
3. Identifier Type (Example: PI)

By default, there are two TypeMaps configured for use with OpenEMPI:

1. registrationcore.local_mpi_identifierTypeMap.OpenMRS ID: 05a29f94-c0ed-11e2-94be-8c13b969e334:
   2.25.7128059287807638113873461180761116318:PI
2. registrationcore.local_mpi_identifierTypeMap.OpenEMPI ID: a5d38e09-efcb-4d91-a526-50ce1ba5011a:2.16.840.1.113883.4.357:hl7

OpenEMPI Specific Global Properties

1. registrationcore.openempi.enableProbabilisticMatching: This determines whether OpenEMPI uses probabilistic matching.
2. registrationcore.openempi.globalIdentifierDomainId

OpenMRS Runtime Properties

In order to authenticate with the MPI, you have to add two properties to the .OpenMRS/openmrs-runtime.properties file. These have to be the credentials configured within the MPI you are using.

1. registrationcore.mpi.username: username for authentication on MPI server
2. registrationcore.mpi.password: password for authentication on MPI server

For example:

registrationcore.mpi.username=admin
registrationcore.mpi.password=admin