

Open HMIS API

Description

API Methods Currently Implemented

[client/add](#)
[client/get](#)
[client/update](#)
[client/delete](#)
[client/outreach/add](#)
[client/search \(using HUD unduplication guidelines\)](#)
[programInfoByID](#)
[programList](#)
[programNameByID](#)
[search/firstname](#)
[search/lastname](#)
[existence/search](#)
[shelterList](#)
[bed_units/occupied](#)
[bed_units/occupied/set](#)
[bed_units/occupied/inc](#)
[bed_units/occupied/dec](#)
[bed_units/available](#)

API Methods to be Implemented Soon

[HMIS/AIRS basic methods](#)
[HUD HMIS report methods](#)
[user/add](#)
[user/get](#)
[user/delete](#)
[user/update](#)
[agency/add](#)
[agency/get](#)
[agency/delete](#)
[agency/update](#)
[program/add](#)
[program/get](#)
[program/delete](#)
[program/update](#)
[code repository](#)
[search demo](#)

Description

The purpose of the Open HMIS Application Programming Interface (API) is to allow third-party apps to interact with the Open HMIS system. The basic model of the Open HMIS is to serve as a data repository with controlled access to outside applications (like mobile phone apps). The third-party apps communicate with the Open HMIS through predefined interactions, and these

definitions are the API. That way, a community can use a variety of purchased HMIS apps with their existing Open HMIS system, and not need to procure for new software each time they want additional application functionality. The current API interactions (aka "API web methods") are listed below.

HMIS users (for example, a shelter employee) at a Continuum of Care (HUD HMIS region) will possess a login, which grants them specific rights within the local Open HMIS system, regardless of which third-party app they are using. The third-party app will use a technology that does not allow it to "see" the HMIS shelter employee's password. This way, the Open HMIS allows the Continuum of Care to have control over its data, as well as who can access what parts of the data set.

The Continuum of Care can select which third-party applications (say, an Android or iOS tablet app) can access their Open HMIS server data. Third-party apps which implement the Open HMIS API, and optionally display the "Powered by OpenHMIS" logo will be able to communicate with any compliant Open HMIS server "out of the box", with no additional third-party app customizations and cost.

API Methods Currently Implemented

(at open-ciss.appspot.com)

client/add

Adds a new client to the system. It doesn't do any matching/unduplication yet, but that's the intended eventual behavior. The API shouldn't need to change when that functionality is working.

method input: application/xml. HTTP POST hmis:Person type instance from the schema at:

http://www.hmis.info/schema/3_0/HUD_HMIS.xsd

method return: text/plain. The newly sequentially generated integer client id in the OpenHMIS. If the method is unsuccessful, a "-1" is returned.

documentation for hmis:Person:

http://www.hmis.info/schema/3_0/docs/HUD_HMIS.xsd0.html

Currently, it only stores the legal first name, legal last name, legal middle name, gender, ethnicity, and date of birth data elements. It ignores everything else in the Person type for now, but it does check for XML validity.

client/get

Retrieves a client record by ID. This example url will retrieve the client with an id of '4'
<http://open-ciss.appspot.com/client/get?id=4>

method input: an HTTP GET url integer ID parameter.

method return: application/xml. An hmis:Person XML instance. It lacks date_effective timestamps to make it valid HMIS XML, since the OpenHMIS database doesn't store those timestamps yet.

If a client ID that does not exist or is not accessible is requested, the method responds with
with a "-1" return value.

client/update

Updates an existing client in the system. The url parameter targets the to-be-updated client recorded. The POSTed XML provides the updated client record information.

Example: <http://open-ciss.appspot.com/client/update?id=5> (with HTTP POST XML attached to the request)

method input: application/xml. HTTP POST hmis:Person type instance from the schema at:

http://www.hmis.info/schema/3_0/HUD_HMIS.xsd

method return: text/plain. Returns the updated client id in the OpenHMIS. If the method is unsuccessful, a "-1" is returned.

documentation for hmis:Person:

http://www.hmis.info/schema/3_0/docs/HUD_HMIS.xsd0.html

Currently, it only updates the legal first name, legal last name, legal middle name, gender, ethnicity, and date of birth data elements. It ignores everything else in the Person type for now, but it does check for XML validity. The XML's person id will eventually be stored in a log, for synchronization purposes.

client/delete

This web method actually simply inactivates the record, if it wasn't already inactive.

Example:

<http://open-ciss.appspot.com/client/delete?id=5>

method input: HTTP GET url integer "id" parameter.

method return: Returns "200" if successful and "-1" if it failed to inactivate the client (if no matching record found, or insufficient permissions, etc.).

client/outreach/add

Adds a single client outreach event [*Insert link to official definition of an outreach*] that is associated with a preexisting client (created initially with client/add). It stores the latitude, longitude, contact date and engagement date for the outreach event.

Example:

Method input: HTTP POST JSON Object representing a single client outreach event. Example identifying elements/values are:

```
{"Client_ID":"2","Latitude":"-1.4432","Longitude":"104.81123","Contact_Date":"2012-11-03T16:18:05GMT","Engagement_Date":"2012-08-00T11:01:59GMT"};
```

Any/all of the identifiers can be absent (better than using blank strings). Latitude and Longitude are stored to 5 decimal places, but fewer may be submitted. Greater than 5 places will be rounded.

Method return: The outreach method returns the unique generated id of the outreach event, for later reference. If the method fails, "-1" is returned.

client/search (using HUD unduplication guidelines)

Searches the Open HMIS for a matching client keys, based on a set of given personal identifiers.

Method input: HTTP POST JSON Object representing a single client to search for. Example identifying elements/values are:

```
{"First_Name":"Hester","First_Name_Alias":"Hes","Last_Name":"","Prynn","Last_Name_Alias":"Oma","Middle_Initial":"B","Date_Of_Birth":"1960-09-04","Gender":"2","SSN":"111-11-1111","SSN_Quality":"1"}
```

Any/all of the identifiers can be absent (better than using blank strings), but absent elements may reduce the likelihood, or number of, matches. This is because the HUD algorithm uses a points system.

Method return: JSON Array of integer client keys that match, like: [4,121,160]. Empty results just return [].

This search fully implements the HUD recommended methodology for matching existing clients as described at:

<http://www.hudhre.info/documents/TechnicalGuidelinesForUnduplicatingAndDeIdentifyingClientRecords.pdf>

There is one exception, wherein multiple last and first name aliases are not checked by the open-ciss implementation, since for now the database only stores one last name alias and one first name alias (a nickname).

programInfoByID

Retrieves a program's basic information, using its ID. This example url will retrieve program information with an ID of '5': <http://open-ciss.appspot.com/programInfoByID?id=5>

method input: HTTP GET url integer "id" parameter.

method return: application/json object. Items are: ProgramKey, ProgramName, AgencyName, ProgramType, SiteGeocode, TargetPopAName, UpdateTimeStamp, UnitsTotal, UnitsAvailable, UnitsOccupied, ContactName, ContactPhone, ProgramAddress, ProgramCity, ProgramZip, ProgramAddressFull.

programList

Retrieves a listing of all IDs for all accessible programs. These IDs can then be individually queried with programInfoByID.

method input: HTTP GET

method return: application/json. A JSON array containing program IDs.

programNameByID

Retrieves the name of a program using its ID. This example url will retrieve the program name with an ID of '5': <http://open-ciss.appspot.com/programNameByID?id=5>

method input: HTTP GET url integer "id" parameter.

method return: application/json. JSON object containing the program's name.

search/firstname

Retrieves a list of first names that match the initial character submitted (case sensitive at the moment). Used for predictive text search boxes. Example:

<http://open-ciss.appspot.com/search/firstname/Ben>

(might return something like: {"firstName":["Benedict","Benjamin"]})

method input: HTTP GET search/firstname/{prefix}.

method return: application/json. JSON object containing an array of first names that match.

search/lastname

Retrieves a list of last names that match the initial character submitted (case sensitive at the moment). Used for predictive text search boxes. Example:

<http://open-ciss.appspot.com/search/lastname/Sag>

(might return something like: {"lastName":["Sage","Sageser","Sagey","Saggett"]})

method input: HTTP GET search/lastname/{prefix}.

method return: application/json. JSON object containing an array of last names that match.

existence/search

Tests if a given full first and full last name (not just the initial characters) combination exists within the Open HMIS. Example:

<http://open-ciss.appspot.com/existence/search?firstname=Benedict&lastname=Saggett>

(might return: {"Exists":true})

method input: HTTP GET url "firstname" parameter, "lastname" parameter.

method return: application/json. JSON object containing a boolean value.

shelterList

Retrieves a list of shelter IDs, which are programs IDs filtered for the shelter type. The IDs in the list can then be individually obtained using programInfoByID.

method input: HTTP GET

method return: application/json. A JSON array containing shelter IDs.

bed_units/occupied

Retrieves the number of occupied shelter units for a given program ID. Example: http://open-ciss.appspot.com/bed_units/occupied?id=5

method input: HTTP GET url integer "id" parameter.

method return: text/plain. An integer representing the number of occupied shelter beds at the program.

bed_units/occupied/set

Assigns the number of occupied shelter units for a given program ID. Example: http://open-ciss.appspot.com/bed_units/occupied/set?id=5&val=31

method input: HTTP GET url integer "id" parameter, integer "val" parameter.

method return: text/plain. An integer representing the echoed number of occupied shelter beds at the program.

bed_units/occupied/inc

Increments the number of occupied shelter units for a given program ID by one unit. Example: http://open-ciss.appspot.com/bed_units/occupied/inc?id=5

method input: HTTP GET url integer "id" parameter

method return: text/plain. An integer representing the now incremented number of occupied shelter beds at the program.

bed_units/occupied/dec

Decrements the number of occupied shelter units for a given program ID by one unit. Example: http://open-ciss.appspot.com/bed_units/occupied/dec?id=5

method input: HTTP GET url integer “id” parameter

method return: text/plain. An integer representing the now decremented number of occupied shelter beds at the program.

bed_units/available

Retrieves the number of available shelter units for a given program ID. Example:
http://open-ciss.appspot.com/bed_units/available?id=5

method input: HTTP GET url integer “id” parameter

method return: text/plain. An integer representing the number of available shelter beds at the program.

bed_units/total

Retrieves the total number of shelter units for a given program ID. Example:
http://open-ciss.appspot.com/bed_units/total?id=5

method input: HTTP GET url integer “id” parameter

method return: text/plain. An integer representing the number of total shelter beds at the program.

bed_units/total/set

Assigns the total number of shelter units for a given program ID. Example:
http://open-ciss.appspot.com/bed_units/total/set?id=5&val=40

method input: HTTP GET url integer “id” parameter, integer “val” parameter.

method return: text/plain. An integer representing the echoed number of total shelter beds now at the program.

bed_units/total/inc

Increments the total number of shelter units for a given program ID by one unit.
Example:
http://open-ciss.appspot.com/bed_units/total/inc?id=5

method input: HTTP GET url integer “id” parameter

method return: text/plain. An integer representing the now incremented total number of shelter beds at the program.

bed_units/total/dec

Decrements the total number of shelter units for a given program ID by one unit.
Example:
http://open-ciss.appspot.com/bed_units/total/dec?id=5

method input: HTTP GET url integer “id” parameter

method return: text/plain. An integer representing the now decremented total number of shelter beds at the program.

API Methods to be Implemented Soon

HMIS/AIRS basic methods

- We plan to add [these](#) basic web methods based on the Homeless Management Information System (HMIS) and (Alliance of Information and Referral Services) AIRS XML

HUD HMIS report methods

- Additional API calls could be created to retrieve each of the each of the various Dept. of HUD HMIS reports, such as the Annual Performance Report and Annual Performance Report, etc.

user/add

user/get

user/delete

user/update

agency/add

agency/get

agency/delete

agency/update

program/add

program/get

program/delete

program/update

code repository

<http://code.google.com/p/open-community-information-sharing-system/source>

search demo

<http://open-ciss.appspot.com/SearchDemo.html>