FRED API

# Document Information

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Author** | **Version** | **Date** | **Rationale** |
| Justin Fyfe | 0.1 | 19-NOV-2012 | Initial Version |
|  |  |  |  |
|  |  |  |  |

## Related Documents

This document relies on or references the following documents

|  |  |  |
| --- | --- | --- |
| **Name** | **Url** | **Relation** |
| Collaborative Health Platform | <http://tinyurl.com/c4zhyru> | High level system description / role description. |
|  |  |  |
|  |  |  |
|  |  |  |

# Introduction

Will introduce the project at a high level.

## Overview of the FRED API

High level overview of the FRED API

## Definitions

1. “CHP” is used to describe the actors, transactions and roles described in the Collaborative Health Platform document released on the HUB in December 2011.
2. “Client” describes a consumer of health care services and it most often interchangeable with “patient”
3. “Facility” describes a logical place or point of care where health services are provided to clients.
4. “System” describes the overall health infrastructure, its components, interactions and actors. This term is often used to describe the overall health infrastructure in which the facility registry will operate
5. “Actor” is used to describe a series of responsibilities that a consumer or provider application must provide in order to participate in a clinical act. This term is interchangeable with “role” used in the CHP document.
6. “API” is used to describe an application programming interface which allows FRED Consumers to consume the services offered by the FRED Provider. It is the concrete data models and operations executed, at runtime, against application acting in the FRED Provider role.
7. “HIX” is a term used in the CHP document to describe a centralized health information exchange, and is used in this document in an informative manner.

## Purpose

Describe the purpose or goal of the document

## Scope

Identifies the scope of the document

## Standards & Real-world Architectures

Will relate the data collected in this document to our audience and the architectures currently deployed in Rwanda, etc.

## Collaborative Health Platform

This document is specifies both abstract data elements and concrete API definitions required for applications to maintain facility registry data. This functionality is a very close map to the abstract Facility Registry Service Supplier (ROL05) and Facility Registry Service Consumer (ROL06) roles defined in the CHP document.

This specification does not assume a full CHP infrastructure has been put in place, and has been specified to operate as a standalone service, or as part of a larger HIX. The specification will make reference to the CHP roles and interactions found between l. 535 (p. 24) and l. 635 (p. 27) of the CHP document.

# Reading this Document

This document will use several types of diagrams to illustrate how the actors within the system interact with one another. Where possible, this verbiage is aligned with the CHP roles/transactions, and many of the diagrams are common with the CHP framework document.

## Communications Diagrams

Communications diagrams are used to illustrate (at a high level) how consumer and provider roles interact with one another. Figure 1 illustrates a sample communication diagram whereby the FRED Provider (identified using CHP ROL05) interacts with the FRED consumer (identified using CHP ROL06). The figure illustrates that a consumer must send a query message (CHP FR03) to the service provider and must be capable of interpreting query results (CHP FR04).



Figure 1 - Sample communications diagram

It is important to note that these communications diagrams have no implied order; they are simply used to show what needs to be sent/received by the actors.

## Data Model Diagrams

Although the FRED RESTful API allows for the use of JSON, data model diagrams are illustrated as XML schema visualizations. Attributes (the JSON representations of which are listed in Appendix B) are illustrated in the attributes box, sequences are illustrated using the sequence icon.

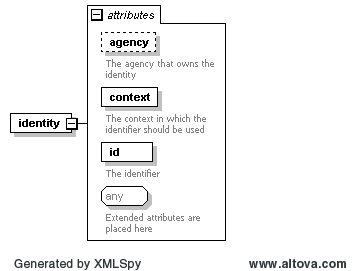


Figure 2 - Sample data model diagram

# FRED Transactions

All FRED transactions are RESTful operations against a collection of facility resources. All operations are performed against the facility resource which is further defined in Appendix A. Any operation that deviates or restricts the facility resource will declare these modifications in the message semantics section.

## Record and Maintain Facility Data

The record and maintain facility data transaction (FRED transaction 1) describes the processes under which a facility data source notifies the facility repository when new facilities are registered, or updated. This transaction fulfills FR03 (Register Facility) and FR05 (Update Facility) interactions identified in the CHP framework.

### Scope

The actors that are involved in this transaction are illustrated in Figure 3.



Figure 3 – Record and maintain facility data actors

|  |  |
| --- | --- |
| **Actor** | **Purpose** |
| Facility Registry | Services facility data for an organization or jurisdiction. Responsible for the maintenance of that data over a long period of time and making that data available to other applications. |
| Facility Data Source | An application which is capable of generating or updating facility data. This application may provide a user interface for editing of facility details or may simply be an export function of a legacy system. |

### Use Case(s)

Place any use cases that support this transaction here, or merely reference them and provide them in an index.

### Open Data Formats / Standards Referenced

This transaction makes use of the following standards:

* HTTP 1.1
* W3C WGS84 Basic Geographic Latitude & Longitude Vocabulary

### Interactions

Figure 4 illustrates the sequence of messaging between the Facility Data Source (FRED\_SRC) actor and Facility Registry (FRED\_REG).



Figure 4 - Record and Maintain facility interactions

### Triggering Events

The facility data source will execute one of register facility or update facility events against the facility registry.

#### Register Facility

When a new facility has been added to the facility data source or when the facility data source deems it appropriate to share a facility it has not previously shared with the facility registry, it will notify the facility registry of this addition using the record facility transaction.

Changes to an existing record on the facility data source will trigger a revise facility action.

##### Message Semantics

**HTTP Method:** POST  
**Resource:** {base}/facilities  
**Content Type:** text/xml or text/json

Facility data sources SHALL submit either an XML or JSON encoded facility resource as the payload of the HTTP message. All data sources SHALL send an appropriate content-type header which describes the type of data conveyed in the payload.

###### “url” / “id” Element Restrictions

The “url” and “id” elements of the facility resource SHALL NOT carry a value on the register facility request as this value is to be populated by the facility registry.

##### Examples

Figure 5 illustrates a sample request to create a facility named “Good Health Hospital” which was created sometime in November 2012.

POST [http://example.com/api/fred/1.1/facilities HTTP/1.1](http://example.com/api/fred/1.1/facilities%20HTTP/1.1)

Content-Type: text/xml

Host: [example.com](http://www.example.com)

Content-Length: 692

<fac:facility xmlns:fac="http:www.openfacility.org/v1.0" xmlns:geo="http://www.w3.org/2003/01/geo/wgs84\_pos#">

<fac:name>Good Health Hospital</fac:name>

<fac:identity agency="MOH" context="HR" id="20394"/>

<fac:active>true</fac:active>

<fac:creationTime precision="M">2012-11-01</fac:creationTime>

<geo:lat>1.69172</geo:lat>

<geo:long>29.52505</geo:long>

<fac:link name="providers" url="http://providers.moh.gov.za/providers"/>

</fac:facility>

Figure 5 - Sample register facility operation

##### Expected Behaviors

When the facility registry receives a request to register a facility, the facility registry will first check its current datastore to determine if an existing facility already exists. The matching algorithm used by the facility registry is not specified in this document and should be whatever algorithm is deemed appropriate for the deployment environment of the facility registry.

Depending on the outcome of the match one of three actions are to be taken by the facility registry:

1. If the facility registry finds exactly one matching facility, it will merge data stored in the registry with the data provided by the facility data source. This operation should only be performed if one (and only one) guaranteed match is found.
2. If the facility registry finds multiple matching facilities on file, it will create a new facility entry in its datastore. It is recommended that the facility registry somehow indicate to administrators of the system that potential matches exist so that they can be merged at a later time.
3. If the facility registry finds no matching facilities on file, it will create a new facility entry.

After the facility registry has completed its write operation, it SHALL make the facility data available to consumers and SHALL respond with an HTTP 200 code with the url to the newly created facility in the format “{base}/facilities/{id}”. The facility registry SHALL NOT make facility records available prior to ensuring all facility data has been committed to its datastore (no partial data shall be disclosed or available for update).

#### Revise Facility

When a facility registration record changes in the facility data source’s datastore, it will notify the facility registry of this change using the revise facility operation.

##### Message Semantics

**HTTP Method:** PUT  
**Resource:** {base}/facilities/{id}  
**Content Type:** text/xml or text/json

Facility data sources SHALL submit either an XML or JSON encoded facility resource as the payload of the HTTP message. All data sources SHALL send an appropriate content-type header which describes the type of data conveyed in the payload.

Revision operations SHALL be executed against the fully qualified url for the resource which is being updated.

##### Examples

Todo

##### Expected Behavior

When the facility registry receives a request to revise a facility, the facility registry SHALL validate that the requested facility exists. If the requested target of revision (the facility to be updated) does not exist the facility registry SHALL reply with an HTTP 404 error.

If the facility resource exists, the facility registry SHALL update its datastore with the new information and SHALL respond with an HTTP 200 response code.

#### Obsolete Facility

When a facility record is no longer relevant, or was created in error, the facility data source will notify the facility registry of this change using the obsolete facility.

##### Message Semantics

**HTTP Method:** DELETE  
**Resource:** {base}/facilities/{id}

##### Examples

Todo

##### Expected Behavior

When the facility registry receives a request to obsolete a facility, the facility registry SHALL validate that the facility exists. If the requested target of obsoletion does not exist, the facility registry SHALL respond with an HTTP 404 error.

If the facility resource exists, the facility registry SHALL mark the facility resource as “obsolete”. Obsolete records SHALL NOT be accessible after obsoletion (i.e. SHALL NOT be returned in queries and SHALL NOT be available via GET). The process by which the facility registry marks the facility as obsolete is not specified in this document, and is left to implementers to determine the most appropriate obsoletion method.

Once the record is obsolete, the facility registry SHALL return an HTTP 200 response with the URL of the deleted facility.

### Extended Attributes / Elements

The facility registry SHALL be capable of receiving facility resources which have extended attributes. There is no requirement that the facility registry be able to meaningfully process and/or store additional elements outside the scope of the core facility resource defined in Appendix A.

The facility registry SHALL raise an HTTP 422 error if any extended element carries the “mustUnderstand” attribute if the attribute value is set to “true”. The error message retuned must read: “Don’t understand mustUnderstand element X” where X is the element which carries the mustUnderstand attribute.

## Query Facility Data

The query facility data transaction (FRED transaction 2) describes the process whereby a facility data consumer queries and consumes facility data from a facility registry.

### Scope

The actors that are involved in this transaction are illustrated in Figure 6.



Figure 6 – Record and maintain facility data actors

|  |  |
| --- | --- |
| **Actor** | **Purpose** |
| Facility Registry | Services facility data for an organization or jurisdiction. Responsible for the executing of queries, and fetching of facility detail data. |
| Facility Data Consumer | An application which is capable of constructing and consuming queries against the facility registry. |

### Use Case(s)

Place any use cases that support this transaction here, or merely reference them and provide them in an index.

### Open Data Formats / Standards Referenced

This transaction makes use of the following standards:

* HTTP 1.1
* W3C WGS84 Basic Geographic Latitude & Longitude Vocabulary
* RSS 2.0 Specification

### Interactions

Figure 7 illustrates the sequence of messaging between the Facility Data Consumer (FRED\_CONSUMER) actor and Facility Registry(FRED\_REG).



Figure 7 - Record and Maintain facility interactions

### Triggering Events

The facility registry consumer will execute one of query facility or get facility details to list and/or get facility details respectively.

#### Query Facilities

Todo

##### Message Semantics

**HTTP Method:** GET  
**Resource:** {base}/facilities[.xml|.json]  
**Content Type:** text/xml or text/json

Facility data consumers SHALL execute a GET against the facilities collection to initiate a query. Consumers SHALL pass query parameters to the facility registry via query parameters in the format *propertyName=filterValue*.

Query parameters SHALL be passed as one value per parameter, core filter properties are listed in Table 1. Implementers MAY choose to extend the available query parameters made available to consumers. When this is done, the extended query parameters SHALL follow the same format as the core query parameters.

Table 1 - Query facility filter parameters

|  |  |  |
| --- | --- | --- |
| **Query Parameter** | **Filter Format** | **Example** |
| name | String | ?name=GHHS |
| url | URL | ?url=http://example.com |
| active | Boolean | ?active=true |
| approvalDate | Date | ?approvalDate=2012 |
| lat | Decimal | ?lat=1.23 |
| long | Decimal | ?long=1.32 |
| identity | id@agency | ?1234@moh |

###### “url” Element Restrictions

The “url” element and “identities/guid” element of the facility resource SHALL NOT carry a value on the register facility request as this value is to be populated by the facility registry.

The facility registry SHALL generate a globally unique identifier for all facilities which it registers, and SHALL make this identifier available via the identities/guid element. The type of identifier generated is not specified here however it SHOULD be representable using a URI syntax and SHALL be globally unique. Some recommended identifier formats are:

* UUIDs in the format : urn:uuid:E8A1650D-7FF9-4d3e-B390-D6FDD4CFB2E0
* URLs which point to the resource: http://example.com/api/fred/1.1/ facilities/10293
* ISO OIDs in the format: urn:oid:1.3.6.1.5.6.7.8343

###### “x” Element Restrictions

Todo

##### Examples

Examples

POST [http://example.com/api/fred/1.1/facilities HTTP/1.1](http://example.com/api/fred/1.1/facilities%20HTTP/1.1)

Content-Type: text/xml

Host: [example.com](http://www.example.com)

Content-Length: 692

<fac:facility xmlns:fac="http:www.openfacility.org/v1.0" xmlns:geo="http://www.w3.org/2003/01/geo/wgs84\_pos#">

<!-- TODO: Schema says this is NCNAME which means I have to use this naming convention

perhaps we should change this? -->

<fac:name>GHHS</fac:name>

<fac:identities>

<fac:identity agency="MOH" context="HR" id="20394"/>

</fac:identities>

<fac:active>true</fac:active>

<fac:approvalDate precision="M">2012-11-01</fac:approvalDate>

<geo:lat>1.69172</geo:lat>

<geo:long>29.52505</geo:long>

<fac:link name="providers" url="http://providers.moh.gov.za/providers"/>

</fac:facility>

Figure 8 - Sample register facility operation

### Expected Behavior

Todo

#### Facility Registry

Todo

#### Facility Data Consumer

Todo

# Facility resource implementation details

This appendix describes additional details related to the

Include a schema reference here

# JSON representation of facility resources

All FRED resources may be represented as JSON objects. This specification

# Facility Registry Behaviors

## Reporting of Errors

If the facility registry encountered an internal error (datastore is not available, full, etc…) the facility registry SHALL respond with an HTTP 500 error, signaling that an internal registry problem occurred. Implementers may choose to use extended HTTP 500 error codes to convey more detailed error messages. These extended 500 error codes are outside the scope of this specification however must adhere to HTTP status code conventions.