API Functionality Document

API-2015032901-03

Final Copy

Project: OptiRec version 1.0

Authors:
Andrew Madison

Date: 2015-10-07

Revision Record

Level	Date	Author	Change
01	2015-03-29	Andrew Madison	Initial document creation
02	2015-04-27	Andrew Madison	Updated methods returns
03	2015-10-07	Andrew Madison	Prepared document for publishing

Table of Contents

Introduction

Purpose

The Discharge Reports Application Programming Interface (API) is a load balanced and fault tolerant solution for client integration with the datastore.

The API allows clients to make ZMQ (ZeroMQ http://zeromq.org/) based requests to retrieve, upload and modify data based on pre-configured Electronic Medical Records (EMR) platforms. Data storage (datastore) is provided by Cassandra (http://cassandra.apache.org/) and the communication protocol used is JSON (http://json.org/)

Scope

To define available API functions and establish the base client to API interaction methodology.

Definitions

- **EMR:** Electronic Medical Record, commonly referred to the application which manages and stores the patient data (Epic, Cerner, McKesson, NextGen, etc.)
- **Meaningful use:** Improved patient care and access to medical records as defined in Stages 1 and Stages 2 by Centers for Medicare and Medicaid Services (http://www.cms.gov/Regulations-and-Guidance/Legislation/EHRIncentivePrograms/Meaningful_Use.html)
- **Light Weight:** Application developed for high performance and limited resource utilization (CPU and Memory). Such applications are designed for fast response times and enhanced user experience. To achieve this goal all functions and component variables must be followed by a garbage collection service where memory is cleared upon completion of the request.
- **MRN:** Medical record number, also known as unique patient identifier

References

- Apache Cassandra: http://cassandra.apache.org/
- Java: https://www.oracle.com/java/
- ZeroMQ: http://zeromq.org/
- Twitter Bootstrap: http://getbootstrap.com/
- JQuery Auto-Complete: http://jqueryui.com/autocomplete/
- UUID: http://en.wikipedia.org/wiki/Universally_unique_identifier

API Functionality

The Basics

API interaction is performed using JSON requests.
Requests are served with JSON response.

Requests

A JSON request must include 3 major components:

1. AuthHeader
2. method
3. request

AuthHeader is the authentication message object sent on every request. It contains 4 variables:

1. AuthKey – UUID value for your account (key value for the record in the auth_accounts column family). This parameter is not mandatory yet should be used for auditing purposes.
2. Auth_UName – Authentication username
3. Auth_UPass – Authentication clear text password
4. Auth_UToken – UUID value for authentication token. Initial request will have this value blank. All subsequent requests must include the authentication token received in the first response.

Sample AuthHeader message:

```
"AuthHeader": {
    "AuthKey":     "00000000-aaaa-2222-bbbb-34567890cdef",
    "Auth_UName":  "apiuser",
    "Auth_UPass":  "iyGHRciDe.1GS",
    "Auth_UToken": "11111111-dddd-2222-eeee-34567890f098"
}
```

method is the actual function name requested from the service. It must be provided in all requests and has to contain one of the following values:

- query_validation
- connection_test
- auth_emr_user
- audit_list
- report_set
- report_get
- report_list
- report_config_set
- report_config_get
- report_config_list
- report_send_set
- report_send_get
- report_send_list
- config_set

- config_get
- config_list
- account_set
- account_get
- account_list

Sample method message:

```
"method": "query_validation"
```

request is the message containing the request objects specific to the method. The request can contain no values, as well as a single object or array of objects. All request messages must be JSON formatted.

Sample request message:

```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "iyGHRciDe.1GS",
        "Auth_UToken": ""
    },
    "method": "query_validation",
    "request": {
        "emr_id": "emr01",
        "query_params": {},
        "query_statement": "SELECT * FROM TABLENAME"
    }
}
```

In the above message we are authenticating as user `apiuser` with password `iyGHRciDe.1GS` password and auth key `00000000-aaaa-2222-bbbb-34567890cdef`
The message requests data from the query_validation function where the following parameters are provided in the request message:
emr_id – string value of the EMR unique identifier
query_params – array of parameters to be used by the query (in our case the query has no parameters so we are sending a blank array {})
query_statement – SQL statement to be executed against the configured database connection for emr_id `emr01`

Responses

Responses are messages received from the client. All responses include the following 5 major components:

1. AuthHeader
2. method
3. error_code
4. error_message
5. response

AuthHeader is the authentication message object received on every response. It contains the same 4 variables as the request AuthHeader with a few differences:

1. AuthKey – UUID value for your account (key value for the record in the auth_accounts column family). This parameter is returned even if no key was provided in the request and should be used in subsequent requests.
2. Auth_UName – Authentication username
3. Auth_UPass – Authentication password as MD5 hashed string
4. Auth_UToken – UUID value for authentication token. All subsequent requests must include the authentication token received in the response.

error_code is a numeric value with leading 0 (zero) representing the API error code if such event is to occur.

error_message is the message explaining the details of the error for the provided code.

method is a string value representing the function requested from the API and will match the method parameter provided in the request.

response is a JSON formatted message containing a single object, group of objects or array of objects. In case of error, this object will be blank or not present.

Sample response message for our `query_validation` request from above:


```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "d231bdd3e009690bbafac9146788be31",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "error_code": "",
    "error_message": "",
    "method": "query_validation",
    "response": {
        "record_count": 2,
        "results": {
            "1": [
                {
                    "param_name": "first_name",
                    "param_value": "JOHN"
                },
                {
                    "param_name": "last_name",
                    "param_value": "DOE"
                },
                {
                    "param_name": "medrec_id",
                    "param_value": "000000000001"
                }
            ],
            "2": [
                {
                    "param_name": "first_name",
```

```
                    "param_value": "JANE"
                },
                {
                    "param_name": "last_name",
                    "param_value": "DOE"
                },
                {
                    "param_name": "medrec_id",
                    "param_value": "000000000002"
                }
            ]
        }
    }
}
```

In the above response we received the `AuthHeader` objects containing a key (`AuthKey`), requesting API user (`Auth_UName`), MD5 encrypted password (`Auth_UPass`) and authentication token (`Auth_UToken`).
No error was encountered so the `error_code` and `error_message` parameters contain no data.
The method returned, is the method requested: `query_validation`
The `response` segment contains an object with 2 parameters:
`record_count` – number of results returned
`results` – array of results from the requested query

A sample response returning error would look similar to:

```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "d231bdd3e009690bbafac9146788be31",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "error_code": "01788",
    "error_message": "ORA-00942: table or view does not exist",
    "method": "query_validation"
}
```

The API supports the following methods:

API Account Configuration Methods

1. account_set

Purpose: Creates or updates an API account to be used for authentication on remote applications. Note that in order to create an account another account must be already available, as it will fail header authentication.
This account will be used in header authentication messages only and does not represent a physical (individual) account but rather a service account that may be mapped to a support person (application admin or system admin).

Request message objects:

1. AuthHeader
    a. AuthKey - UUID value for your account (key value for the record in the auth_accounts column family)
    b. Auth_UName - Authentication username
    c. Auth_UPass - Authentication clear text password
    d. Auth_UToken - UUID value for authentication token
2. method – value: account_set
3. request - Request object
    a. key - UUID value for account. Leave blank (not nul) if adding a new account
    b. username – username value for account. This must be unique for additions and must match original value if updating an existing account
    c. password - clear text password
    d. account data – object containing user's details. You can specify any set of values needed in String: String format. A few examples used:
        i. address – physical address location (can also be a server name or datacenter)
        ii. account_type – user or admin values (current version does not implement security levels so either value would work)
        iii. email – account email
        iv. first_name – Account primary identification name (application admin)
        v. last_name - Account secondary identification name (application admin)
        vi. phone – additional contact information for system or application admin

Mandatory objects in a request are:
- username
- password
- key
- account_data. Must contain at least the following 3 objects:
    - account_type
    - first_name
    - last_name

Sample request message:

```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "iyGHRciDe.1GS",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
```

```
    "method": "account_set",
    "request": {
        "account_data": {
            "address": "123  Main St., Beverly Hills, CA 90210 USA",
            "account_type": "user",
            "email": "apiusr@domain.tld",
            "first_name": "API",
            "last_name": "Test",
            "phone": "1235554142"
        },
        "key": "",
        "password": "KuiL09-tUbE34",
        "username": "test_api_user"
    }
}
```

In the above message we are creating a new account by not specifying a user key value. The account is “user” level type, with name `test_api_user` and password `KuiL09-tUbE34`.

The response message from our request will include the following objects:

1. AuthHeader
    a. AuthKey - UUID value for your account (key value for the record in the auth_accounts column family)
    b. Auth_UName - Authentication username
    c. Auth_UPass - Authentication md5 hashed password
    d. Auth_UToken - UUID value for authentication token
2. error_code - numeric value with leading 0 (zero) representing the API error code if such event is to occur.
3. error_message - message explaining the details of the error for the provided code.
4. method - value: account_set
5. response – response message object
    a. key - UUID value for account (identical to request value if updating an account)
    b. username – username value for account
    c. password - clear text password
    d. account data – object containing user’s details (same values as in request)

Sample response message:

```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "d231bdd3e009690bbafac9146788be31",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "error_code": "",
    "error_message": "",
    "method": "account_set",
    "response": {
        "account_data": {
            "address": "123  Main St., Beverly Hills, CA 90210 USA",
            "account_type": "user",
```

```
            "email": "apiusr@domain.tld",
            "first_name": "API",
            "last_name": "Test",
            "phone": "1235554142"
        },
        "key": "79c16f90-de69-11e4-8b93-03074fa3d6ec",
        "password": "KuiL09-tUbE34",
        "username": "test_api_user"
    }
}
```

If errors are encountered during requests (e.g.: duplicated account, or insufficient parameters passed) the response will include an error_code and error_message objects:

```
...
    "error_code": "01209",
    "error_message": "Username test_api_user is already in use",
    "method": "account_set",
    "response": {}
...
```

2. account_get

Purpose: Retrieves API account information for an existing record based on key or username.
Request message objects:

1. AuthHeader
    a. AuthKey - UUID value for your account (key value for the record in the auth_accounts column family)
    b. Auth_UName - Authentication username
    c. Auth_UPass - Authentication clear text password
    d. Auth_UToken - UUID value for authentication token
2. method – value: account_get
3. request – Request object
    a. username – username of the account to be retrieved
    OR
    1. key – UUID key value of the account to be retrieved

Sample request:

```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "iyGHRciDe.1GS",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "method": "account_get",
    "request": {
        "username": "test_api_user"
```

```
    }
}
```

In the above message, we are requesting information about username `test_api_user`
Replace `"username": "test_api_user"` with `"key": "79c16f90-de69-11e4-8b93-03074fa3d6ec"` and the same information will be retrieved.

The response message will include the following objects:

1. AuthHeader
    a. AuthKey - UUID value for your account (key value for the record in the auth_accounts column family)
    b. Auth_UName - Authentication username
    c. Auth_UPass - Authentication md5 hashed password
    d. Auth_UToken - UUID value for authentication token
2. error_code - numeric value with leading 0 (zero) representing the API error code if such event is to occur.
3. error_message - message explaining the details of the error for the provided code.
4. method - value: account_get
5. response – response message object
    a. key - UUID value for account (identical to request value if updating an account)
    b. username – username value for account
    c. password - clear text password
    d. account data – object containing user's details

Sample response:

```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "d231bdd3e009690bbafac9146788be31",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "error_code": "",
    "error_message": "",
    "method": "account_get",
    "response": {
        "account_data": {
            "address": "123  Main St., Beverly Hills, CA 90210 USA",
            "account_type": "user",
            "email": "apiusr@domain.tld",
            "first_name": "API",
            "last_name": "Test",
            "phone": "1235554142"
        },
        "key": "79c16f90-de69-11e4-8b93-03074fa3d6ec",
        "password": "cf1abc1d0b3e043538060372a091bbf2",
        "username": "test_api_user"
    }
}
```

Requests for invalid objects or values will return an error_code and error_message value as seen in below example:

```
...
    "error_code": "01215",
    "error_message": "Supplied username value [nouser] returned no results",
    "method": "account_get",
    "response": {}
...
```

3. account_list

Purpose: Retrieves all API accounts.

Request message objects:

1. AuthHeader
    a. AuthKey - UUID value for your account (key value for the record in the auth_accounts column family)
    b. Auth_UName - Authentication username
    c. Auth_UPass - Authentication clear text password
    d. Auth_UToken - UUID value for authentication token
2. method – value: account_list
3. request – null

Sample request:

```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "iyGHRciDe.1GS",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "method": "account_list",
    "request": null
}
```

The response message will include the following objects:

1. AuthHeader
    a. AuthKey - UUID value for your account (key value for the record in the auth_accounts column family)
    b. Auth_UName - Authentication username
    c. Auth_UPass - Authentication md5 hashed password
    d. Auth_UToken - UUID value for authentication token
2. error_code - numeric value with leading 0 (zero) representing the API error code if such event is to occur.
3. error_message - message explaining the details of the error for the provided code.
4. method - value: account_list
5. response – array of account data objects serialized by key value
    a. key – UUID value for user key identifier
        i. key – UUID value for user key identifier

ii. username – username value for account
iii. password – md5 encrypted password
iv. account data – array of account data details

Sample response:


```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "d231bdd3e009690bbafac9146788be31",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "error_code": "",
    "error_message": "",
    "method": "account_list",
    "response": {
        "79c16f90-de69-11e4-8b93-03074fa3d6ec": {
            "account_data": {
                "address": "123  Main St., Beverly Hills, CA 90210 USA",
                "account_type": "user",
                "email": "apiusr@domain.tld",
                "first_name": "API",
                "last_name": "Test",
                "phone": "1235554142"
            },
            "key": "79c16f90-de69-11e4-8b93-03074fa3d6ec",
            "password": "cf1abc1d0b3e043538060372a091bbf2",
            "username": "test_api_user"
        },
        ...
        "ee221c30-de6a-11e4-8b93-03074fa3d6ec": {
            "account_data": {
                "account_type": "user",
                "first_name": "API",
                "last_name": "Test"
            },
            "key": "ee221c30-de6a-11e4-8b93-03074fa3d6ec",
            "password": "cf1abc1d0b3e043538060372a091bbf2",
            "username": "test_api_user4"
        }
    }
}
```


EMR Configuration Methods

1. config_set

Purpose: Sets or updates the configuration for an existing EMR product.

EMR applications are composed of 3 major components: source, authentication and master record all identified by a unique value of emr_id.
Source defines the connectivity protocols to the EMR database platform.
Authentication defines the query that will be used to authenticate an EMR user via UI. This query will generally include two parameters: username and password
Master record defines the general query that will collect information about a patient. This information must include first name, last name and record id (MRN) for the specific EMR platform. This information is used to identify a patient.

Request message objects:

1. AuthHeader
    a. AuthKey - UUID value for your account (key value for the record in the auth_accounts column family)
    b. Auth_UName - Authentication username
    c. Auth_UPass - Authentication clear text password
    d. Auth_UToken - UUID value for authentication token
2. method – value: config_set
3. request – Request object
    a. app_auth – authentication object identified by emr_id
        i. app_auth_path – can be a SQL prepared statement or LDAP path
        ii. app_auth_type – SQL or LDAP
        iii. app_dest – IN or OUT. Version 1.0 of the API only supports IN value
        iv. app_name – Generic EMR application name
    b. app_license – no licensing implemented in version 1.0 of the API. This value must be left blank (“”, not null) and may be used in future releases
    c. app_master_rec – master record object identified by emr_id
        i. master_rec_query – SQL prepared statement to be used to retrieve patient record. This query must include the following objects (defined in lower case):
            - p_f_name – patient’s first name
            - p_l_name – patient’s last name
            - p_m_name – patien’t middle name (if not available, set to blank “”, not null)
            - p_mrn – patient’s medical record number for this EMR
            - p_dob – patient’s date of birth in ‘yyyy-mm-dd’ format
        ii. app_master_rec_params – list of parameters to be passed to the query
    d. app_source – database connection source object identified by emr_id
        i. app_jdbc – jdbc connection string
        ii. app_jdbc_password – password used by the database account for the EMR. Password must be stored in clear text
        iii. app_jdbc_user - username for the database account for the EMR. This account should have read only access to the EMR database
        iv. app_name – Generic EMR application name
    e. key – UUID identifier provided during API installation (will always have the same value)

For LDAP authentication, a full path to the LDAP server including the organization unit (OU) or common name (CN) has to be specified:
`ldap://ldap.domain.local:389/OU=Clin,DC=dc,DC=domain,DC=local`

Windows Active Directory LDAP authentication paths should only include the path to AD server. Example:
`ldap://ad.domain.local:389`

Supported database platforms for JDBC strings:

- Oracle (version 9 and newer)
- Microsoft SQL (version 2005 and newer)

- MySQL (version 4 and newer)
- DB2 (version 9.1 and newer)
- Cache (KB_SQL version 4.5 and newer)

For a list of JDBC examples, see the *connection_test* method.

Sample request message:

```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "iyGHRciDe.1GS",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "method": "config_set",
    "request": {
        "app_auth": {
            "clin01": {
                "app_auth_path": "SELECT USERNAME FROM ENT.USERS WHERE USERNAME = ? AND
PASSWORD = DBMS_OBFUSCATION_TOOLKIT.md5 (input => UTL_RAW.cast_to_raw(?))",
                "app_auth_type": "SQL",
                "app_dest": "IN",
                "app_name": "Clinical app 1.0"
            }
        },
        "app_license": "",
        "app_master_rec": {
            "clin01": {
                "master_rec_query": "SELECT FIRST_NAME AS p_f_name, LAST_NAME AS p_l_name,
MIDDLE_NAME AS p_m_name, MRN AS p_mrn, TO_CHAR(DOB, 'YYYY-MM-DD') AS p_dob FROM MED.PATIENT
WHERE MRN = ? AND PATIENT_ID = ?",
                "master_rec_query_params": {
                    "1": "MRN",
                    "2": "PATIENT_ID"
                }
            }
        },
        "app_source": {
            "clin01": {
                "app_jdbc":
"jdbc:oracle:thin:@(DESCRIPTION=(CONNECT_TIMEOUT=10)(RETRY_COUNT=3)(ADDRESS_LIST=(LOAD_BALANCE
=ON)(FAILOVER=ON)(ADDRESS=(PROTOCOL=TCP)(HOST=oradb-
scan)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=EMRTEST)(FAILOVER_MODE=(TYPE=SELECT)(METHOD=BASI
C)(RETRIES=3)(DELAY=5))))",
                "app_jdbc_password": "Bdi5p-gmYUpM",
                "app_jdbc_user": "rpt_db_user",
                "app_name": " Clinical app 1.0"
            }
        },
        "key": "992d7de0-a699-11e4-86e1-d1c3ddd53489"
    }
```

```
}
```

In the above example we are setting up a new EMR platform identified as clin01 (emr_id) with application name “Clinical app 1.0”. The EMR platform uses SQL authentication against an Oracle RAC instance. Patients are identified by query `SELECT FIRST_NAME AS p_f_name, LAST_NAME AS p_l_name, MIDDLE_NAME AS p_m_name, MRN AS p_mrn, TO_CHAR(DOB, 'YYYY-MM-DD') AS p_dob FROM MED.PATIENT WHERE MRN = ? AND PATIENT_ID = ?` and users authenticated with query `SELECT USERNAME FROM ENT.USERS WHERE USERNAME = ? AND PASSWORD = DBMS_OBFUSCATION_TOOLKIT.md5 (input => UTL_RAW.cast_to_raw(?))`.

Request responses are sent containing the same identical parameters as the request plus the error_code and error_message objects.

1. AuthHeader
    a. AuthKey - UUID value for your account (key value for the record in the auth_accounts column family)
    b. Auth_UName - Authentication username
    c. Auth_UPass - Authentication md5 hashed password
    d. Auth_UToken - UUID value for authentication token
2. error_code - numeric value with leading 0 (zero) representing the API error code if such event is to occur.
3. error_message - message explaining the details of the error for the provided code.
4. method – value: config_set
5. response – Response object
    a. app_auth – authentication object identified by emr_id
        i. app_auth_path – authentication query or LDAP path
        ii. app_auth_type – SQL or LDAP
        iii. app_dest – IN or OUT
        iv. app_name – Generic EMR application name
    b. app_license –blank (“”)
    c. app_master_rec – master record object identified by emr_id
        i. master_rec_query – SQL prepared statement to be used to retrieve patient record.
        ii. app_master_rec_params – list of parameters to be passed to the query
    d. app_source – database connection source object identified by emr_id
        i. app_jdbc – jdbc connection string
        ii. app_jdbc_password – clear text password used by the database account for the EMR.
        iii. app_jdbc_user - username for the database account for the EMR.
        iv. app_name – Generic EMR application name
    e. key – UUID identifier provided during API installation

Sample response message:

```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "d231bdd3e009690bbafac9146788be31",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "error_code": "",
    "error_message": "",
    "method": "config_set",
    "response": {
```

```
        "app_auth": {
            "clin01": {
                "app_auth_path": "SELECT USERNAME FROM ENT.USERS WHERE USERNAME = ? AND
PASSWORD = DBMS_OBFUSCATION_TOOLKIT.md5 (input => UTL_RAW.cast_to_raw(?))",
                "app_auth_type": "SQL",
                "app_dest": "IN",
                "app_name": "Clinical app 1.0"
            }
        },
        "app_license": "",
        "app_master_rec": {
            "clin01": {
                "master_rec_query": "SELECT FIRST_NAME AS p_f_name, LAST_NAME AS p_l_name,
MIDDLE_NAME AS p_m_name, MRN AS p_mrn, TO_CHAR(DOB, 'YYYY-MM-DD') AS p_dob FROM MED.PATIENT
WHERE MRN = ? AND PATIENT_ID = ?",
                "master_rec_query_params": {
                    "1": "MRN",
                    "2": "PATIENT_ID"
                }
            }
        },
        "app_source": {
            "clin01": {
                "app_jdbc":
"jdbc:oracle:thin:@(DESCRIPTION=(CONNECT_TIMEOUT=10)(RETRY_COUNT=3)(ADDRESS_LIST=(LOAD_BALANCE
=ON)(FAILOVER=ON)(ADDRESS=(PROTOCOL=TCP)(HOST=oradb-
scan)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=EMRTEST)(FAILOVER_MODE=(TYPE=SELECT)(METHOD=BASI
C)(RETRIES=3)(DELAY=5))))",
                "app_jdbc_password": "Bdi5p-gmYUpM",
                "app_jdbc_user": "rpt_db_user",
                "app_name": " Clinical app 1.0"
            }
        },
        "key": "992d7de0-a699-11e4-86e1-d1c3ddd53489"
    }
}
```


Errors will be returned if invalid parameters are passed or if duplication of records exists.

2. config_get

Purpose: Gets the configuration for an existing EMR product based on a given emr_id here known as app_id.

Request message objects:

1. AuthHeader
    a. AuthKey - UUID value for your account (key value for the record in the auth_accounts column family)
    b. Auth_UName - Authentication username
    c. Auth_UPass - Authentication clear text password
    d. Auth_UToken - UUID value for authentication token
2. method – value: config_get
3. request – Request object

a. app_id – emr_id value

Sample request message:

```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "iyGHRciDe.1GS",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "method": "config_get",
    "request": {
        "app_id": "clin01"
    }
}
```

In the above request we are retrieving from the API the configuration values for emr_id clin01.

The response message will include the following objects:

1. AuthHeader
    a. AuthKey - UUID value for your account (key value for the record in the auth_accounts column family)
    b. Auth_UName - Authentication username
    c. Auth_UPass - Authentication md5 hashed password
    d. Auth_UToken - UUID value for authentication token
2. error_code - numeric value with leading 0 (zero) representing the API error code if such event is to occur.
3. error_message - message explaining the details of the error for the provided code.
4. method – value: config_set
5. response – Response object
    a. app_auth – authentication object identified by emr_id
        i. app_auth_path – authentication query or LDAP path
        ii. app_auth_type – SQL or LDAP
        iii. app_dest – IN or OUT
        iv. app_name – Generic EMR application name
    b. app_license –blank (“”)
    c. app_master_rec – master record object identified by emr_id
        i. master_rec_query – SQL prepared statement to be used to retrieve patient record.
        ii. app_master_rec_params – list of parameters to be passed to the query
    d. app_source – database connection source object identified by emr_id
        i. app_jdbc – jdbc connection string
        ii. app_jdbc_password – clear text password used by the database account for the EMR.
        iii. app_jdbc_user - username for the database account for the EMR.
        iv. app_name – Generic EMR application name
    e. key – UUID identifier provided during API installation

Sample response message:

```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
```

```
        "Auth_UName": "apiuser",
        "Auth_UPass": "d231bdd3e009690bbafac9146788be31",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "error_code": "",
    "error_message": "",
    "method": "config_get",
    "response": {
        "app_auth": {
            "clin01": {
                "app_auth_path": "SELECT USERNAME FROM ENT.USERS WHERE USERNAME = ? AND
PASSWORD = DBMS_OBFUSCATION_TOOLKIT.md5 (input => UTL_RAW.cast_to_raw(?))",
                "app_auth_type": "SQL",
                "app_dest": "IN",
                "app_name": "Clinical app 1.0"
            }
        },
        "app_license": "",
        "app_master_rec": {
            "clin01": {
                "master_rec_query": "SELECT FIRST_NAME AS p_f_name, LAST_NAME AS p_l_name,
MIDDLE_NAME AS p_m_name, MRN AS p_mrn, TO_CHAR(DOB, 'YYYY-MM-DD') AS p_dob FROM MED.PATIENT
WHERE MRN = ? AND PATIENT_ID = ?",
                "master_rec_query_params": {
                    "1": "MRN",
                    "2": "PATIENT_ID"
                }
            }
        },
        "app_source": {
            "clin01": {
                "app_jdbc":
"jdbc:oracle:thin:@(DESCRIPTION=(CONNECT_TIMEOUT=10)(RETRY_COUNT=3)(ADDRESS_LIST=(LOAD_BALANCE
=ON)(FAILOVER=ON)(ADDRESS=(PROTOCOL=TCP)(HOST=oradb-
scan)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=EMRTEST)(FAILOVER_MODE=(TYPE=SELECT)(METHOD=BASI
C)(RETRIES=3)(DELAY=5))))",
                "app_jdbc_password": "Bdi5p-gmYUpM",
                "app_jdbc_user": "rpt_db_user",
                "app_name": " Clinical app 1.0"
            }
        },
        "key": "992d7de0-a699-11e4-86e1-d1c3ddd53489"
    }
}
```


In the event an error is to occur, the error_code and error_message parameters will be populated. The response object will include blank values for app_auth, app_license, app_master_rec and app_source objects and the default key value will be returned.

Sample error message:

```
...
    "error_code": "01618",
    "error_message": "No results matching app_id clin012",
    "method": "config_get",
    "response": {
        "app_auth": {},
        "app_license": "",
        "app_master_rec": {},
        "app_source": {},
        "key": "992d7de0-a699-11e4-86e1-d1c3ddd53489"
    }
...
```

3. config_list

Purpose: Retrieves the configuration information for all built EMRs

Request message objects:

1. AuthHeader
    a. AuthKey - UUID value for your account (key value for the record in the auth_accounts column family)
    b. Auth_UName - Authentication username
    c. Auth_UPass - Authentication clear text password
    d. Auth_UToken - UUID value for authentication token
2. method – value: config_list
3. request – null

Sample request:

```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "iyGHRciDe.1GS",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "method": "config_list",
    "request": null
}
```

Returned data will contain a response object grouped in arrays of app_auth, app_master_rec and app_source datasets together with app_license and key information. emr_id identification is given for each array object as app_id parameter.

Response objects:

1. AuthHeader
    a. AuthKey - UUID value for your account (key value for the record in the auth_accounts column family)
    b. Auth_UName - Authentication username
    c. Auth_UPass - Authentication md5 hashed password

d. Auth_UToken - UUID value for authentication token
2. error_code - numeric value with leading 0 (zero) representing the API error code if such event is to occur.
3. error_message - message explaining the details of the error for the provided code.
4. method – value: config_list
5. response – Response object
    a. app_auth – array of authentication objects
        i. app_id – emr_id identifier
        ii. app_auth – authentication object
            1. app_auth_path – authentication query or LDAP path
            2. app_auth_type – SQL or LDAP
            3. app_dest – IN or OUT
            4. app_name – Generic EMR application name
    b. app_license – blank (“”)
    c. app_master_rec – array of master record objects
        i. app_id - emr_id identifier
        ii. app_master_rec – master record object
            1. master_rec_query – SQL prepared statement to be used to retrieve patient record.
            2. app_master_rec_params – list of parameters to be passed to the query
    d. app_source – array of database connection source objects
        i. app_id – emr_id identifier
        ii. app_source – database connection object
            1. app_jdbc – jdbc connection string
            2. app_jdbc_password – clear text password used by the database account for the EMR.
            3. app_jdbc_user - username for the database account for the EMR.
            4. app_name – Generic EMR application name
    e. key – UUID identifier provided during API installation

Sample response:


```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "d231bdd3e009690bbafac9146788be31",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "error_code": "",
    "error_message": "",
    "method": "config_list",
    "response": {
        "app_auth": [
            {
                "app_auth": {
                    "app_auth_path": "SELECT USERNAME FROM ENT.USERS WHERE USERNAME = ? AND
PASSWORD = DBMS_OBFUSCATION_TOOLKIT.md5 (input => UTL_RAW.cast_to_raw(?))",
                    "app_auth_type": "SQL",
                    "app_dest": "IN",
                    "app_name": "Clinical app 1.0"
                },
                "app_id": "clin01"
            },
...
```

```
            {
                "app_auth": {
                    "app_auth_path": "SELECT COUNT(USERID) FROM USERS WHERE USERNAME ? AND
PASSWORD ?",
                    "app_auth_type": "SQL",
                    "app_dest": "IN",
                    "app_name": "Optimal Med Rec"
                },
                "app_id": "emr06"
            }
        ],
        "app_license": "",
        "app_master_rec": [
            {
                "app_id": "clin01",
                "app_master_rec": {
                    "master_rec_query": "SELECT FIRST_NAME AS p_f_name, LAST_NAME AS p_l_name,
MIDDLE_NAME AS p_m_name, MRN AS p_mrn, TO_CHAR(DOB, 'YYYY-MM-DD') AS p_dob FROM MED.PATIENT
WHERE MRN = ? AND PATIENT_ID = ?",
                    "master_rec_query_params": {
                        "1": "MRN",
                        "2": "PATIENT_ID"
                    }
                }
            },
...
            {
                "app_id": "emr06",
                "app_master_rec": {
                    "master_rec_query": "SELECT FIRST_NAME, LAST_NAME, DOB FROM PATIENT WHERE
PATIENT_ID = ?",
                    "master_rec_query_params": {
                        "1": "PATIENT_ID"
                    }
                }
            }
        ],
        "app_source": [
            {
                "app_id": "clin01",
                "app_source": {
                    "app_jdbc":
"jdbc:oracle:thin:@(DESCRIPTION=(CONNECT_TIMEOUT=10)(RETRY_COUNT=3)(ADDRESS_LIST=(LOAD_BALANCE
=ON)(FAILOVER=ON)(ADDRESS=(PROTOCOL=TCP)(HOST=oradb-
scan)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=EMRTEST)(FAILOVER_MODE=(TYPE=SELECT)(METHOD=BASI
C)(RETRIES=3)(DELAY=5))))",
                    "app_jdbc_password": "Bdi5p-gmYUpM",
                    "app_jdbc_user": "rpt_db_user",
                    "app_name": " Clinical app 1.0"
                }
            },
```

```
...
            {
                "app_id": "emr06",
                "app_source": {
                    "app_jdbc": "jdbc:db2://db2db.domain.tld:50000/PROD",
                    "app_jdbc_password": "9pbQtuEEegGo7UpP",
                    "app_jdbc_user": "Q3yQQMhTyEXj",
                    "app_name": "Optimal Med Rec"
                }
            }
        ],
        "key": "992d7de0-a699-11e4-86e1-d1c3ddd53489"
    }
}
```


Report Configuration Methods

1. report_config_set

Purpose: Sets or updates the configuration for a report. Reports are built on configuration blocks. Each block represents a section of the report, beginning with a header, body and ending with footer. The unique identifier is report_name and must be provided in all requests. No spaces in the report name are allowed. Spaces (if exist) are automatically replaced by underscores (_). The key value is also a unique identifier and is stored in UUID format. To update an existing report, both key and report_name must be provided and valid (matching).

Due to the complexity of report_config parameter which includes XSL formatted strings, the value must be sent as an array of json objects converted to string value.

Request parameters:

1. AuthHeader
    a. AuthKey - UUID value for your account (key value for the record in the auth_accounts column family)
    b. Auth_UName - Authentication username
    c. Auth_UPass - Authentication clear text password
    d. Auth_UToken - UUID value for authentication token
2. method – value: report_config_set
3. request – request object
    a. emr_id – name of the EMR the report is connected to. This must be a valid entry and always provided
    b. key – UUID value of the report. Leave blank (“”, not null) if creating a new report.
    c. report_name – report name value with no spaces (spaces will be replaced with _). This parameter is required in all instances and must be unique if adding a new report
    d. report_config – report configuration parameters as array of json strings
        i. data_return_type – what dataset is expected in the return: object for single record or array for multiple records
        ii. section_name – name for the defined report block. The value must not include spaces. Blank spaces will be replaced by underscores
        iii. section_order – numeric value of the section (starts at 1 and increments by 1). The values must be sequential
        iv. section_xsl – FO XSL string of report block
        v. data_query – SQL prepared statement to be used against EMR database to retrieve data. A query must be provided in all cases even if no SQL data is required.

vi. data_params – parameters to be passed to SQL prepared statement from data_query. Parameters must be provided as a list numerically identified. Example:

```
"data_params": {
 "1": "PAT_SEQ",
 "2": "MEDREC_ID",
 "3": "CPI_SEQ"
}
```

If no parameters are to be used, then provide a 0 size array {}

Sample request:

```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "iyGHRciDe.1GS",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "method": "report_config_set",
    "request": {
        "emr_id": "clin01",
        "key": "",
        "report_config": "[{\"data_return_type\": \"object\", \"data_params\": {},
\"data_query\": \"SELECT SYSDATE FROM DUAL\", \"section_xsl\": \"<?xml version=\\\"1.0\\\"
encoding=\\\"utf-8\\\"?>\", \"section_name\": \"org_information\", \"section_order\": 1},
{\"data_return_type\": \"object\", \"data_params\": {\"1\": \"PAT_SEQ\", \"3\": \"CPI_SEQ\",
\"2\": \"MEDREC_ID\"}, \"data_query\": \"SELECT FIRST_NAME FROM PATIENT WHERE PAT_SEQ = ? AND
MEDREC_ID = ? AND CPI_SEQ = ?\", \"section_xsl\": \"<fo:block>Hello patient first name:
<xsl:value-of select=\\\"FIRST_NAME\\\"/>!</fo:block>\", \"section_name\": \"report_body\",
\"section_order\": 2}, {\"data_return_type\": \"array\", \"data_params\": {\"1\":
\"CPI_SEQ\"}, \"data_query\": \"SELECT
MEDICATION_STATUS_LSEQ,ADMIN_DOSE_QUANTITY_HI,LAST_TAKEN_DT_STRING,NEXT_DUE_DT_STRING FROM
CPI_MEDICATION WHERE CPI_SEQ = ?\", \"section_xsl\": \"<fo:block></fo:block>\",
\"section_name\": \"medication_information\", \"section_order\": 3}, {\"data_return_type\":
\"object\", \"data_params\": {}, \"data_query\": \"SELECT SYSDATE FROM DUAL\",
\"section_xsl\": \"</xsl:stylesheet>\", \"section_name\": \"footer\", \"section_order\": 4}]",
        "report_name": "discharge_report"
    }
}
```

In the above example we are creating a new configuration for emr_id clin01 by not providing a key value. The report name is discharge_report and includes 4 sections formatted as a json array of strings:

Section 1 – org_information

```
{
    "data_params": {},
    "data_query": "SELECT SYSDATE FROM DUAL",
    "data_return_type": "object",
    "section_name": "org_information",
    "section_order": 1,
```

```
    "section_xsl": "<?xml version=\"1.0\" encoding=\"utf-8\"?> ... "
}
```

Section 2 – report_body

```
{
    "data_params": {
        "1": "PAT_SEQ",
        "2": "MEDREC_ID",
        "3": "CPI_SEQ"
    },
    "data_query": "SELECT FIRST_NAME FROM PATIENT WHERE PAT_SEQ = ? AND MEDREC_ID = ? AND
CPI_SEQ = ?",
    "data_return_type": "object",
    "section_name": "report_body",
    "section_order": 2,
    "section_xsl": " ... <fo:block>Hello patient first name: <xsl:value-of
select=\"FIRST_NAME\"/>!</fo:block> ... "
}
```

Section 3 – medication_information

```
{
    "data_params": {
        "1": "CPI_SEQ"
    },
    "data_query": "SELECT
MEDICATION_STATUS_LSEQ,ADMIN_DOSE_QUANTITY_HI,LAST_TAKEN_DT_STRING,NEXT_DUE_DT_STRING FROM
CPI_MEDICATION WHERE CPI_SEQ = ?",
    "data_return_type": "array",
    "section_name": "medication_information",
    "section_order": 3,
    "section_xsl": " ... <fo:block></fo:block> ... "
}
```

Section 4 – footer

```
{
    "data_params": {},
    "data_query": "SELECT SYSDATE FROM DUAL",
    "data_return_type": "object",
    "section_name": "footer",
    "section_order": 4,
    "section_xsl": " ... </xsl:stylesheet>"
}
```

The response message includes the following objects:

1. AuthHeader
    a. AuthKey - UUID value for your account (key value for the record in the auth_accounts column family)
    b. Auth_UName - Authentication username
    c. Auth_UPass - Authentication md5 hashed password
    d. Auth_UToken - UUID value for authentication token

2. error_code - numeric value with leading 0 (zero) representing the API error code if such event is to occur.
3. error_message - message explaining the details of the error for the provided code.
4. method – value: report_config_set
5. response – Response object
    a. emr_id – name of the EMR the report
    b. key – UUID value of the report
    c. report_name – report name value
    d. report_config – report configuration parameters as array of json objects
        i. data_return_type – dataset return: object or array
        ii. section_name – name for the defined report block
        iii. section_order – numeric value of the section
        iv. section_xsl – FO XSL string of report block
        v. data_query – SQL prepared statement to be used against EMR database to retrieve data
        vi. data_params – parameters to be passed to SQL prepared statement from data_query

Sample response message:


```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "d231bdd3e009690bbafac9146788be31",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "error_code": "",
    "error_message": "",
    "method": "report_config_set",
    "response": {
        "emr_id": "clin01",
        "key": "9d747fa0-c554-11e4-a9e5-c35bb6b29756",
        "report_config": [
            {
                "data_params": {},
                "data_query": "SELECT SYSDATE FROM DUAL",
                "data_return_type": "object",
                "section_name": "org_information",
                "section_order": 1,
                "section_xsl": "<?xml version=\"1.0\" encoding=\"utf-8\"?> ... "
            },
            {
                "data_params": {
                    "1": "PAT_SEQ",
                    "2": "MEDREC_ID",
                    "3": "CPI_SEQ"
                },
                "data_query": "SELECT FIRST_NAME FROM PATIENT WHERE PAT_SEQ = ? AND MEDREC_ID
= ? AND CPI_SEQ = ?",
                "data_return_type": "object",
                "section_name": "report_body",
                "section_order": 2,
                "section_xsl": " ... <fo:block>Hello patient first name: <xsl:value-of
select=\"FIRST_NAME\"/>!</fo:block> ... "
```

```
            },
            {
                "data_params": {
                    "1": "CPI_SEQ"
                },
                "data_query": "SELECT
MEDICATION_STATUS_LSEQ,ADMIN_DOSE_QUANTITY_HI,LAST_TAKEN_DT_STRING,NEXT_DUE_DT_STRING FROM
CPI_MEDICATION WHERE CPI_SEQ = ?",
                "data_return_type": "array",
                "section_name": "medication_information",
                "section_order": 3,
                "section_xsl": " ... <fo:block></fo:block> ... "
            },
            {
                "data_params": {},
                "data_query": "SELECT SYSDATE FROM DUAL",
                "data_return_type": "object",
                "section_name": "footer",
                "section_order": 4,
                "section_xsl": " ... </xsl:stylesheet>"
            }
        ],
        "report_name": "discharge_report"
    }
}
```

2. report_config_get

Purpose: Gets the configuration for a report. Reports are identified by a key or report_name value.

Request parameters:

1. AuthHeader
    a. AuthKey - UUID value for your account (key value for the record in the auth_accounts column family)
    b. Auth_UName - Authentication username
    c. Auth_UPass - Authentication clear text password
    d. Auth_UToken - UUID value for authentication token
2. method – value: report_config_get
3. request – request object
    a. report_name: string value of report name (no spaces)
    OR
    a. key: UUID value for report key

Example request for a request for discharge_report report:

```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "iyGHRciDe.1GS",
```

```
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "method": "report_config_get",
    "request": {
        "report_name": "discharge_report"
    }
}
```

To retrieve a `report based on key, simply use a request like:

```
...
    "request": {
        "key": "9d747fa0-c554-11e4-a9e5-c35bb6b29756"
    }
}
```

The response message will include the following objects:

1. AuthHeader
    a. AuthKey - UUID value for your account (key value for the record in the auth_accounts column family)
    b. Auth_UName - Authentication username
    c. Auth_UPass - Authentication md5 hashed password
    d. Auth_UToken - UUID value for authentication token
2. error_code - numeric value with leading 0 (zero) representing the API error code if such event is to occur.
3. error_message - message explaining the details of the error for the provided code.
4. method – value: report_config_set
5. response – Response object
    a. emr_id – name of the EMR the report
    b. key – UUID value of the report
    c. report_name – report name value
    d. report_config – report configuration parameters as array of json objects
        i. data_return_type – dataset return: object or array
        ii. section_name – name for the defined report block
        iii. section_order – numeric value of the section
        iv. section_xsl – FO XSL string of report block
        v. data_query – SQL prepared statement to be used against EMR database to retrieve data
        vi. data_params – parameters to be passed to SQL prepared statement from data_query

Sample response message:

```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "d231bdd3e009690bbafac9146788be31",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "error_code": "",
    "error_message": "",
```

```
    "method": "report_config_get",
    "response": {
        "emr_id": "clin01",
        "key": "9d747fa0-c554-11e4-a9e5-c35bb6b29756",
        "report_config": [
            {
                "data_params": {},
                "data_query": "SELECT SYSDATE FROM DUAL",
                "data_return_type": "object",
                "section_name": "org_information",
                "section_order": 1,
                "section_xsl": "<?xml version=\"1.0\" encoding=\"utf-8\"?>"
            },
            {
                "data_params": {
                    "1": "PAT_SEQ",
                    "2": "MEDREC_ID",
                    "3": "CPI_SEQ"
                },
                "data_query": "SELECT FIRST_NAME FROM PATIENT WHERE PAT_SEQ = ? AND MEDREC_ID
= ? AND CPI_SEQ = ?",
                "data_return_type": "object",
                "section_name": "report_body",
                "section_order": 2,
                "section_xsl": "<fo:block>Hello patient first name: <xsl:value-of
select=\"FIRST_NAME\"/>!</fo:block>"
            },
            {
                "data_params": {
                    "1": "CPI_SEQ"
                },
                "data_query": "SELECT
MEDICATION_STATUS_LSEQ,ADMIN_DOSE_QUANTITY_HI,LAST_TAKEN_DT_STRING,NEXT_DUE_DT_STRING FROM
CPI_MEDICATION WHERE CPI_SEQ = ?",
                "data_return_type": "array",
                "section_name": "medication_information",
                "section_order": 3,
                "section_xsl": "<fo:block></fo:block>"
            },
            {
                "data_params": {},
                "data_query": "SELECT SYSDATE FROM DUAL",
                "data_return_type": "object",
                "section_name": "footer",
                "section_order": 4,
                "section_xsl": "</xsl:stylesheet>"
            }
        ],
        "report_name": "discharge_report"
    }
}
```

In case of error, no response object will be present and values for error_message and error_code will be returned.

3. report_config_list

Purpose: Gets a list of all configured reports.

Request parameters:

1. AuthHeader
    a. AuthKey - UUID value for your account (key value for the record in the auth_accounts column family)
    b. Auth_UName - Authentication username
    c. Auth_UPass - Authentication clear text password
    d. Auth_UToken - UUID value for authentication token
2. method – value: report_config_list
3. request – empty object

Sample request:

```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "iyGHRciDe.1GS",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "method": "report_config_list",
    "request": {}
}
```

The response message will include the following objects:

1. AuthHeader
    a. AuthKey - UUID value for your account (key value for the record in the auth_accounts column family)
    b. Auth_UName - Authentication username
    c. Auth_UPass - Authentication md5 hashed password
    d. Auth_UToken - UUID value for authentication token
2. error_code - numeric value with leading 0 (zero) representing the API error code if such event is to occur.
3. error_message - message explaining the details of the error for the provided code.
4. method – value: report_config_list
5. response – array of report configuration objects
    a. emr_id – name of the EMR the report
    b. key – UUID value of the report
    c. report_name – report name value
    d. report_config – report configuration parameters as array of json objects
        i. data_return_type – dataset return: object or array
        ii. section_name – name for the defined report block
        iii. section_order – numeric value of the section
        iv. section_xsl – FO XSL string of report block
        v. data_query – SQL prepared statement to be used against EMR database to retrieve data
        vi. data_params – parameters to be passed to SQL prepared statement from data_query

Sample response message:

```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "d231bdd3e009690bbafac9146788be31",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "error_code": "",
    "error_message": "",
    "method": "report_config_list",
    "response": {
        "reports": [
            {
                "emr_id": "clin01",
                "key": "9d747fa0-c554-11e4-a9e5-c35bb6b29756",
                "report_config": [
                    {
                        "data_params": {},
                        "data_query": "SELECT SYSDATE FROM DUAL",
                        "data_return_type": "object",
                        "section_name": "org_information",
                        "section_order": 1,
                        "section_xsl": "<?xml version=\"1.0\" encoding=\"utf-8\"?>"
                    },
                    {
                        "data_params": {
                            "1": "PAT_SEQ",
                            "2": "MEDREC_ID",
                            "3": "CPI_SEQ"
                        },
                        "data_query": "SELECT FIRST_NAME FROM PATIENT WHERE PAT_SEQ = ? AND
MEDREC_ID = ? AND CPI_SEQ = ?",
                        "data_return_type": "object",
                        "section_name": "report_body",
                        "section_order": 2,
                        "section_xsl": "<fo:block>Hello patient first name: <xsl:value-of
select=\"FIRST_NAME\"/>!</fo:block>"
                    },
                    {
                        "data_params": {
                            "1": "CPI_SEQ"
                        },
                        "data_query": "SELECT
MEDICATION_STATUS_LSEQ,ADMIN_DOSE_QUANTITY_HI,LAST_TAKEN_DT_STRING,NEXT_DUE_DT_STRING FROM
CPI_MEDICATION WHERE CPI_SEQ = ?",
                        "data_return_type": "array",
                        "section_name": "medication_information",
                        "section_order": 3,
                        "section_xsl": "<fo:block></fo:block>"
                    },
                    {
```

```
                        "data_params": {},
                        "data_query": "SELECT SYSDATE FROM DUAL",
                        "data_return_type": "object",
                        "section_name": "footer",
                        "section_order": 4,
                        "section_xsl": "</xsl:stylesheet>"
                    }
                ],
                "report_name": "discharge_report"
            },
            ...
        ]
    }
}
```

Report Generation Methods

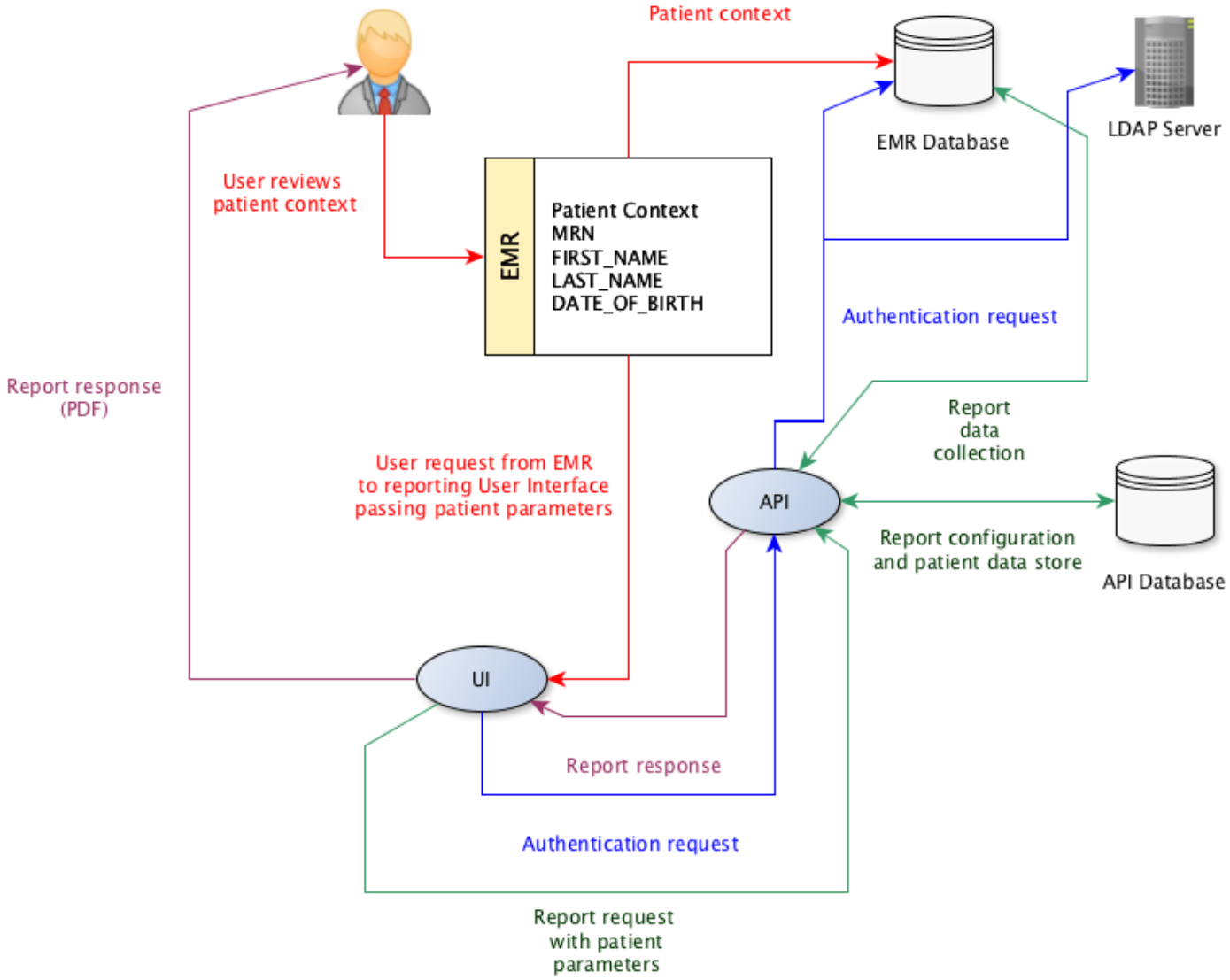
1. report_set

Purpose: Retrieves patient data and compiles it a report.
This is the report generation function where a pre-configured report (see report_config_set function) is combined with EMR data into an output PDF binary stream.

Modus Operandi (Method of Operation):

User sends data to UI via GET method directly from EMR pre-configured link (button or menu option).
UI validates user against EMR database.
Upon successful authentication, UI passes patient context parameters to API for processing together with desired report.
Report configuration is retrieved from API database
Patient data is collected from EMR database based on report configuration
Data stream (XML) and report format (XSL) are processed for report (PDF)
Report data and format is saved in API database and a new record (if patient not found) is created
PDF report is returned to UI for display as base64 encoded byte array.

Data flow and operation diagram:



Request parameters:

1. AuthHeader
    a. AuthKey - UUID value for your account (key value for the record in the auth_accounts column family)
    b. Auth_UName - Authentication username
    c. Auth_UPass - Authentication clear text password
    d. Auth_UToken - UUID value for authentication token
2. method – value: report_set
3. request – request object
    a. emr_user_id – username used during UI authentication process. This parameter is required for auditing purposes
    b. p_emr_id – patient EMR ID (EMR unique identifier as described in config_set section
    c. report_name – value of report name (as described in report_config_set section)

d. report_params – list of report parameters in “string: parameter name”: “string: parameter value” format. All combined parameters for a given report must be provided. For example if the report configuration has
```
"data_params": {
       "1": "PAT_SEQ",
       "2": "MEDREC_ID",
       "3": "CPI_SEQ"
}
```
then the following parameters must be provided in format:
```
"report_params": {
       "PAT_SEQ": "000002",
       "MEDREC_ID": "000003",
       "CPI_SEQ": "000004"
}
```

Example request message:

```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "iyGHRciDe.1GS",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "method": "report_set",
    "request": {
        "emr_user_id": "emruser01",
        "p_emr_id": "emr02",
        "report_name": "test_rpt_02",
        "report_params": {
            "MEDS_END": "'2016-01-01 00:00:00'",
            "MEDS_START": "'2015-01-01 00:00:00'",
            "PATIENT_ID": "3"
        }
    }
}
```

In the above example, EMR user “emruser01” for configured EMR ID “emr02” is requesting pre-configured report “test_rpt_2”: using 3 patient parameters (patient id, medication start and end).

The response will include the following objects:

1. AuthHeader
    a. AuthKey - UUID value for your account (key value for the record in the auth_accounts column family)
    b. Auth_UName - Authentication username
    c. Auth_UPass - Authentication md5 hashed password
    d. Auth_UToken - UUID value for authentication token
2. error_code - numeric value with leading 0 (zero) representing the API error code if such event is to occur.
3. error_message - message explaining the details of the error for the provided code.
4. method – value: report_set
5. response – list of report data objects
    a. emr_user_id – same as requested emr_user_id value
    b. key – patient UUID key value

c. p_dob – patient date of birth (defined in config_set app_master_rec query)
d. p_emr_id – EMR unique identifier (same as request p_emr_id)
e. p_f_name – patient first name (defined in config_set app_master_rec query)
f. p_l_name – patient last name (defined in config_set app_master_rec query)
g. p_m_name – patient middle name (defined in config_set app_master_rec query)
h. p_mrn – unique identifier value for the given EMR (defined in config_set app_master_rec query)
i. pdf_data – base64 encoded string of PDF byte array
j. xml_data – XML string of collected report data
k. xsl_data – XSL string of report format

Sample message response (long strings truncated):


```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "d231bdd3e009690bbafac9146788be31",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "error_code": "",
    "error_message": "",
    "method": "report_set",
    "response": {
        "emr_user_id": "emruser01",
        "key": "0443d6b0-c86e-11e4-a52b-03074fa3d6ec",
        "p_dob": " ",
        "p_emr_id": "emr02",
        "p_f_name": "JIM",
        "p_l_name": "JAMES",
        "p_m_name": " ",
        "p_mrn": "3",
        "pdf_data": "JVBERi ... GCg==",
        "report_name": "test_rpt_02",
        "xml_data": "<?xml version=\"1.0\" ... </test_rpt_02>",
        "xsl_data": "<?xml version=\"1.0\" ... </xsl:stylesheet>"
    }
}
```


2. *report_get*

Purpose: Retrieves existing report based on patient key and report generation time.

Request parameters:

1. AuthHeader
    a. AuthKey - UUID value for your account (key value for the record in the auth_accounts column family)
    b. Auth_UName - Authentication username
    c. Auth_UPass - Authentication clear text password
    d. Auth_UToken - UUID value for authentication token
2. method – value: report_get
3. request – request parameters

a. key – UUID formatted patient key (API database, not EMR patient identifier)
b. report_date – UTC formatted report timestamp (example: Thu Mar 12 04:18:25 UTC 2015)
c. emr_user_id – username used during UI authentication process. This parameter is required for auditing purposes

Sample request:

```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "iyGHRciDe.1GS",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "method": "report_get",
    "request": {
        "key": "0443d6b0-c86e-11e4-a52b-03074fa3d6ec",
        "report_date": "Thu Mar 12 04:18:25 UTC 2015",
        "emr_user_id": "emruser01"
    }
}
```

In the above example, EMR user id emruser01 is requesting a report generated on `Thu Mar 12 04:18:25 UTC 2015` for patient key `d43f3f30-c86e-11e4-8603-c35bb6b29756`

Response objects are identical to response message for report_set method:

1. AuthHeader
    a. AuthKey - UUID value for your account (key value for the record in the auth_accounts column family)
    b. Auth_UName - Authentication username
    c. Auth_UPass - Authentication md5 hashed password
    d. Auth_UToken - UUID value for authentication token
2. error_code - numeric value with leading 0 (zero) representing the API error code if such event is to occur.
3. error_message - message explaining the details of the error for the provided code.
4. method – value: report_get
5. response – list of report data objects
    a. emr_user_id – same as requested emr_user_id value
    b. key – patient UUID key value
    c. p_dob – patient date of birth (defined in config_set app_master_rec query)
    d. p_emr_id – EMR unique identifier (same as request p_emr_id)
    e. p_f_name – patient first name (defined in config_set app_master_rec query)
    f. p_l_name – patient last name (defined in config_set app_master_rec query)
    g. p_m_name – patient middle name (defined in config_set app_master_rec query)
    h. p_mrn – unique identifier value for the given EMR (defined in config_set app_master_rec query)
    i. pdf_data – base64 encoded string of PDF byte array
    j. xml_data – XML string of collected report data
    k. xsl_data – XSL string of report format

Sample message response (long strings truncated):

```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "d231bdd3e009690bbafac9146788be31",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "error_code": "",
    "error_message": "",
    "method": "report_get",
    "response": {
        "emr_user_id": "emruser01",
        "key": "0443d6b0-c86e-11e4-a52b-03074fa3d6ec",
        "p_dob": " ",
        "p_emr_id": "emr02",
        "p_f_name": "JIM",
        "p_l_name": "JAMES",
        "p_m_name": " ",
        "p_mrn": "3",
        "pdf_data": "JVBERi ... GCg==",
        "report_name": "test_rpt_02",
        "xml_data": "<?xml version=\"1.0\" ... </test_rpt_02>",
        "xsl_data": "<?xml version=\"1.0\" ... </xsl:stylesheet>"
    }
}
```


In case of error, the returned error_code and error_message will present values and no response object will be available.

Sample error message for a request where the report date parameter is invalid:


```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "d231bdd3e009690bbafac9146788be31",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "error_code": "01793",
    "error_message": "Could not generate report",
    "method": "report_get"
}
```


3. report_list

Purpose: Lists all reports already created for all patients.

Request parameters:

1. AuthHeader

   a. AuthKey - UUID value for your account (key value for the record in the auth_accounts column family)
   b. Auth_UName - Authentication username
   c. Auth_UPass - Authentication clear text password
   d. Auth_UToken - UUID value for authentication token
2. method – value: report_list
3. request – empty object

Sample request:

```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "iyGHRciDe.1GS",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "method": "report_list",
    "request": {}
}
```

The response message will contain the following parameters:

1. AuthHeader
   a. AuthKey - UUID value for your account (key value for the record in the auth_accounts column family)
   b. Auth_UName - Authentication username
   c. Auth_UPass - Authentication md5 hashed password
   d. Auth_UToken - UUID value for authentication token
2. error_code - numeric value with leading 0 (zero) representing the API error code if such event is to occur.
3. error_message - message explaining the details of the error for the provided code.
4. method – value: report_list
5. response – array of response objects encapsulated in a response object
   a. key – UUID value of patient key
   b. p_details – timestamp indexed list of patent details in param_name: param_value format
   c. p_dob – patient date of birth (defined in config_set app_master_rec query)
   d. p_emr_id – EMR unique identifier (same as request p_emr_id)
   e. p_f_name – patient first name (defined in config_set app_master_rec query)
   f. p_l_name – patient last name (defined in config_set app_master_rec query)
   g. p_m_name – patient middle name (defined in config_set app_master_rec query)
   h. p_mrn – unique identifier value for the given EMR (defined in config_set app_master_rec query)
   i. report_data – report information indexes by report creation timestamp excluding PDF byte array stream
      i. rpt_requester – EMR user ID creating report
      ii. rpt_xml - XML string of collected report data
      iii. rpt_xsl - XSL string of report format

Sample message response (long strings and repetitive objects truncated):

```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "d231bdd3e009690bbafac9146788be31",
```

```
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "method": "report_list",
    "response": {
        "response": [
            {
                "key": "0af04860-c86c-11e4-8603-c35bb6b29756",
                "p_details": [
                    {
                        "Thu Mar 12 03:58:28 UTC 2015":
"[{\"param_name\":\"p_f_name\",\"param_value\":\"JOHN\"} ...
{\"param_name\":\"p_l_name\",\"param_value\":\"DOE\"}
                    },
                    ...
                    {
                        "Thu Mar 12 04:02:15 UTC 2015":
"[{\"param_name\":\"p_f_name\",\"param_value\":\"JOHN\"} ...
{\"param_name\":\"p_l_name\",\"param_value\":\"DOE\"}
                    }
                ],
                "p_dob": " ",
                "p_emr_id": "emr01",
                "p_f_name": "JOHN",
                "p_l_name": "DOE",
                "p_m_name": "TEST",
                "p_mrn": "000000000001",
                "report_data": [
                    {
                        "Thu Mar 12 03:58:28 UTC 2015": {
                            "rpt_requester": "emruser01",
                            "rpt_xml": "<?xml version=\"1.0\" ... ",
                            "rpt_xsl": "<?xml version=\"1.0\" ... "
                        }
                    },
                    ...
                    {
                        "Thu Mar 12 04:02:15 UTC 2015": {
                            "rpt_requester": "emruser01",
                            "rpt_xml": "<?xml version=\"1.0\" ... ",
                            "rpt_xsl": "<?xml version=\"1.0\" ... "
                        }
                    }
                ]
            },
            {
                "key": "0443d6b0-c86e-11e4-a52b-03074fa3d6ec",
                "p_details": [
                    ...
                ]
            }
        ]
```

```
    }
}
```

Auxiliary Methods

This is a set of methods designed for validation purposes. None of the below methods make changes to application data.

1. ***query_validation***

Purpose: Allows a client to validate a query for a given EMR and set of parameters. This function should be used only to validate queries against the source database and NOT for mass extracts of data. For this reason, the number of returned records is limited to 50.

Example request for a query against emr01 that has no parameters:

```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "iyGHRciDe.1GS",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "method": "query_validation",
    "request": {
        "emr_id": "emr01",
        "query_params": {},
        "query_statement": "SELECT FIRST_NAME, LAST_NAME, MEDREC_ID FROM ENT.PATIENT WHERE
ROWNUM < 3"
    }
}
```

Example response:

```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "d231bdd3e009690bbafac9146788be31",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "error_code": "",
    "error_message": "",
    "method": "query_validation",
    "response": {
        "record_count": 2,
        "results": {
```

```
            "1": [
                {
                    "param_name": "first_name",
                    "param_value": "JOHN"
                },
                {
                    "param_name": "last_name",
                    "param_value": "DOE"
                },
                {
                    "param_name": "medrec_id",
                    "param_value": "000000000001"
                }
            ],
            "2": [
                {
                    "param_name": "first_name",
                    "param_value": "JANE"
                },
                {
                    "param_name": "last_name",
                    "param_value": "DOE"
                },
                {
                    "param_name": "medrec_id",
                    "param_value": "000000000002"
                }
            ]
        }
    }
}
```

To pass parameters to a query validation request, use the following request format:

```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "iyGHRciDe.1GS",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "method": "query_validation",
    "request": {
        "emr_id": "emr02",
        "query_params": {
            "1": "3",
            "2": "JIM"
        },
        "query_statement": "SELECT * FROM PATIENT WHERE PATIENT_ID = ? AND FIRST_NAME = ?"
    }
```

```
}
```

In this case, we pass 2 parameters to our prepared statement: 1 (PATIENT_ID) with a value of 3 and 2 (FIRST_NAME) with a value of JIM

Our sample response would look like:

```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "d231bdd3e009690bbafac9146788be31",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "error_code": "",
    "error_message": "",
    "method": "query_validation",
    "response": {
        "record_count": 1,
        "results": {
            "1": [
                {
                    "param_name": "patient_id",
                    "param_value": "3"
                },
                {
                    "param_name": "first_name",
                    "param_value": "JIM"
                },
                {
                    "param_name": "last_name",
                    "param_value": "JAMES"
                }
            ]
        }
    }
}
```

If the request returns no results, an error message will be present in the response:

```
...
    "error_code": "01788",
    "error_message": "No records returned",
    "method": "query_validation"
...
```

2. *connection_test*

Purpose: This function is designed to validate connectivity to a database source. The following database platforms are supported:

- Oracle (version 9 and newer)
- Microsoft SQL (version 2005 and newer)
- MySQL (version 4 and newer)
- DB2 (version 9.1 and newer)
- Cache (KB_SQL version 4.5 and newer)

The connection test method is a pure JDBC implementation where a jdbc string together with credentials is provided for connectivity test. Based on the data source type, a platform specific query is executed to ensure not only that connectivity is made but also that proper permissions are available for the connecting user. Based on platform, the following queries are executed:

Platform:	Query:
Oracle	`SELECT COUNT(*) AS MYCNT FROM DUAL;`
Microsoft SQL	`SELECT COUNT(@@VERSION) AS MYCNT;`
MySQL	`SELECT COUNT(*) AS MYCNT FROM INFORMATION_SCHEMA.PROCESSLIST;`
DB2	`SELECT COUNT(service_level) AS MYCNT FROM TABLE (sysproc.env_get_inst_info()) as INSTANCEINFO;`
Cache	`N/A`

If connectivity is established and the query will return a value greater than 0 for the `MYCNT` column then validation occurred. Otherwise a native SQL error message will be returned.

Sample JDBC strings to be used:

Platform:	JDBC String:
Oracle	`jdbc:oracle:thin:@orahost.domain.tld:1521:emr_test`
Oracle (RAC)	`jdbc:oracle:thin:@(DESCRIPTION=(CONNECT_TIMEOUT=10)(RETRY_COUNT=3)(ADDRESS_LIST=(LOAD_BALANCE=ON)(FAILOVER=ON)(ADDRESS=(PROTOCOL=TCP)(HOST=ora-scan.domain.tld)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=EMR_TEST)(FAILOVER_MODE=(TYPE=SELECT)(METHOD=BASIC)(RETRIES=3)(DELAY=5))))`
Microsoft SQL	`jdbc:sqlserver://mssqlhost.domain.tld:1433;DatabaseName=emr_test`
MySQL	`jdbc:mysql://mysqlhost.domain.tld:3306/emr_test`
DB2	`jdbc:db2://db2db.domain.tld:50000/EMR_TEST`
Cache	`jdbc:kbsjdbc://cachedb01.domain.tld:5080`

Because the connectivity tests module is used in conjunction with an EMR configuration, the following parameter objects have to be provided in an API request for validation:

4. AuthHeader
    a. AuthKey - UUID value for your account (key value for the record in the auth_accounts column family)
    b. Auth_UName - Authentication username
    c. Auth_UPass - Authentication clear text password
    d. Auth_UToken - UUID value for authentication token
5. method – value: connection_test

6. request
    a. app_auth – EMR application authentication method
        i. emr_id – defined emr_id
            1. app_auth_path – can be a SQL select statement for user identification or an LDAP path
            2. app_auth_type – SQL (for SQL connections) or LDAP (for LDAP servers)
            3. app_destination – this value will always be IN. Future API releases may include an OUT value for connections to imaging systems or external destinations (FTP, HTTP, MAIL, etc.)
            4. app_name – short name for the defined EMR
    b. app_license – no value required. Future API releases will implement licensing validation
    c. app_source –
        i. emr_id – defined emr_id
            1. app_jdbc - JDBC string to EMR database
            2. app_jdbc_user - database user account (read-only account preferred)
            3. app_jdbc_password - clear text password for database user account
            4. app_name – short name for the defined EMR
    d. key – UUID value for your EMR configuration record. This value will be provided during API installation and configuration and will always have the same value

Sample request:

```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "iyGHRciDe.1GS",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "method": "connection_test",
    "request": {
        "app_auth": {
            "emr01": {
                "app_auth_path": "SELECT USERNAME FROM ENT_CONFIG.STAFF WHERE USERNAME = ? AND
PASSWORD = DBMS_OBFUSCATION_TOOLKIT.md5 (input => UTL_RAW.cast_to_raw(?))",
                "app_auth_type": "SQL",
                "app_dest": "IN",
                "app_name": "Test EMR"
            }
        },
        "app_license": "",
        "app_source": {
            "emr01": {
                "app_jdbc":
"jdbc:oracle:thin:@(DESCRIPTION=(CONNECT_TIMEOUT=10)(RETRY_COUNT=3)(ADDRESS_LIST=(LOAD_BALANCE
=ON)(FAILOVER=ON)(ADDRESS=(PROTOCOL=TCP)(HOST=ora-
scan.domain.tld)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=EMR_TEST)(FAILOVER_MODE=(TYPE=SELECT)
(METHOD=BASIC)(RETRIES=3)(DELAY=5))))",
                "app_jdbc_password": "mo4cXv5wNUSLDK",
                "app_jdbc_user": "sql_ro_dcrpt_user",
                "app_name": "Test EMR"
```

```
            }
        },
        "key": "dc8292e0-d8e6-11e4-8d1a-c35bb6b29756"
    }
}
```

In the above sample request we are defining an EMR with ID emr01 and an application name “Test EMR”.
This application uses an SQL IN authentication query and connects to an Oracle RAC instance.

The response message will include the following objects:

6. AuthHeader
    a. AuthKey - UUID value for your account (key value for the record in the auth_accounts column family)
    b. Auth_UName - Authentication username
    c. Auth_UPass - Authentication md5 hashed password
    d. Auth_UToken - UUID value for authentication token
7. error_code - numeric value with leading 0 (zero) representing the API error code if such event is to occur.
8. error_message - message explaining the details of the error for the provided code.
9. method - value: connection_test
10. response – response message object
    a. app_jdbc – JDBC connection string identical to the value passed in request
    b. app_jdbc_user - database user account
    c. app_jdbc_password - clear text password for database user account
    d. connection_status – `"valid"` for valid connections, `"invalid"` for invalid connection. `"invalid"` response will also have details in the error_code and error_message objects

Sample response:

```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "d231bdd3e009690bbafac9146788be31",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "error_code": "",
    "error_message": "",
    "method": "connection_test",
    "response": {
        "app_jdbc":
"jdbc:oracle:thin:@(DESCRIPTION=(CONNECT_TIMEOUT=10)(RETRY_COUNT=3)(ADDRESS_LIST=(LOAD_BALANCE
=ON)(FAILOVER=ON)(ADDRESS=(PROTOCOL=TCP)(HOST=ora-
scan.domain.tld)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=EMR_TEST)(FAILOVER_MODE=(TYPE=SELECT)
(METHOD=BASIC)(RETRIES=3)(DELAY=5))))",
        "app_jdbc_password": "mo4cXv5wNUSLDK",
        "app_jdbc_user": "sql_ro_dcrpt_user",
        "connection_status": "valid"
    }
}
```

Sample error response:

```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "d231bdd3e009690bbafac9146788be31",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "error_code": "01412",
    "error_message": "ORA-01017: invalid username/password; logon denied\n",
    "method": "connection_test",
    "response": {
        "app_jdbc": "",
        "app_jdbc_password": "",
        "app_jdbc_user": "",
        "connection_status": "invalid"
    }
}
```


3. auth_emr_user

Purpose: This function is designed to validate an EMR user against the source database. This function is required for the user interface to process users requesting via web patient reports.
Authentication protocols (SQL or LDAP) are defined in the config_set function

The following parameters must be provided for EMR authentication:

1. AuthHeader
    a. AuthKey - UUID value for your account (key value for the record in the auth_accounts column family)
    b. Auth_UName - Authentication username
    c. Auth_UPass - Authentication clear text password
    d. Auth_UToken - UUID value for authentication token
2. method – value: auth_emr_user
3. request
    a. emr_id – defined emr_id
    b. username – EMR account username
    c. password – clear text password for EMR account

Sample request:


```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "iyGHRciDe.1GS",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "method": "auth_emr_user",
    "request": {
        "emr_id": "emr01",
```

```
        "password": "qwKS0x6w2baK",
        "username": "dr001emracct"
    }
}
```

In the above sample request we are asking the API to perform an account validation against the source EMR (emr01) for username `dr001emracct` with password `qwKS0x6w2baK`.

The return message will contain the following objects:

1. AuthHeader
    a. AuthKey - UUID value for your account (key value for the record in the auth_accounts column family)
    b. Auth_UName - Authentication username
    c. Auth_UPass - Authentication md5 hashed password
    d. Auth_UToken - UUID value for authentication token
2. error_code - numeric value with leading 0 (zero) representing the API error code if such event is to occur.
3. error_message - message explaining the details of the error for the provided code.
4. method - value: auth_emr_user
5. response – response message object
    a. app_auth_type – authentication methods used (either SQL or LDAP)
    b. emr_id – pre-configured emr_id
    c. username – EMR account username
    d. valid – `true` if the account and password were valid, `false` if invalid credentials were provided. A value of `false` will also have details in the error_code and error_message objects.

Sample response:

```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "d231bdd3e009690bbafac9146788be31",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "error_code": "",
    "error_message": "",
    "method": "auth_emr_user",
    "response": {
        "app_auth_type": "SQL",
        "emr_id": "emr01",
        "username": "dr001emracct",
        "valid": "true"
    }
}
```

Sample error response:

```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
```

```
        "Auth_UName": "apiuser",
        "Auth_UPass": "d231bdd3e009690bbafac9146788be31",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "error_code": "01224",
    "error_message": "Invalid account information",
    "method": "auth_emr_user",
    "response": {
        "app_auth_type": "LDAP",
        "emr_id": "emr05",
        "username": "dr001emracct",
        "valid": "false"
    }
}
```

Audit Methods

1. audit_list

Purpose: This function is designed to provide a list of audit items for a given user or patient.

The following parameters must be provided for audit list retrieval:

1. AuthHeader
    a. AuthKey - UUID value for your account (key value for the record in the auth_accounts column family)
    b. Auth_UName - Authentication username
    c. Auth_UPass - Authentication clear text password
    d. Auth_UToken - UUID value for authentication token
2. method – value: audit_list
3. request
    a. patient_key – pre-defined patient key (UUID value for patient record from patient_report columnfamily)
    b. user_key – pre-defined API user key (UUID key value for user from auth_accounts columnfamily)

Sample request:

```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "iyGHRciDe.1GS",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "method": "audit_list",
    "request": {
        "patient_key": "",
        "user_key": "278cbcb0-c6e6-11e4-8246-4f0ac8a69cff"
    }
}
```

In the above sample request we are requesting all audits for a given API user ID `278cbcb0-c6e6-11e4-8246-4f0ac8a69cff`

The return message will contain the following objects:

1. AuthHeader
    a. AuthKey - UUID value for your account (key value for the record in the auth_accounts column family)
    b. Auth_UName - Authentication username
    c. Auth_UPass - Authentication md5 hashed password
    d. Auth_UToken - UUID value for authentication token
2. error_code - numeric value with leading 0 (zero) representing the API error code if such event is to occur.
3. error_message - message explaining the details of the error for the provided code.
4. method - value: audit_list
5. response – response message object
    a. audit_info – array object of audit data
        i. audit_data – audit data object
            1. message – Audit message
            2. old_value – old value for parameter. This is present only if a change was performed
            3. new_value – new value for parameter. This is present only if a change or addition was performed.
        ii. timestamp – UTC timestamp value in “YYYY-MM-DD<T>HH24:MI:SS.sss<Z>” format
    b. user_key - UUID key value for user from auth_accounts columnfamily. This value is returned only if the initial request included a user_key value. Otherwise this parameter will not be returned.
    c. patient_key - UUID value for patient record from patient_report columnfamily. This value is returned only if the initial request included a patient_key value. Otherwise this parameter will not be returned.

Sample response:


```
{
    "AuthHeader": {
        "AuthKey": "00000000-aaaa-2222-bbbb-34567890cdef",
        "Auth_UName": "apiuser",
        "Auth_UPass": "d231bdd3e009690bbafac9146788be31",
        "Auth_UToken": "1279bdd0-d827-11e4-8d1a-c35bb6b29756"
    },
    "error_code": "",
    "error_message": "",
    "method": "audit_list",
    "response": {
        "audit_info": [
            {
                "audit_data": {
                    "message": "Added account dr002emracct",
                    "new_value": "dr002emracct"
                },
                "timestamp": "2015-03-10T05:27:33.718Z"
            }
        ],
        "user_key": "278cbcb0-c6e6-11e4-8246-4f0ac8a69cff"
    }
}
```

END OF DOCUMENT