Oracle® Health Sciences Mobile Clinical Research Associate Server

Application Programming Interface Guide

Release 1.3

E47797-02

March 2014

This document contains information on public Application Programming Interfaces (APIs) that can be used to interface with Oracle Health Sciences Mobile Clinical Research Associate Server (Mobile CRA).

It contains the following topics:

- Section 1, "Using Application Programming Interfaces"
- Section 2, "RESTful Application Programming Interfaces"

1 Using Application Programming Interfaces

This section describes how to use open Site-at-a-Glance (SAAG) Service Representational State Transfer (REST) APIs. It contains the following topics:

- Section 1.1, "About Application Programming Interfaces"
- Section 1.2, "Calling APIs from Outside the Mobile Clinical Research Associate Server"
- Section 1.3, "Security Setup Required"

1.1 About Application Programming Interfaces

Mobile CRA includes a set of APIs that lets you do most things through the user interface, including creating, modifying, and installing objects.

You can call Oracle Mobile CRA APIs from source code in a defined program in Mobile CRA. In this case, no additional security or setup is required.

You can let people in your company perform actions on Mobile CRA objects from an external system.

1.2 Calling APIs from Outside the Mobile Clinical Research Associate Server

There are two kinds of APIs created in the system, that is, admin APIs and non-admin APIs. Both have different authentication mechanisms. The admin APIs can be authenticated using the basic authentication and non-admin APIs can be called by a token mechanism. For more information, see Section 1.3.

1.3 Security Setup Required

There is a Token jar file available with the package. You can use this to generate the token.

Tokens are available at the following levels:

1. Client level - The company using this installation. You can create multiple clients (companies), however, this is not necessary in most cases.
2. Source level - Lets you create separate keys for separate source of information. Once the keys are generated, you can create tokens and distribute them to the sources.
3. User level - Lets you build the site-at-a-glance KPIs, subscriptions, and alerts at the application level.

You can generate tokens using the ID and key, based on the required level and details mentioned in the API. For example, if you want to generate a client token, you need the client ID and client key.

The following is an example on how to generate a token:

```
TokenService service =

TokenServiceFactory.getInstance().withServiceUrl("http://<hostname>:<port>
").getTokenService();

String tokenString = service.createToken(<Id>, <Key>, <userId>, <Level>);
```

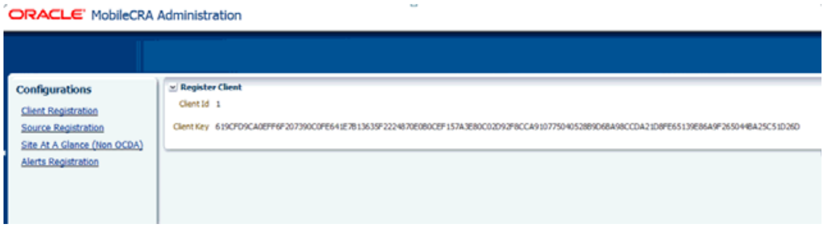
`tokenString` returns the created Token to the caller.

Where, `hostname` and `port` is the configuration where Adaptive SAAG server is hosted.

`userID` is applicable only for generating token at the user level. You must set this to null for all cases.

`Key` is the value generated when the client or source is established. You can view the value in the Mobile CRA Administration Configuration screen.

Figure 1 Client Registration



Client Id 1

Client Key 619CFD9CA0EFF6F207390C0FE641E7B13635F2224870E0B0CEF157A3E80C02D92F8CCA9107750405289D6BA98CCDA21D8FE65139E86A9F265044BA25C51D26D

2 RESTful Application Programming Interfaces

REST is a software architecture for distributed hypermedia systems such as the World Wide Web. An API is described as RESTful when it conforms to the tenets of REST.

This section contains the following topics:

- Section 2.1, "Administration Application Programming Interfaces"
- Section 2.2, "Non-Administration Application Programming Interfaces"

2.1 Administration Application Programming Interfaces

This section contains the following topic:

- Section 2.1.1, "Creating the Client"

2.1.1 Creating the Client

This API is used to create a new client. You cannot create the client in the Administration User Interface (UI). After the APIs are called, you can view the key assigned to the client in the Administration UI. Only one client must be created.You must create the client before creating sources.

URL

http://server:port/{app_root_context}/v1/clients

Request Type

HTTP POST

Request Header Parameters

Username: Adaptive SAAG Admin user name.

Password: Adaptive SAAG Admin password.

Request Parameters

clientName: Name of the client to be created.

clientDesc (optional): Short description.

numKpiToStore: Number of KPIs that are used to display in the trending graph. By default, the value is 6. This value is the number of bars shown in the SAAG UI, that is, the three trending graphs shown in the site selection slide. You cannot change this value once it is set.

Response Parameters

clientId: ID of the client created.

clientKey: Used to obtain tokens for authentication, which can be viewed in the Administration UI.

Error Code: Indicates success or failure.

2.2 Non-Administration Application Programming Interfaces

This section contains the following topics:

- Section 2.2.1, "Source"
- Section 2.2.2, "KPI Definition"
- Section 2.2.3, "KPI Data"
- Section 2.2.4, "User"
- Section 2.2.5, "Alert Definition"
- Section 2.2.6, "Alert Subscription"
- Section 2.2.7, "Alert Message"
- Section 2.2.8, "Alert Message Data"

2.2.1 Source

This section contains the following APIs:

- Section 2.2.1.1, "Create Source"
- Section 2.2.1.2, "List All Sources"

2.2.1.1 Create Source

This API is used to create a source for a client. You can create the source in the Administration UI. One source must be created for each external data source. Sources are subordinate to a client, which requires a client ID and client token when calling the API.

URL

http://server:port/{app_root_context}/v1/sources

Request Type

HTTP POST

Request Header Parameters

Type: Access Type (Client)

Token: Generated Token

ID: ClientId

Request Parameters

sourceName: Name of the source to be created.

sourceDesc (optional): Short description.

sourceType: Indicates a KPI source or alert source.

Response Parameters

Source ID: ID of the source created. This should be used for registering and populating the KPI.

sourceKey: Used for authentication for future REST APIs.

Error Code: Indicates success or failure.

2.2.1.2 List All Sources

This API is used to list all sources for a Client.

URL

http://server:port/{app_root_context}/v1/sources

Request Type HTTP GET

Request Header Parameters Type: Access Type (Client)

Token: Generated Token

ID: ClientId

Response Parameters

List of sources: List of Sources registered for a client (ID, name, description, and type).

Error Code: Indicates success or failure.

2.2.2 KPI Definition

This section contains the following APIs. You can create KPIs manually in the Administration UI. These APIs lets you create scripts and replicate setup in different instances (that is, test, QC, validation, and production).

> **Note:** The order of the KPIs in the KPI Administration screen impacts the display of the KPIs items. You cannot change the order through API.

- Section 2.2.2.1, "Create KPI Definition"
- Section 2.2.2.2, "List All KPI Definition"
- Section 2.2.2.3, "Update KPI Definition"
- Section 2.2.2.4, "Delete KPI Definition"

2.2.2.1 Create KPI Definition This API is used to create a new KPI Definition for a particular source.

URL

http://server:port/{app_root_context}/v1/kpidefs

Request Type

HTTP POST

Request Header Parameters

Type: Access Type (Client)

Token: Generated Token

ID: ClientId

Request Parameters

sourceId: The source for which the KPI needs to be created.

kpiName: KPI Name.

kpiScope: Scope of KPI (generic, study, or study-site).

kpiDesc (optional): Description of the KPI.

kpiDataType (optional): Data type of the KPI (Integer or Float). By default, the value is Float.

Response Parameters

kpiId: ID of the KPI definition created. It is used to populate the KPI data.

kpiName: KPI name.

Error Code: Indicates success or failure.

2.2.2.2 List All KPI Definition

This API is used to register all KPIs for a client.

URL

http://server:port/{app_root_context}/v1/kpidefs?sourceType=KPI

Request Type

HTTP GET

Request Header Parameters

Type: Access type (Client)

Token: Generated token

ID: ClientId

Request Parameters

Client ID: The ID of the client for listing the KPI definition.

Response Parameters

List of KPI: List of KPI registered for the client (ID, name, description, or data type).

Error Code: Indicates success or failure.

2.2.2.3 Update KPI Definition

This API is used to update an existing KPI definition for a Client.

URL http://server:port/{app_root_context}/v1/kpidefs/{kpiid}

Request Type

HTTP PUT

Request Header Parameters

Type: Access type (Client)

Token: Generated token

ID: ClientId

Request Parameters

kpiName: KPI Name.

kpiDesc: Description of the KPI.

kpiDataType: Data type of the KPI.

kpiScope: Scope of KPI (generic, study, or study-site).

Response Parameters

Error Code: Indicates success or failure.

2.2.2.4 Delete KPI Definition

This API is used to delete an existing KPI definition for a client.

URL

http://server:port/{app_root_context}/v1/kpidefs/{kpiid}

Request Type

HTTP DELETE

Request Header Parameters

Type: Access type (Client)

Token: Generated token

ID: ClientId

Response Parameters

Error Code: Indicates success or failure.

2.2.3 KPI Data

This section contains the following APIs. These APIs lets you submit data to display in SAAG and retrieve information for the display.

- Section 2.2.3.1, "Post KPI Data"
- Section 2.2.3.2, "Get SAAG KPI Data"
- Section 2.2.3.3, "Get Trending Graph KPI Data"

2.2.3.1 Post KPI Data

This API is used to post KPI data. Data can be submitted at the client or source level (which does not impact the display), and at the study or study-site level. Study KPIs display the same data regardless of the study-site selected. You must submit a token that matches the Access type and the ID. That is, if the source is equal to the client, token must be generated using the client, and the ID must be equal to the client ID.

URL

http://server:port/{app_root_context}//v1/kpidata

Request Type

HTTP POST

Request Header Parameters

Type: Access type (Source or Client)

Token: Generated token

ID: Client ID or Source ID

Request Parameters

List< KPIData>: List of KPI data having the following fields.

kpiInfoId: KPI ID

studyId: Required in case of study or study-site KPI.

kpiValue: KPI value (float or integer).

kpiName: Required along with source ID in case the KPI ID is not entered.

sourceId: Required in case of client only.

Response Parameters

Error Code: Indicates Success or Failure.

2.2.3.2 Get SAAG KPI Data

This API is used to get the SAAG KPI data.

URL

http:// server:port /{app_root_context}/v1/kpidata/saag?studyId="<Study_ID>"&siteId="<Site_id>"

Request Type

HTTP GET

Request Header Parameters

Type: Access type (Client)

Token: Generated token

ID: Client ID

Request Parameters

studyId: Study ID

siteId: Site ID

Response Parameters

Error Code: Indicates success or failure.

studyData

```
{
"creationDate":,
"kpiDataType":,
"kpiId":,
"kpiName": "",
"kpiValue": "",
"kpiDisplayName":,
"kpiSeqNo":
}
```

genericData

```
{
    "creationDate":,
     "kpiDataType":,
    "kpiId":,
      "kpiName": "",
     "kpiValue": "",
      "kpiDisplayName":,
     "kpiSeqNo":
}
```

siteListData

```
{
    "siteId": "A",
     "siteData":  [{
        "creationDate":,
        "kpiDataType":,
        "kpiId":,
       "kpiName": "",
        "kpiValue": "",
        "kpiDisplayName":,
        "kpiSeqNo":
      }]
}
```

2.2.3.3 Get Trending Graph KPI Data

This API is used to get trending graph KPI data.

URL

http:// server:port /{app_root_context}//v1/kpidata/graph?studyId="<Study_ID>"&siteId="<Site_ID>"&count=1

Request Type

HTTP GET

Request Header Parameters

Type: Access Type (Client)

Token: Generated Token

ID: Client ID

Request Parameters

kpiInfoId: KPI ID

studyId: Study ID

siteId (Optional): Site ID

Count (Optional): Number of KPIs to be used for graph. By default, the value is 3.

Response Parameters

Error Code: Indicates success or failure.

studyId

siteDataList:

```
[
        {
     "studyId": "A",
      "siteId": "Site A",
      "kpiList":          []
         }]
```

kpiList:

```
[
      {
         "data":            [ ],
        "kpiId": 1,
         "kpiName": "A",
        "kpiDisplayName:"Display""
       }
 ]
```

Data:

```
{
     "creationDate": 1,
      "kpiData": "1"
 }
```

2.2.4 User

This section contains the following user level APIs. For an implementation that involves Adaptive SAAG server, the server code handles the user registration automatically during login.

- Section 2.2.4.1, "Registering the User"
- Section 2.2.4.2, "Update User"

2.2.4.1 Registering the User

This API is used to register a user.

URL

http://server:port/{app_root_context}/v1/user

Request Type

HTTP POST

Request Header Parameters

Type: Access Type (Client)

Token: Generated Token

ID: Client ID

Request Parameters

userName: Name of the user to be created.

deviceId: Device ID of the device on which user has logged.

registrationToken: Registration token from Health Sciences Push Notification Service (HSPNS).

Response Parameters

userId: ID of registered user.

Error Code: Indicates success or failure.

2.2.4.2 Update User

This API is used to update user details.

URL

http://server:port/{app_root_context}/v1/user/{userId}

Request Type

HTTP PUT

Request Header Parameters

Type: Access type (Client)

Token: Generated token

ID: Client ID

Request Parameters

deviceId: Device ID of the device on which user has logged.

registrationToken: Device ID of the device on which user has logged.

registrationToken: Registration token from HSPNS.

Response Parameters

Error Code: Indicates success or failure.

2.2.5 Alert Definition

This section contains the following APIs. Alerts can be manually created in the Administration UI. These APIs lets you create scripts and replicate setup in different instances (that is, test, QC, validation, and production).

Note: These alerts on the UI is displayed in a separate section called the 3rd Party Alerts.

- Section 2.2.5.1, "Create Alert Definition"
- Section 2.2.5.2, "List All Alert Definition"
- Section 2.2.5.3, "Update Alert Definition"
- Section 2.2.5.4, "Delete Alert Definition"

2.2.5.1 Create Alert Definition

This API is used to create a new alert definition for a particular source.

URL

http://server:port/{app_root_context}/v1/alertdefs

Request Type

HTTP POST

Request Header Parameters

Type: Access type (Client)

Token: Generated token

ID: Client ID

Request Parameters

sourceId: The source for which the alert needs to be created.

alertName: Alert name.

alertDesc (optional): Description of the alert.

alertDisplayName: Display name of the alert.

Response Parameters

alertId: ID of the alert definition that is used to populate the alert data.

Error Code: Indicates success or failure.

2.2.5.2 List All Alert Definition

This API is used to get all the alerts registered for a client.

URL

http://server:port/{app_root_context}/v1/alertdefs

Request Type

HTTP POST

Request Header Parameters

Type: Access type (User or Client)

Token: Generated token

ID: User ID or Client ID

Request Parameters

Client ID: ID of the client for listing alert definitions.

Response Parameters

List of Alerts: List of alerts registered for the client (ID, source ID, name, description, displayName, and isSubscribed).

Error Code: Indicates success or failure.

2.2.5.3 Update Alert Definition

This API is used to update an existing alert definition.

URL

http://server:port/{app_root_context}/v1/alertdefs/{alertId}

Request Type

HTTP PUT

Request Header Parameters

Type: Access type (Source)

Token: Generated token

ID: Source ID

Request Parameters

alertName: Name of the KPI.

alertDesc (optional): Description of the alert.

alertDisplayName: Display name of the alert.

Response Parameters

Error Code: Indicates success or failure.

2.2.5.4 Delete Alert Definition

This API is used to delete an existing alert definition for a Client.

URL

http://server:port/{app_root_context}/v1/alertdefs/{alertId}

Request Type

HTTP DELETE

Request Header Parameters

Type: Access type (Source)

Token: Generated token

ID: Source ID

Response Parameters

Error Code: Indicates success or failure.

2.2.6 Alert Subscription

When you subscribe to 3rd party alerts, the system automatically registers you for such alerts. The following APIs are involved during the subscription process that uses the user ID.

- Section 2.2.6.1, "Subscribe Alert"
- Section 2.2.6.2, "Unsubscribe Alert"

2.2.6.1 Subscribe Alert

This API is used to subscribe an alert for a user.

URL

http://server:port/{app_root_context}/v1/alertsub

Request Type

HTTP POST

Request Header Parameters

Type: Access type

(User) Token: Generated token

ID: User ID

Request Parameters

alertSubId: The alert subscription ID for which the alert is subscribed.

Response Parameters

SubscriptionId: Subscription ID of subscribed alert.

Error Code: Indicates success or failure.

2.2.6.2 Unsubscribe Alert

This API is used to unsubscribe an alert for a user.

URL

http://server:port/{app_root_context}/v1/alertsub/{alertSubId}

Request Type

HTTP DELETE

Request Header Parameters

Type: Access type

(User) Token: Generated token

ID: User ID

Response Parameters

Error Code: Indicates success or failure.

2.2.7 Alert Message

This section contains the following APIs. The post alert message API sends alert to the Alerts repository.

- Section 2.2.7.1, "Post Alert Message"
- Section 2.2.7.2, "List All Alert Message"

2.2.7.1 Post Alert Message

This API is used to post alert messages.

URL

http://server:port/{app_root_context}//v1/notifications

Request Type

HTTP POST

Request Header Parameters

Type: Access type (Source or Client)

Token: Generated token

ID: Client ID or Source ID

Request Parameters

List< AlertMessage> contains: List of alert data having the following fields:

alertId: Alert ID.

alertName: Required in case alert ID is not present.

sourceId: Required in case of client only.

Message: Alert message.

Response Parameters

Error Code: Indicates success or failure.

2.2.7.2 List All Alert Message

This API is used to get all the alert messages for a user.

URL

http://server:port/{app_root_context}/v1/alertmessage

Request Type

HTTP GET

Request Header Parameters

Type: Access type (User)

Token: Generated token

ID: User ID

Response Parameters

List of Alert messages: List of alerts registered for the client (ID, message, and readFlag).

Error Code: Indicates success or failure.

2.2.8 Alert Message Data

This section contains the following APIs:

- Section 2.2.8.1, "Get Alert Message Data"
- Section 2.2.8.2, "Update Alert Message Data"

2.2.8.1 Get Alert Message Data

This API is used to post alert messages.

URL

http://server:port/{app_root_context}//v1/notifications?startid=5&limit=10

Request Type

HTTP GET

Request Header Parameters

Type: Access type (User)

Token: Generated token

ID: User ID

Request Parameters

List< AlertMessage> contains: List of alert data having the following fields:

startId (optional): Start ID of the next batch of notification. Defaults to the first notification.

limit (optional): Number of messages to be retrieved. By default, the value is 50.

Response Parameters

List of Notifications: Notification list with the following data:

```
{
      "date": "2013-06-21T16:57:06+0530",
     "userId": null,
     "header": "Display Name",
     "notificationId": 54,
      "read": true,
      "important": true,
     "notificationText": "Test Message 1"
   }
```

nextPageStartId: Start ID to be used to get the next batch.

Error Code: Indicates success or failure.

2.2.8.2 Update Alert Message Data

This API is used to post alert messages.

URL

http://server:port/{app_root_context}//v1/{notificationId}

Request Type

HTTP PUT

Request Header Parameters

Type: Access type (User)

Token: Generated token

ID: User ID

Request Parameters

read: Flag to mark as read.

important: Flag to mark important (either read or important needs to be present as part of request).

limit (optional): Number of messages to be retrieved. By default, the value is 50.

Response Parameters

Error Code: Indicates success or failure.

3 Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Oracle® Health Sciences Mobile Clinical Research Associate Server Application Programming Interface Guide, Release 1.3
E47797-02