**Medtronic MITG**

**Research & Development**

**Detail Design for**

**Gateway Device Management Platform 4.0**

**RE00062904 Rev A**

|  |  |  |  |
| --- | --- | --- | --- |
| Issue Date: 04/14/2017 | | | Page 1 of 43 |
|  | | | |
| Revision History: | | | |
| Revision | SDA | Description | |
| A | RC090544 | Initial Release | |
|  |  | | |
| Author: Refer to Agile for approvals | | | |
| Xishi Pan – Insigma US (development partner) | | | |
|  | | | |
| **Approvals:** Refer to Agile for approvals | | | |
| Cliff Schorr – QA- PMR | | | |
| *Paul Ourada – MITG R&D* | | | |

Contents

[1. INTRODUCTION 4](#_Toc479951155)

[1.1 Purpose 4](#_Toc479951156)

[1.2 Scope 4](#_Toc479951157)

[2. REFERENCED DOCUMENTS 4](#_Toc479951158)

[3. DEFINITIONS AND ACRONYMS 4](#_Toc479951159)

[4. ARCHITECTURE OVERVIEW 5](#_Toc479951160)

[4.1 High Level Architecture 5](#_Toc479951161)

[5. System Detail Design 7](#_Toc479951162)

[5.1 Application Server 7](#_Toc479951163)

[5.1.1 High Level Components Overview 7](#_Toc479951164)

[5.1.2 Server-Side Business Rule Configuration 9](#_Toc479951165)

[5.1.3 Annotation 10](#_Toc479951166)

[5.1.4 Restful API Request/Response Filter 10](#_Toc479951167)

[5.1.5 Secure Communication with Agent 11](#_Toc479951168)

[5.1.6 Feature License File Generation 11](#_Toc479951169)

[5.1.7 LDAP Design 12](#_Toc479951170)

[5.1.8 Server-Agent Restful API 15](#_Toc479951171)

[5.1.9 Server-Web Restful API 17](#_Toc479951172)

[5.1.10 Reporting 18](#_Toc479951173)

[5.2 Agent 24](#_Toc479951174)

[5.2.1 Overview 24](#_Toc479951175)

[5.2.2 Communication with Application Server 24](#_Toc479951176)

[5.2.3 Common Request Handling 25](#_Toc479951177)

[5.2.4 Restful API Requests Handling 25](#_Toc479951178)

[5.2.5 File Encryption 25](#_Toc479951179)

[5.2.6 Configuration Protection 26](#_Toc479951180)

[5.2.7 H2 Database 26](#_Toc479951181)

[5.3 Web 27](#_Toc479951182)

[5.3.1 High Level Components Overview 27](#_Toc479951183)

[5.3.2 Template Structure 28](#_Toc479951184)

[5.3.3 General Table Structure 30](#_Toc479951185)

[5.4 Tools 31](#_Toc479951186)

[5.4.1 Connection Tool 31](#_Toc479951187)

[5.4.2 Remapper Tool 31](#_Toc479951188)

[5.4.3 ETL 31](#_Toc479951189)

[5.5 Appendix 32](#_Toc479951190)

[5.5.1 Agent-Server API 32](#_Toc479951191)

[5.5.2 Server APIs for Common Client 55](#_Toc479951192)

# INTRODUCTION

This chapter introduces purpose and scope of this document.

## Purpose

This document describes detail system design of the GDMP 4.0 system.

## Scope

The scope of this document is describing detail system component design of Gateway Device Management Platform (GDMP) 4.0.

# REFERENCED DOCUMENTS

| Ref # | Reference Document | Rev |
| --- | --- | --- |
| [1] | RE00062903\_A System Architecture Gateway v4.0 | A |
| [2] | R0030891\_S System Requirements Spec Gateway | S |
| [3] | RE00062811 SPEC Interface Messaging Theory Gateway 4 | A |
|  |  |  |

# DEFINITIONS AND ACRONYMS

| Term or Abbreviation | Description |
| --- | --- |
| GDMP | Gateway Device Management Platform |
| LDAP | Lightweight Directory Access Protocol |
| LB | Load Balance |
| DB | MySQL database |
| ROLE | A set of user privileges to authorized user can use. |
| PUBLIC SCHEMA | Integration point with Enterprise one for exchange data |
| MVC | Module-View-Controller design pattern |
| CoT | Class of Trade |
| SPOF | Single Point of Failure |
| SMTP | Simple Mail Transfer Protocol |
| Legacy Client | Existing client applications, such as VLEX client application |
| VLEX | Valleylab™ Exchange (VLEX) Software Update System |
| SLA | Service Level Agreement |
| API | Application Programming Interface |
| AD | Active Directory Service |
| ETL | Extract, Transform and Load (ETL) refers to a process in database usage and especially in data warehousing that: Extracts data from homogeneous or heterogeneous data sources. Transforms the data for storing it in the proper format or structure for the purposes of querying and analysis. |

# ARCHITECTURE OVERVIEW

GDMP 4.0 is designed with a multitier architecture including GDMP Front-End Tier, GDMP Core Tier and Storage Tier. GDMP Front-End Tier consists of GDMP Web and GDMP Agent, while GDMP Core Tier consists of GDMP Server.

The following sections will address these in detail.

## High Level Architecture

The current GDMP is built based on Drupal framework, along with more and more business needs, the maintenance and supporting cost of the current GDMP increases rapidly. The intent of GDMP 4.0 is to redesign internal component to reduce on-going maintenance and supporting cost, along with enhancement in security and critical system functionalities.

GDMP 4.0 will keep current application interfaces and behavior intact to all existing Legacy Client and focus on renovation of the components inside GDMP system.

GDMP 4.0 will consist of multiple components and services.

* GDMP Web

GDMP Web is a Web application providing web-based user interface to allow end users, such as CoT admin and application support, to use GDMP to conduct medical device management and configuration tasks including hardware/software configuration, report generation, device/user management, and so on.

* GDMP Agent

GDMP Agent is a Java application working as integration point between Common Clients/Legacy Clients and GDMP Server. It is mainly responsible for downloading specified software and document from GDMP Server for clients and devices, and pushing device status information and operation audit information (e.g.: log files) to the GDMP Server. GDMP Agent is designed to support both online (working with an Internet connection to GDMP Server) and offline (working without an Internet connection to GDMP Server) mode.

* GDMP Server

GDMP Server is the core business logic tier to manage hardware, software, document, named configuration, device, reports, alert, and user information. It provides RESTful interfaces to GDMP Web and GDMP Agent.

* Database Server

Inside GDMP, database will store all information except user information, PERMISSIONs and ROLEs information.

* LDAP Server

LDAP Server stores all roles & permission information and non-Medtronic users’ profiles. Besides, LDAP Server will provide such roles & permission and user profile information to GDMP Server, GDMP Server will determine how to authorize and control access based on this information.



Figure 2 High Level Architecture

The diagram above illustrates high level components of GDMP 4.0 and how components communicate with each other inside the system.

# System Detail Design

## Application Server

### High Level Components Overview

Server consists of several major components that handle various requests

* Main Restful API Controller

Server provides two main Restful API Controllers.

* com.medtronic.gdmp.server.rest.resources.web.WebResponder – Restful API Controller to handle Web application requests
* com.medtronic.gdmp.server.rest.resources.agent.AgentResponder – Restful API Controller to handle Agent application requests
* Sub Modules

This section provides a list of service components that handle various data processing requests.

* com.medtronic.gdmp.server.service.core.impl.AlertServiceImpl – Alert Service that handles alert-related data requests and alert delivery
* com.medtronic.gdmp.server.service.core.impl.AuthenticationServiceImpl – Authentication Service that authenticate login requests
* com.medtronic.gdmp.server.service.core.impl.AuthorizationServiceImpl – Authorization Service that validates authorization of an authenticate user to specific access
* com.medtronic.gdmp.server.service.core.impl.ConfigurationServiceImpl –Service to handle named configurations
* com.medtronic.gdmp.server.service.core.impl.CustomerServiceImpl – Service to handle customer data
* com.medtronic.gdmp.server.service.core.impl.DeviceServiceHistoryServiceImpl – Service to handle device service history records
* com.medtronic.gdmp.server.service.core.impl.DeviceServiceImpl – Service to handle devices
* com.medtronic.gdmp.server.service.core.impl.DeviceTypeMetaDataServiceImpl – Service to handle device type meta data requests for Class of Trade and Device Types.
* com.medtronic.gdmp.server.service.core.impl.DocumentServiceImpl – Service to handle documents
* com.medtronic.gdmp.server.service.core.impl.FeatureServiceImpl – Service to handle feature and feature entitlements, including feature license file generation
* com.medtronic.gdmp.server.service.core.impl.HardwareServiceImpl – Service to handle hardware items
* com.medtronic.gdmp.server.service.core.impl.SoftwareServiceImpl – Service to handle software items
* com.medtronic.gdmp.server.service.core.impl.LogFileServiceImpl – Service to handle device log files
* com.medtronic.gdmp.server.service.core.impl.ReportServiceImpl – Service to handle report generation requests
* com.medtronic.gdmp.server.service.core.impl.TradingEmbargoServiceImpl – Service to handle trade embargo settings
* com.medtronic.gdmp.server.service.core.impl.TrainingServiceImpl – Service to handle training records related operations
* com.medtronic.gdmp.server.service.core.impl.UserServiceImpl – Service to handle user/role related operations
* com.medtronic.gdmp.server.service.common.beans.CertificateServiceImpl – Service to handle certificates files requests
* com.medtronic.gdmp.server.service.common.beans.ClamAvScanService – Service to handle ClamAV anti-virus scanning requests
* com.medtronic.gdmp.server.service.common.beans.DeflatedEncryptedFileService – Service to handle file encryption/decryption
* com.medtronic.gdmp.server.service.common.beans.EncryptedSessionService – Service to handle session-in-cookie encryption/decryption
* com.medtronic.gdmp.server.service.common.beans.LDAPService – Service to handle all LDAP related operations
* com.medtronic.gdmp.server.service.common.beans.SmtpServiceImpl – Service to handle email delivery to SMTP server
* com.medtronic.gdmp.server.service.common.beans.ValidationServiceImpl – Service to handle various validation requests like checking duplicate software with same name/version/part#.
* com.medtronic.gdmp.server.service.audit.impl.AuditServiceImpl – Service to handle user audit trail records
* com.medtronic.gdmp.server.repos.config.BasicApplicationConfiguration – Service to provide application server configuration
* com.medtronic.gdmp.server.repos.config.BusinessRulesConfiguration – Service to provide business rule configuration at server side
* com.medtronic.gdmp.server.repos.config.ReportConfigRepository – Service to provide report configuration meta data
* com.medtronic.gdmp.server.repos.component.TemporaryFileUploadRepository – Service to handle files that are temporarily uploaded to application server

### Server-Side Business Rule Configuration

There are a couple of business rules configured in business-rules.json to provide rules to assist server to handle various requests.

* business\_rules.started\_but\_incomplete\_upgrade\_waiting\_period – Max waiting period in minutes used to determine started software upgrade but not completed for alert generation. For example, 2880 means if a software upgrade started but not completed in 48 hours, an alert will be generated.
* business\_rules.software\_retention\_period – Software expiry setting per device type, in days.
* business\_rules.document\_retention\_period – Document expiry setting per device type, in days.
* business\_rules.status\_transit – A map for an item status transition by item type. For example, for software in status of ‘In Production` - 1, next available status could be ‘Limited Release` - 2 and `Archived` -3, but ‘Unknown` - 0 is not allowed.
* business\_rules.display\_hardware\_component\_serial\_number – Setting per Class of trade to toggle display of hardware component SN on Web
* business\_rules.software\_version\_match\_criteria – Settings per Device Type on software component matching criteria. Name matching is required so for each device type, only needs to set true/false to “version” and “partnumber”.
* business\_rules.hardware\_version\_match\_criteria - Settings per Device Type on hardware component matching criteria.
* business\_rules.device\_country\_change – Settings per Device Type on the rules to set device country code.

The structure of the configuration is organized in the following –

<create new device – true/false>

<user type – 0 – MDT User / 1 – Non-MDT User>

<device type name> : [ Sources of Info]

[ Sources of Info] is an array that could contains none or multiple of these values –

* + CLIENT – from client application (device request)
  + USER – from current user

As long as a source has provided the information, following source will not be used.

For example, [CLIENT, USER] means if client provides country code, use it as priority. If not, check country code of user’s profile. If still not available, no change.

If [Sources of Info] is empty, it means no change.

* business\_rules.device\_customer\_change – Settings per Device Type on the rules to set device customer

The structure of the configuration is organized in the following –

<create new device – true/false>

<user type – 0 – MDT User / 1 – Non-MDT User>

<device type name> : [ Sources of Info]

[ Sources of Info] is an array that could contains none or multiple of these values –

* + CLIENT – from client application (device request)
  + USER – from current user
  + UNKNOWN – Unknown customer

As long as a source has provided the information, following source will not be used.

For example, [CLIENT, USER] means if client provides country code, use it as priority. If not, check country code of user’s profile. If still not available, no change. UNKNOWN means use customer named ‘Unknown’

If [Sources of Info] is empty, it means no change.

* business\_rules.device\_log\_handling – Settings per device type and device log type name to allow direct download/view on web application or not

Configuration is organized in the following structure –

<device type name>

<device log type name>

By default, device log files with the following extensions will be allowed to view/download on web

* + .html.gz
  + .xml.gz
  + .html
  + .log
  + .xml
  + .txt
* business\_rules.device\_log\_type\_mime – Setting on MIME type per device log file extension name
* business\_rules.device\_log\_sqlite\_query – Settings of SQL statement to retrieve and display SQLite-based device log file. Configured by device type and device log type name.
* business\_rules.registration – Settings on how COT/Device Types should display on self-registration form
* business\_rules.machine\_accounts – Define a list of machine accounts that need special handling
* business\_rules.catalog\_device\_types\_cot – Settings per Class Of Trade to specify if it is Catalog device types or not. Default is false if not configured
* business\_rules.check\_device\_in\_account – Define device type name that should trigger alert – “Device not in User in Account” when performing a software upgrade. Default is false. Only PB980 is configured.
* business\_rules.software\_upgrade\_update\_component – By default, when a Software Upgrade is successfully completed, server should update device’s software component info immediately while it makes exception for VLEX client reported devices. Default is true, which is for common existing devices and new device types.

### Annotation

A couple of annotations are created to control shared workflow.

* com.medtronic.gdmp.server.annotation.Authorization – any Restful API function that annotated should require authentication
* com.medtronic.gdmp.server.annotation.DoubleEncryptedCommunication – Used to control double encryption communication for communication between Agent/Server.

### Restful API Request/Response Filter

There are several API resource filters, inspecting request headers or updating information in response header.

* com.medtronic.gdmp.server.rest.filter.request.TokenAuthenticationRequestFilter – Use to authenticate Agent Restful API requests
* com.medtronic.gdmp.server.rest.filter.request.SessionInCookieAuthenticationRequestFilter – Used to authenticate Web Restful API requests
* com.medtronic.gdmp.server.rest.filter.response.SessionInCookieResponseFilter – Used to update session-in-cookie before a response is sent to Web client
* com.medtronic.gdmp.server.rest.filter.response.CommonResponseFilter – Set expiry time to -1 so don’t let web cache any data. For Web only
* com.medtronic.gdmp.server.rest.filter.response.CorsFilter – Set CORS headers in API responses. For Web only

Inside com.medtronic.gdmp.server.rest.filter.FiltersDynamicBinding, it controls how such filters are registered to Agent API responder or Web API responder.

### Secure Communication with Agent

Communication with Agent will have double-encryption enabled most of the time.

Annotation com.medtronic.gdmp.server.annotation.DoubleEncryptedCommunication controls encryption/decryption should be enabled when server read requests from Agent and/or send response to Agent. By default, it is enabled for both request receiving and response sending.

Server has implemented Jersey framework’s ReaderInterceptor and WriterInterceptor to decrypt Agent request and encrypt response to Agent.

* com.medtronic.gdmp.server.rest.interceptor.AESDecryptionReader – Implement ReaderInterceptor and decrypt Agent request body
* com.medtronic.gdmp.server.rest.interceptor.AESEncryptionWriter – Implement WriterInterceptor and encrypt response to Agent

Encrypted response is encoded in Base64 as a single line data before sending to Agent while Agent will also send Base64-encoded encrypted request to server.

The encryption algorithm and key generation algorithm is same to the ones used to encrypt files. By default, it is AES/CFB8/NOPADDING and key generation algorithm is MD5 that hashes a plain text passcode.

### Feature License File Generation

DRM Tool installed at /opt/medtronic/gdmp4-server/bin is used to generate feature license file.

The following work flow is followed when creating/updating feature license file for a device – identified by device type id and serial number:

* Pre-condition
  + Feature item has to be created with proper feature identification, which is a number predefined to identify a unique feature
* When user create new feature entitlements or update existing feature entitlements for a device
  + Collect all current feature entitlements for the target device in status of ‘In Production’ and ‘Limited Release’
  + Create temporary XML file to list all feature entries with entitlement info
  + Call DRM tool to generate feature license file
  + Encrypt and Compress feature license file
  + Update feature\_license table with file id and associated feature entitlements id

A sample of XML file used by DRM tool to create feature license file can be found in the DRM tool bundle.

### LDAP Design

#### Organization Structure

LDAP structures are organized as below –

| DN | Description |
| --- | --- |
| dc=medtronic,dc=com | base dn |
| o=gdmp,dc=medtronic,dc=com | organization |
| ou=users,o=gdmp,dc=medtronic,dc=com | organization unit, under this ou, 3 types of user will be stored |
| ou=customerUsers,ou=users,o=gdmp,dc=medtronic,dc=com | organization unit, all the customer users will be stored here |
| ou=medronicUsers,ou=users,o=gdmp,dc=medtronic,dc=com | organization unit, all the medronic users will be stored here |
| ou=built-inUsers,ou=users,o=gdmp,dc=medtronic,dc=com | organization unit, store some special users used by system self or administrator   1. admin is a default super administrator account belong to role "uiAdmin" who can access all the pages from web, but cannot login from client 2. proxyAdmin is a gdmp system using account to operate "Add, Modify, Delete" from gdmp server |
| ou=roles,o=gdmp,dc=medtronic,dc=com | organization unit, the role named with COT and role information will be stored here, role is of gdmpRole objectclass which is created by ourselves |
| ou=userRelations,o=gdmp,dc=medtronic,dc=com | organization unit, this ou is parent ou to store the relationship between user and role |
| ou=userInRoles,ou=userRelations,o=gdmp,dc=medtronic,dc=com | organization unit, this ou stores some "memberOfgroup" in which the user member will act as specific role. |
| ou=userInDefaultRoles,ou=userRelations,o=gdmp,dc=medtronic,dc=com | organization unit, this ou stores some "memberOfgroup" in which the user member will act as a default role. |
| ou=groupRelations,o=gdmp,dc=medtronic,dc=com | organization unit, this ou stores some "memberOfgroup" in which the user member will own the right to approve registration. |

#### Permission Control

In each role, a set of permission object controls user’s web and device access. These access objects are defined as below –

| Name |  | Type | Description |
| --- | --- | --- | --- |
| gdmpAlertPageEditable |  | Auxiliary | the permission with which user can edit the alert setting from web |
| gdmpAlertPageViewOnly |  | Auxiliary | the permission with which user only can view the alert page |
| gdmpConfigPageEditable |  | Auxiliary | the permission with which user can edit the named configuration setting from web |
| gdmpConfigPageViewOnly |  | Auxiliary | the permission with which user only can view the named configuration page |
| gdmpDevicePageAccessible |  | Auxiliary | the permission with which user can access the page about device setting from web |
| gdmpDocPageEditable |  | Auxiliary | the permission with which user can edit the document setting from web |
| gdmpDocPageViewOnly |  | Auxiliary | the permission with which user only can view the document page |
| gdmpFeaturePageEditable |  | Auxiliary | the permission with which user can edit the feature license setting from web |
| gdmpFeaturePageViewOnly |  | Auxiliary | the permission with which user only can view the feature license page |
| gdmpHardwarePageEditable |  | Auxiliary | the permission with which user can edit the hardware setting from web |
| gdmpHardwarePageViewOnly |  | Auxiliary | the permission with which user only can view the hardware page |
| gdmpReportPageAccessible |  | Auxiliary | the permission with which user can access the report page from web |
| gdmpSoftwarePageEdiable |  | Auxiliary | the permission with which user can edit the software setting from web |
| gdmpSoftwarePageViewOnly |  | Auxiliary | the permission with which user only can view the software page |
| gdmpUserPageEditable |  | Auxiliary | the permission with which user can edit the user setting from web |
| gdmpUserPageViewOnly |  | Auxiliary | the permission with which user only can view the user page |
| gdmpLimitedReleaseSoftwareAccessible |  | Auxiliary | User can access limited release software |
| gdmpSoftwareInstallable |  | Auxiliary | User can install software |
| gdmpLatestVersionSoftwareAccessible |  | Auxiliary | User can access latest version software only |
| gdmpFeatureLicenseInstallable |  | Auxiliary | User can install feature license |
| gdmpLimitedReleaseFeatureAccessible |  | Auxiliary | User can access limited release feature |

#### LDAP Data Entry Process

To process LDAP data entries, a couple of modules will manage data ACUD –

* com.medtronic.gdmp.directoryservice.implement.OpenDJCRUDUtil – Basic LDAP entry ACUD operations
* com.medtronic.gdmp.directoryservice.implement.UserServiceOpenDJImpl – User-related operations
* com.medtronic.gdmp.directoryservice.implement.RoleServiceOpenDJImpl – Role-related operations
* com.medtronic.gdmp.directoryservice.implement.GroupServiceOpenDJImpl – General group membership related operations

For different types of entities in LDAP, corresponding data entities in Java class are defined in package com.medtronic.gdmp.directoryservice.entities.

* com.medtronic.gdmp.directoryservice.entities.LDAPUser – Main User Entry
* com.medtronic.gdmp.directoryservice.entities.LDAPRole – Main Role Entry
* com.medtronic.gdmp.directoryservice.entities.Group – Abstract Group Entry, which is extended by several groups that stores user’s group membership to store information like Default Trainer
  + com.medtronic.gdmp.directoryservice.entities.CustomerUserApprovingProxy – Customer User Approving Proxy per Device Type
  + com.medtronic.gdmp.directoryservice.entities.DefaultCustomerUserApprovingProxy – default Customer User Approving Proxy per Device Type
  + com.medtronic.gdmp.directoryservice.entities.DefaultTrainerGroup – default trainer group per Device Type
  + com.medtronic.gdmp.directoryservice.entities.TrainerGroup – trainer group per Device Type
  + com.medtronic.gdmp.directoryservice.entities.DefaultMedtronicUserApprovingManagers – default MDT user approving manager per Device Type

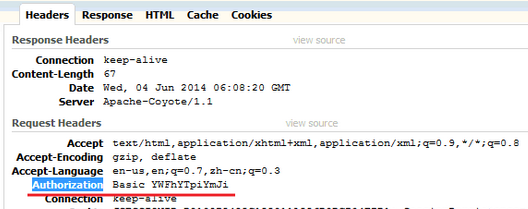
### Server-Agent Restful API

This section provides detailed Restful API definition between Server and Agent.

#### Authentication

Each request will contain user email address as user unique ID and a MD5 hash of password.

The authentication information will be placed in field Authorization of HTTP header. For example:



You can see that the user ID and password are base64 encoded. After decoded, it should be in following format, user\_email:password\_hash. Response with HTTP status code 401 if authentication failed.

#### Session

Since concept of session in 3.x between Server and Agent has been removed, and the 4.0 system uses totally stateless RESTful service, there will be no session related API.

#### Error result

When an API failed, server will indicate the failure in HTTP status code. And a JSON string will return. For example:

{

"code": 400,

"msg": "timestamp is missing",

"detail": "no timestamp"

}

#### Common Parameters

Each request will contain a time stamp (milliseconds) for recording when user sends the request. (As Agent support the offline mode, the time stamp will be the time Agent received the request, not the time Agent sent it to Server), the timestamp will be appended at the end of the endpoint URL.

Agent session GUID shall be included when available too with parameter “session”.

A reference number for Agent internal logging purpose is passed over too with parameter ‘ref’. This is optional.

Take Login for example:

/medtronic/rest/agent/api/users/login?**timestamp**=1120330201123&**session**=f0678c2d-be95-4539-8c9a-98197bb73846&**ref**=13860556626992

If no timestamp is attached, Server responses with 400 Bad Request.

### Server-Web Restful API

##### Session-in-Cookie Authentication

To authenticate web API requests, application server requires a cookie to be set along with requests that require authentication. The content of cookie is base64-encoded encrypted user session info.

Session is defined at com.medtronic.gdmp.server.domain.rest.common.Session which stores the following info

* Brief user info like user id, UUID
* User login time
* Last API request time

This session object is serialized into JSON string, encrypted by AES/CBC/PKCS5Padding using SHA-1 as key generation algorithm and then base64 encoded.

When Web application call /web/api/login and user is successfully authenticated, Application Server will generate this data and return to web as a cookie. /web/api/logoff request will remove this cookie.

It is important not to add too much info in the session object to prevent generating large JSON/encrypted base64 string as size of all cookies cannot exceed 4093 bytes (cookie overhead counted).

##### General Data Access API

The pattern of API is always in the following format:

/web/api/<name>

Accordingly, to access devices, endpoint is /web/api/devices. To access software, endpoint is /web/api/software

| Method | Parameters | Purpose | Comment |
| --- | --- | --- | --- |
| GET | **Required Parameters:**  cot=<name> - Specify id of COT  **Optional Parameters:**  limit=<number> - Specify limit number of records to return  offset=<number> - Specify offset of records to start retrieving  include\_filter=<true or false> - Specify if server should return available filters  <filter\_name>=<filter\_values> - Specify filters  sortby=<fieldname> - Specify the field to sort in response records | Query records | See section for filters |
| POST | API endpoint: /web/api/<name>/<id>    Body: TBD | Create record |  |
| PUT | /web/api/<name>/<id> | Update record |  |
| DELETE | /web/api/<name>/<id> | Delete record | Not actually delete but set delete time of a record |

### Reporting

#### Basic Reporting Model

Three basic elements compose reporting model -

* Report
* ReportBuilder
* ReportDataModel

##### Report

Report is the the core container of a report including 3 basic report elements:

* Report Config
* Report Criteria
* Report Data

##### Report Config

Report Config configure each report including report name, report columns and so on:

##### Report Name

Reports of same type share a same name (for now). Macro replacement-based dynamic report name generation is considered later.

##### Report Columns

Report column is a configuration of a column in report. One report could consist of multiple columns - which is also known as report fields. Report column contains the following information, which is re-loadable from file.

* Column (Header) Name - like User Name or Customer Name or Country Name
* Column Width - width of column in the report. May be not used, depending on report format. PDF/Excel may use this value while it is not applicable to CSV
* Column Type - String, Numeric or Date
* Column Value Index - to fill in cell of column, an index of the ReportElement values array is required
* Column Format String - to format specific column data type. For example, if column value is a DATE/TIME/DATETIME value, format field specifies the format string to initialize SimpleDateFormat. See "Activity Time" in example below. Optional.
* Column Comment - an optional string that will be used to create a comment in the column header

Such configuration can be loaded from a JSON file like:

 {  
 "name": "Software Audit By CoT Report",  
 "headers" : [  
 {  
 "name" : "CoT",  
 "type" : "STRING",  
 "index" : 0  
 },  
 {  
 "name" : "Total # of Devices",  
 "type" : "NUMERIC",  
 "index" : 1  
 }  
 ],  
 "columns": [  
 {  
 "name": "Device Type",  
 "width": 5000,  
 "type": "STRING",  
 "index": 0  
 },  
 {  
 "name": "Country",  
 "width": 5000,  
 "type": "STRING",  
 "index": 1  
 },  
 {  
 "name" : "Customer Name",  
 "width" : 5000,  
 "type" : "STRING",  
 "index" : 2  
 },  
 {  
 "name" : "Software Version",  
 "width" : 5000,  
 "type" : "STRING",  
 "index" : 3  
 },  
 {  
 "name" : "# Of Device",  
 "width" : 3000,  
 "type" : "NUMERIC",  
 "index" : 4  
 },  
 {  
 "name" : "% Country",  
 "width" : 3000,  
 "type" : "NUMERIC",  
 "index" : 5  
 },  
 {  
 "name" : "% Overall",  
 "width" : 3000,  
 "type" : "NUMERIC",  
 "index" : 6  
 }  
 ]  
}

 Report configuration can be found gdmp-server\src\main\resources\reports

##### Report Criteria

Report Criteria contains various filters while not all filters might be used for a specific report type.

Abstract ReportCriteria contains most of the common criteria (e.g.: CoT Name List, Device Type List, ...) and specific report type should define their own Report Criteria that extends ReportCriteria and implement method

isMandatoryFilterSet() which checks if mandatory filter is set for this report.

RowBounds can be used to provide information for pagination.

Leverage MyBatis Dynamic SQL (choose, when, otherwise) for using a single SQL statement mapping with multiple nullable parameters in Criteria. No need to define multiple SQL select according to existence of parameters.

##### Report Data

Report Data holds data of a report. Two types of data is stored, in separate container - which is a List<? extends ReportElement>: one is report header data and the other is report body data.

ReportElement is an interface that report element of each report has to implement. Every single report element should report an array of Serializable [] that holds values of a single row in report.

If values of ReportElement for a report type can be simply mapped to its Domain Object / Table columns, it is easier to implement ReportElement interface by the Domain Object.

 Report Column configuration tells how to pick up value from this array and fill in actual report.

#### ReportDataModel

ReportDataModel implements how data is retrieved for each type of report. Each report type usually have to implement their own ReportDataModel as queries are almost not-shareable.

By using the ReportCriteria, ReportDataModel would load data from one or more DAO and conduct necessary pre-processing.

In returned ReportData instance, ReportElements should be ready to be read by ReportBuilder to generate an actual report.

##### Annotation "@ReportConfiguration" on ReportDataModel

Each concret implementation of ReportDataModel must have annotation "@ReportConfiguration" to indicate JSON setting file location for the report configuration.

**com.medtronic.gdmp.server.repos.ReportConfigRepository** will scan and discover report configuration automatically at server startup. And they are reloadable during runtime.

##### Annotation "@ReportModel" on ReportDataModel

Each concret implementation of ReportDataModel must have annotation "@ReportModel" to indicate report type code this Model can handle.

**com.medtronic.gdmp.server.repos.ReportModelRepository** will scan and discover report data models automatically at server startup.

#### ReportBuilder

ReportBuilder defines interface a builder of report in supported formats like CSV, Excel or PDF.

Each report format has to implement this ReportBuilder interface that leverage Report object to generate report.

#### Data for Web UI

Report can be serialized as JSON string so web UI can process and display. Call AbstractReportDataModel.toJSONString(report).

RowBound object in ReportCriteria can be used to paginate on web page.

#### Report Download

Report Name is configurable before server startup and actual file name will replace all spaces with underscore "\_".

#### Integration with RESTful API

To respond reporting request, API client has to provide appropriate report filter conditions in well-structured JSON string so Server can map it to ReportCriteria object.

API client has to tell Server the target report. Otherwise, server may ignore additional filters that are not part of the abstract ReportCriteria.

API to download report will set appropriate response headers like normal file download and open data stream directly.

## Agent

### Overview

Agent 4.0 is derived from Agent 3.x with the following new changes added to adapt the 4.0 architecture

* Provide a Restful API interface, for Common Client
* Communicate with 4.0 Application Server in Restful API without a Session
* Protect encrypted configuration files where security restriction applies by Certificate instead of hard-coded password in Java class

### Communication with Application Server

To communicate with server, Agent implements com.medtronic.gateway.agent.server.invoker.AbstractInvoker per server API. AbstractInvoker’s base functions will take care of the following tasks, except override function is implemented per request invoker –

* Call xmlToJson to transform legacy IXMLMessage request into POJO request which later will be serialized into JSON request body
* Call preProcessServerRequest to encrypt JSON request body if double encryption is enabled (by default)
* Call Server API to process the request
* Call preProcessServerResponse to decrypt JSON response body if double encryption is enabled (by default)
* Call jsonToXml to transform JSON response data into legacy IXMLMessage message response

com.medtronic.gateway.crypto.AESMessageCrypto will handle pre-processing of request/response JSON body.

It uses AES/CFB8/NOPADDING and key generation algorithm is MD5. Application Server keeps the same configuration so server can decrypt Agent’s message correctly.

### Common Request Handling

XMLMessage request handling is done by com.medtronic.gateway.agent.request.RequestProcessor.

RequestProcessor has a full map of request type vs request handler. For each request, RequestProcessor will follow this workflow:

* Determine RequestHandler by request type (login, getheader, postnotification, …)
* Create a new instance of RequestHandler by calling com.medtronic.gateway.agent.request.RequestHandler#newRequestHandler, usually overriden by each implementation of RequestHandler
* Call com.medtronic.gateway.agent.request.RequestHandler#handleMessage to handle request message
  + If Agent is connected to server, request will be sent to Application Server by invoking request invoker – an implementation to com.medtronic.gateway.agent.server.invoker.AbstractInvoker
  + If Agent is disconnected to server, call com.medtronic.gateway.agent.request.RequestHandler#localProcess, usually overriden to do offline processing

For each request type there is an implementation extending com.medtronic.gateway.agent.request.RequestHandler. Each implementation will override functions to handle requests in both online and offline mode where applicable. When online, after application server processes request successfully, further processing will be handled by com.medtronic.gateway.agent.request.RequestHandler #handleServerResponse, usually overriden by each implementation to update local H2 database or local files for offline purpose.

### Restful API Requests Handling

To maximize re-use and unify client request processing logic, a Restful API request (currently from Common Client only) is usually handled in Agent in the following workflow

* Validate user session and request
* Generate XMLMessage that contains an XMLMessageRequest, just like how legacy client produces request message
* Invoke com.medtronic.gateway.agent.request.RequestProcessor’s process() method to process the message, just like how Agent process requests from legacy client
* When a response message that implements IXMLMessage is returned, re-wrap result into a POJO that will be later serialized into JSON response to Common Client

For those requests that cannot share the same logic as legacy client, like function exclusive in Restful API – for example /DownloadSoftware, Agent implements its logic separately.

com.medtronic.gateway.agent.service.rest.agent.TransferHelper implements those functions to handle specific requests, without sharing the legacy request handling workflow – which is unavailable for a set of APIs.

com.medtronic.gateway.agent.service.rest.agent.RestController implements all the Restful API interface, as a request Responder.

### File Encryption

Agent uses the same algorithm to process files downloaded from application server. By default, it is AES/CFB8/NOPADDING and key generation algorithm is MD5 that hashes a plain text passcode.

Passcode is stored in an encrypted configuration file – crypto.properties.

When downloading files from server, file content is already encrypted and compressed. Agent stores them on local disk as is.

When requested by client for a specific file, Agent will uncompress and decrypt it.

The same algorithm is used to encrypt device log files.

### Configuration Protection

Most of the configuration files are protected – encrypted and encoded in Base64. There are two sets of encrypted configuration files –

* crypto.properties

This file is encrypted by asymmetric algorithm with public key and has to be decrypted by private key. Inside this file, a passcode that will be used access other protected files is stored. Certificate is stored in Windows Certificate Store under user “gdmpagent” with private key export disabled when importing the certificate during the installation of Agent. User “gdmpagent” has flags ADS\_UF\_PASSWD\_CANT\_CHANGE and ADS\_UF\_DONT\_EXPIRE\_PASSWD set during the installation process.

* server.properties/string.properties and H2 database schema initialization scripts (config/schema)

These files are encrypted by symmetric algorithm with a passcode stored in crypto.properties. Algorithm is Java’s standard PBE algorithm which uses a key generation algorithm hashing passcode 1000 times with salt derived from passcode.

com.medtronic.gateway.crypto.Crypto implements all functions to encrypt/decrypt various protected files.

### H2 Database

Agent uses H2 database as embedded mode to store information like device info for offline use. Database file is encrypted by H2’s built-in security feature. Selected algorithm is AES.

When Agent starts, it will check if H2 database has been created or not. If not, it will create H2 database and initialize table structures.

SQL scripts to initialize table structures store in config/schema. There are several SQL scripts to run to prepare all tables.

## Web

### High Level Components Overview

Web consists of several major components that handle various responses.

* Main Modules

Web provides three main modules.

* ModAPI module
  + apiServices – Restful API Factory to handle Server-side responses.
* ModAuth module
  + authServices – Restful API Factory to handle Server-Side authentication service.
  + featureServices – Restful API Factory to handle Server-Side feature service.
* MedApp module
  + adminRoleController – Controller to handle UI in Admin page.
  + adminSubTabController – Controller to handle default subtab routing under Admin page.
  + adminTabController – Controller to handle default tab routing under Admin page.
  + agentDownloadController – Controller to handle UI in Agent Download page.
  + alertRoleController – Controller to handle UI in Alert page and Alert Edit page.
  + configurationController – Controller to handle UI in Configuration Edit page.
  + defaultTrainerController – Controller to handle UI in Default Trainer tab under Admin page.
  + deviceController – Controller to handle UI in Device Detail page.
  + editController – Controller to handle UI in edit modal of Hardware and Document page.
  + facilityController – Controller to handle UI in Customer & Facility tab under Admin page.
  + featureController – Controller to handle UI in Feature Edit page.
  + featureEntitlementController – Controller to handle UI in Feature Entitlement Edit page.
  + featureListController – Controller to handle UI in Feature page.
  + fileExtensionController – Controller to handle UI in File Extension Whitelist subtab under Miscellaneous tab under Admin page.
  + generalEditController – Controller to handle UI in General Notification Edit page.
  + homeController – Controller to handle UI in Home page.
  + loginController – Controller to handle UI in Login page.
  + miscellaneousTabController – Controller to handle UI in Miscellaneous tab under Admin page.
  + passwordController – Controller to handle UI in Password Resetting page.
  + pendingController – Controller to handle UI in Pending Registration link page opened from email.
  + pendingRegController – Controller to handle UI in Pending Registration Edit page.
  + reportController – Controller to handle UI in each report of Report page.
  + roleController – Controller to handle UI in Role Entry, Customer & Facility, Approving Manager, Client Software tabs under Admin page, Alert page, Configurations page, Devices page, Documents page, Hardware page, Report page and Software page.
  + rolePermissionsController – Controller to handle UI in Role & Permissions tab under Admin page
  + selfRegController – Controller to handle UI in Self-Registration page.
  + softwarePackagesController – Controller to handle UI in Software Packages tab under Admin page.
  + swController – Controller to handle UI in Software Edit page.
  + templatesEditController – Controller to handle UI in Templates Edit page
  + tradeEmbargoController – Controller to handle UI in Trade Embargo subtab in Miscellaneous tab under Admin page.
  + userController – Controller to handle UI in User Edit page.
  + userListController – Controller to handle UI in User page.
  + app – Configuration Provider to provide routing servers.
  + otherServices – Contains other services, filters and directives.

### Template Structure

Each folder contains several templates

* Admin
  + client-software – UI template for Client Software Tab.
  + default-trainer – UI template for Default Trainer Tab.
  + edit-clientsoftware – UI template for Client Software Edit Modal under Client Software Tab.
  + edit-customer – UI template for Customer Edit Modal for Customer Subtab under Customer & Facility Tab.
  + edit-facility – UI template for Facility Edit Modal for Facility Subtab under Customer & Facility Tab.
  + edit-mdtapprovemgr – UI template for Approving Manager Edit Modal for Approving Manager Tab.
  + edit-role – UI template for Role Edit Modal for Role Entry Tab.
  + file-extension – UI template for File Extension Whitelist Subtab under Miscellaneous Tab.
  + main-tab – UI template for main tabs in Admin Page.
  + role-permissions – UI template for Role & Permissions Tab.
  + search-general – UI template for general search box for pages where allows user to search by keywords.
  + software-package – UI template for Software Package Tab.
  + sub-tab – UI template for sub tabs in Admin page.
  + trade-embargo – UI template for Trade Embargo Tab.
* Alert
  + alertsEdit – UI template for Alert Edit Modal in My Alert Tab.
  + edit – UI template for including each edit template by tab name.
  + general – UI template for General Notification Subtab in Admin Tab.
  + generalEdit – UI template for Gerneral Notification Edit Modal in General Notification Subtab.
  + manageAlerts – UI template for subtabs under Admin Tab.
  + manageSub – UI template for Manage Subscriptions Subtab in Admin Tab.
  + myAlerts – UI template for My Alerts Tab.
  + role – UI template for main tabs in Alerts page.
  + subscription – UI template for My Subscriptions Tab.
  + template – UI template for Template subtab in Admin Tab.
* Configuration
  + edit – UI template for Configuration Edit Modal in Configuration page.
* Devices
  + configurations – UI template for Configuration Tab in Device Detail Modal in Device page.
  + discrepancies – UI template for Discrepancies Tab in Device Detail Modal in Device page.
  + edit – UI template for Device Detail Modal in Device page.
  + feature-history – UI template for Feature History Tab in Device Detail Modal in Device page.
  + history – UI template for History Tab in Device Detail Modal in Device page.
  + logs – UI template for Logs Tab in Device Detail Modal in Device page.
* Document
  + edit – UI template for Document Edit Modal in Document page.
* Feature
  + edit – UI template for Feature Edit Modal in Feature Catalog Tab in Feature page.
  + entitlement-edit – UI template for Feature Entitlement Edit Modal in Feature Entitlement Tab in Feature page.
  + role – UI template for main tabs in Feature page.
* Hardware
  + edit – UI template for Hardware Edit Modal in Hardware page.
* Report
  + auditTrailReport – UI template for Audit Trail Report Page in Report page.
  + cloneReport – UI template for Clone Report Page in Report page.
  + componenetDiscrepancyReport – UI template for Component Discrepancy Report Page in Report page.
  + configurationWhereUsedReport – UI template for Configuration Where Used Report Page in Report page.
  + deviceCountryChangeReport – UI template for Device Country Change Report Page in Report page.
  + deviceCurrentConfigurationReport – UI template for Device Current Configuration Report Page in Report page.
  + deviceHistoricalConfigurationReport – UI template for Device Historical Configuration Report Page in Report page.
  + footer – UI template for footer of the table in each report page.
  + header – UI template for header of the table in each report page.
  + modalError – UI template for Error Alert Modal in Report page.
  + namedConfigurationReport – UI template for Named Configuration Report Page in Report page.
  + printReport – UI template for Print Preview Page for each report page.
  + reportHeader – UI template for Report page.
  + serviceRecordsReport – UI template for Service Record Report Page in Report page.
  + softwareAuditReport – UI template for Software Audit Report Page in Report page.
  + softwareUpdateReport – UI template for Software Update Report Page in Report page.
  + softwareVersionReport – UI template for Software Version Report Page in Report page.
  + trainingReport – UI template for Training Report Page in Report page.
  + userActivityReport – UI template for User Activity Report Page in Report page.
  + viewReport – UI template for View Button of each report page.
* Shared
  + alert – UI template for General Alert Prompt Box in application.
  + alertWithConfirmation – UI template for General Alert Prompt Box With Confirmation in application.
  + associatedConf – UI template for Associated Configuration List.
  + associateItemTypes – UI template for Associated Item Input Box.
  + associateTypes – UI template for Associated Configuration Input Box.
  + cloneMap – UI template for Clone Map Section in item detail modals.
  + cloneMapItem – UI template for Clone Map Table in item detail modals.
  + compatibleItems – UI template for Compatible Items Section in item detail modals.
  + countryExclusion – UI template for Country Exclusion Section in item detail modals.
  + deviceItems – UI template for Configuration Information Table in Configurations Tab in Device Detail Modal.
  + facilityItems – UI template for Customer Detail Modal in Customer subtab under Customer & Facility Tab in Admin page.
* Software
  + edit – UI template for Software Edit Modal in Software page.
* User
  + edit-pending-reg – UI template for Pending Registration Edit Modal in Pending Registration Tab in User page.
  + edit – UI template for User Edit Modal in User Tab in User page.
  + role – UI template for Tabs in User page.
* Others
  + agent-download – UI template for Download Agent and Client Application page.
  + change-password-popup – UI template for Change Password Popup Box.
  + change-password – UI template for Change Password Modal.
  + edit – UI template for including other edit templates.
  + editError – UI template for Error Prompt when no data.
  + forget-password – UI template for Forget Password Modal.
  + header – UI template for navigation bar of the Home page.
  + home – UI template for Home page.
  + login – UI template for Login page.
  + pdfView – UI template for PDF preview modal in Document page.
  + pending – UI template for Pending Registration Link Page.
  + report – UI template for gathering header and body for each report page.
  + reset-password – UI template for Reset Password page.
  + role – UI template for common header and body for each role page.
  + self-registration – UI template for Self-Registration page.
  + user-profile – UI template for User Information Section.

### General Table Structure

* The table for each role shares the same structure.
* Columns are listed based on responded columns data.
* Filters options are listed in each category based on responded filter data with default value.
* Detail of each item are displayed based on table columns.

## Tools

Tools like Remapper Tool, Connection Tool and ETL are derived from 3.x system while a couple of changes have been made to adapt the 4.0 Agent/Application Server/Data Schema design.

### Connection Tool

The following changes have to be made in Connection Tool 4.0 to make it work with Agent 4.0

* Agent Service Name
* Location of file to read for current server address

### Remapper Tool

The following changes have to be made in Remapper Tool to make it work with Agent 4.0

* Agent Service Name
* Re-use Agent’s encryption/decryption method to update configuration file (for purpose like changing server address/port)

### ETL

The following changes have to be made in ETL to make it work with 4.0 Database schema

* Eliminate nid but use id – primary key of data table
* Update SQL statements to fit new table structures including
  + customer – for Customer data
  + postal\_address – for Location data
  + postal\_address\_customer\_relation – for Location Role data
  + device – for Device data
  + device\_sku – for Device SKU data

Here is a list of data access object to implement in ETL for 4.0 -

| **DAO** | **Description** |
| --- | --- |
| CustomerDAO | 1. delete customer softly 2. update customer with new name, voice address 3. add new customer |
| DeviceDAO | 1. delete device softly 2. update device with customer, maintenance expire date, location,country 3. add new device |
| LocationDAO | 1. delete location softly 2. update location with new address, city, province, post code, country 3. add new location |
| LocationRoleDAO | 1. delete location role softly 2. add new location role |
| CustomerRenumberDao | 1.update customer with new account number  2. update device owner (customer)  3. update relation between user and customer  4. delete customer softly  5. add new customer if necessary |

## Appendix

### Agent-Server API

#### User login

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/users/login | GET | YES | User login from Client |
| Request Parameter:  None | | | |
| Response:  {  "timestamp": 1456802441753,  "country": "CN",  "covidien\_user": true,  "deactivated": true,  "pass\_expiration": "2016-03-01 09:00:01.0",  "accessible\_device\_types": [{  "name": "SCD 700",  "raw\_name": "SCD 700",  "device\_access": 2,  "software\_access": 1,  "limited\_release": true  }, {  "name": "5768C9CB-8ACE-4421-B1C6-071D131B935F",  "raw\_name": "SigniaTMStapler",  "device\_access": 3,  "software\_access": 6,  "limited\_release": true  },  "trained\_device\_types": ["SCD 700", "5768C9CB-8ACE-4421-B1C6-071D131B935F "]  } | | | |
| Note:  Device access could be  NONE=0 (No rights at all)  LOG\_CFG\_UPLOAD=1 (Can upload log file and device configuration)  SW\_UPDATE=2 (Can update software on device)  LICENSE\_UPDATE=4 (Can update feature license on device)  or any combination of LOG\_CFG\_UPLOAD/SW\_UPDATE/LICENSE\_UPDATE  Software access could be  NONE=0 (No rights to update software)  LATEST\_ONLY=1 (Can only update latest software)  ALL\_RELEASE=2 (Can update all released software)  Limited release: user has right to update limited software or not | | | |

#### User logoff

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/users/logoff | GET | YES | User logoff from Client |
| Request Parameter:  None | | | |
| Response:  {  "timestamp": "1456802441753"  } | | | |

#### Resync user permissions

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/users/permission | POST | YES | Return a list of user permissions |
| Request Parameter:  {  "user\_list": ["jack.ye@medtronic.com", "lye@insgimaus.com"]  } | | | |
| Response:  {  “timestamp”: “1456802441753”,  “users”: [  {sub JSON format for user, see below},  {sub JSON format for user, see below}  ]  } | | | |
| (sub JSON format for user, Highlight part is also in login response)  {  "user\_name": "lye@insigmaus.com",  "password\_md5": "amFjazoxMjM=",  "deactivated": true,  "pass\_expiration": "2016-03-01 09:00:01.0",  "accessible\_device\_types": [{  "name": "SCD 700",  "device\_access": 2,  "software\_access": 1,  "limited\_release": true  }, {  "name": "PB980",  "device\_access": 3,  "software\_access": 2,  "limited\_release": false  }],  "trained\_device\_types": ["SCD 700", "PB980"]  } | | | |
| Note: The user list used to be MD5 string of user email. Since the login information already encrypted on Agent, it won’t hurt to use the email address directly. There is no requirement ask for using MD5 string of user mail. | | | |

#### 6.4 Get Hardware/software configuration

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/config/hwsw | POST | YES | Return HW/SW configuration of device types |
| Request: (Currently it’s one version for all device types, might change to each device type has its own version)  {  "version": "123",  "device\_type\_guids": ["SCD 700 GUID", "PB980 GUID"]  } | | | |
| Response: (No more device embargo, should remove it from Agent’s code accordingly)  {  "timestamp": "1456802441753",  "version": "latest version",  "up\_to\_date": "true or false",  "hwsw\_configs": [{  "device\_type\_guid": "SCD 700 GUID",  "hardware\_list": hardware config JSON, see sub JSON,  "software\_list": software config JSON, see sub JSON,  "relationship\_list": HWSW relationship, see sub JSON,  "document\_list": document JSON, see sub JSON,  "doc\_relationship\_list": document relationship JSON, see sub JSON,  "feature\_list": feature JSON, see sub JSON  }, {  "device\_type\_guid": "PB980 GUID",  "hardware\_list": hardware config JSON, see sub JSON,  "software\_list": software config JSON, see sub JSON,  "relationship\_list": HWSW relationship, see sub JSON,  "document\_list": document JSON, see sub JSON,  "doc\_relationship\_list": document relationship JSON, see sub JSON,  "feature\_list": feature JSON, see sub JSON  }]  } | | | |
| hardware config(Sub format used in hardware\_list)  [{  "hash": "da690031c3a78f0db229e63d827c44d5",  "name": "Control Board",  "part\_number": "0",  "revision": "0"  }, {  "hash": "aa690031c3a78f0db229e63d827c44d4",  "name": "Control Board2",  "part\_number": "1",  "revision": "1"  }] | | | |
| software config(Sub format used in software\_list)  [{  "type": "Business Rules",  "expiration": "0",  "file\_size": "3086",  "hash": "8af306c7c770f90cf5032767e1633675",  "md5": "a61883315b1dd9e663d76bd45f99808ae90b4b7d",  "name": "SCDUBusinessRules",  "part\_number": "0",  "revision": "2.0",  "status": "In Production",  "language": "US English",  "comparison\_order": "1",  "uri": "https://rssqa-app.covidien.com/software/1390316402\_SCD700\_Updater\_3.config.zip",  "regulatory\_exclusion": ["XA", "CN"]  }, {  "type": "Software Bundle",  "expiration": "0",  "file\_size": "3087",  "hash": "8af306c7c770f90cf5032767e1633675",  "md5": "a61883315b1dd9e663d76bd45f99808ae90b4b7d",  "name": "SCDU",  "part\_number": "0",  "revision": "2.0",  "status": "In Production",  "language": "US English",  "comparison\_order": "2",  "uri": "https://rssqa-app.covidien.com/software/1390316402\_SCD700\_Updater\_4.config.zip",  "regulatory\_exclusion": ["XA", "CN"]  }]  Note: in old system, there is a Nid field. Agent use it as comparison\_order for Agent business rules. (The largest Nid is the latest version). I think new system won’t have Nid in database, need figure out another way. | | | |
| document config(Sub format used in document\_list)  [{  "type": "Service Manual",  "expiration": "0",  "external\_user": "true or false",  "file\_size": "86474",  "hash": "918d320de93cf36c315f1bc05105b36d",  "md5": "9ec48aed70f15c07f86f95c4a4e14241982ab277",  "name": "DMP Overview and Contact Information",  "part\_number": "13MAN0089A",  "revision": "12\_13",  "status": "update",  "uri": "https://rssqa-app.covidien.com/document/other/1390392602\_DMP Information.pdf.zip"  }, {  "type": "Service Manual",  "expiration": "0",  "external\_user": "true or false",  "file\_size": "86474",  "hash": "918d320de93cf36c315f1bc05105b36d",  "md5": "9ec48aed70f15c07f86f95c4a4e14241982ab277",  "name": "DMP Overview and Contact Information",  "part\_number": "14MAN0089A",  "revision": "12\_14",  "status": "update",  "uri": "https://rssqa-app.covidien.com/document/other/1390392602\_DMP Information.pdf.zip"  }] | | | |
| hardware/software relationship(Sub format used in relationship\_list)  [{  "hardware\_hash": "da690031c3a78f0db229e63d827c44d5",  "software\_hashes": [  "8af306c7c770f90cf5032767e1633675",  "9bb1f8ba908e32f0dbb4811aa4da0729",  "14bc6e8db8b7b94cf7a0ec5080d007f9"  ]  }, {  "hardware\_hash": "aa690031c3a78f0db229e63d827c44d9",  "software\_hashes": ["9af306c7c770f90cf5032767e1633676"]  }] | | | |
| document relationship with hardware/software (Sub format used in doc\_relationship\_list)  [{  "document\_hash": "da690031c3a78f0db229e63d827c44d5",  "hardware\_hashes": ["8af306c7c770f90cf5032767e1633675"],  "software\_hashes": ["9bb1f8ba908e32f0dbb4811aa4da0729", "14bc6e8db8b7b94cf7a0ec5080d007f9"]  }, {  "document\_hash": "aa690031c3a78f0db229e63d827c44d6",  "hardware\_hashes": ["hash": "aaf306c7c770f90cf5032767e1633676"]  }] | | | |
| feature config (Sub format used in feature\_list)  [{  "name": "feature name x",  "SKU": "feature SKU x",  "description": "feature description x",  "status": "In Production",  "regulatory\_exclusion": ["XA", "CN"]  }, {  "name": "feature name x",  "SKU": "feature SKU x",  "description": "feature description x",  "status": "In Production",  "regulatory\_exclusion": ["XA", "CN"]  }] | | | |
| Sample Full Response:  {  "timestamp": 1464969354917,  "version": 4,  "up\_to\_date": false,  "hwsw\_configs": [  {  "device\_type\_guid": "B0DC2BE4-D744-45c6-AEF6-EBEF319A336B",  "hardware\_list": [],  "software\_list": [],  "document\_list": [],  "relationship\_list": [],  "doc\_relationship\_list": [],  "feature\_list": []  },  {  "device\_type\_guid": "3B682913-6D1E-4355-9E48-208EB7061A3D",  "hardware\_list": [  {  "name": "LS10\_HW\_1",  "hash": "c4ca4238a0b923820dcc509a6f75849b",  "part\_number": "PH01",  "revision": "0.1"  },  {  "name": "LS10\_HW\_2",  "hash": "c81e728d9d4c2f636f067f89cc14862c",  "part\_number": "PH02",  "revision": "0.1"  }  ],  "software\_list": [  {  "name": "LS10\_SW\_1",  "type": "Software Bundle",  "md5": "AAAAAAAAAAAAAAAAA",  "status": "In Production",  "fileId": 1,  "expiration": null,  "hash": "c4ca4238a0b923820dcc509a6f75849b",  "language": "eng",  "part\_number": "PS01",  "revision": "0.1",  "file\_size": 1024,  "comparison\_order": 1,  "regulatory\_exclusion": [  "IR",  "IL",  "KP"  ]    ],  "relationship\_list": [  {  "hardware\_hash": "28c8edde3d61a0411511d3b1866f0636",  "software\_hashes": [  "c4ca4238a0b923820dcc509a6f75849b",  "c81e728d9d4c2f636f067f89cc14862c",  "eccbc87e4b5ce2fe28308fd9f2a7baf3"  ]  },  {  "hardware\_hash": "665f644e43731ff9db3d341da5c827e1",  "software\_hashes": [  "c4ca4238a0b923820dcc509a6f75849b",  "c81e728d9d4c2f636f067f89cc14862c",  "eccbc87e4b5ce2fe28308fd9f2a7baf3"  ]  },  {  "hardware\_hash": "38026ed22fc1a91d92b5d2ef93540f20",  "software\_hashes": [  "c4ca4238a0b923820dcc509a6f75849b",  "c81e728d9d4c2f636f067f89cc14862c",  "eccbc87e4b5ce2fe28308fd9f2a7baf3"  ]  },  {  "hardware\_hash": "011ecee7d295c066ae68d4396215c3d0",  "software\_hashes": []  },  {  "hardware\_hash": "4e44f1ac85cd60e3caa56bfd4afb675e",  "software\_hashes": []  }  ],  "doc\_relationship\_list": [],  "feature\_list": []  },  {  "device\_type\_guid": "CEFC1E07-CFF6-4F27-AB05-4577A33A1BA8",  "hardware\_list": [  {  "name": "FT10\_HW\_1",  "hash": "1679091c5a880faf6fb5e6087eb1b2dc",  "part\_number": "PHA1",  "revision": "0.1"  }  ],  "software\_list": [  {  "name": "FT10\_SW\_1",  "type": "Software Bundle",  "md5": "FFFFFFFFFFFFFFFFF",  "status": "In Production",  "fileId": 6,  "expiration": null,  "hash": "1679091c5a880faf6fb5e6087eb1b2dc",  "language": "eng",  "part\_number": "PSA1",  "revision": "0.1",  "file\_size": 1024,  "comparison\_order": 1,  "regulatory\_exclusion": [  "IR",  "IL",  "KP"  ]  }  ],  "document\_list": [  {  "name": "FT10\_DOC\_1",  "md5": "FFFFFFFFFFFFFFFFF",  "status": "In Production",  "fileId": 21,  "type": "Release Notes",  "expiration": null,  "hash": "8f14e45fceea167a5a36dedd4bea2543",  "part\_number": "PSDA1",  "revision": "0.1",  "file\_size": 1024,  "external\_user": true  },  ],  "relationship\_list": [  {  "hardware\_hash": "3d2f8900f2e49c02b481c2f717aa9020",  "software\_hashes": []  },  {  "hardware\_hash": "cd7fd1517e323f26c6f1b0b6b96e3b3d",  "software\_hashes": []  },  {  "hardware\_hash": "815e6212def15fe76ed27cec7a393d59",  "software\_hashes": []  },  {  "hardware\_hash": "4c0d13d3ad6cc317017872e51d01b238",  "software\_hashes": []  },  {  "hardware\_hash": "8d8e353b98d5191d5ceea1aa3eb05d43",  "software\_hashes": []  }  ],  "doc\_relationship\_list": [],  "feature\_list": []  }  ]  } | | | |

#### Get named configuration

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/config/named/{device\_type\_guid} | GET | YES | Return named configuration of device types |
| Request:  {device\_type\_guid} from URL | | | |
| Response:  {  "timestamp": "1456802441753",  "device\_type\_guid": "3B682913-6D1E-4355-9E48-208EB7061A3D",  "system\_configs": [{  "type": "normal",  "name": "TEST SYSTEM",  "revision": "1",  "description": "TEST ",  "update\_time": "1431675019",  "status": "Archived",  "named\_hw\_config\_list": named hardware config, see sub JSON,  "named\_sw\_config\_list": named software config, see sub JSON,  "named\_fl\_config\_list": named feature config, see sub JSON  }, {  "type": "invalid-hardware",  Same as normal system config  }, {  "type": "incompatible-software",  Same as normal system config  }],  "software\_doc\_list": software document list, see sub JSON,  "hardware\_doc\_list": hardware document list, see sub JSON  } | | | |
| Named hardware config(Sub format used in named\_hw\_config\_list)  [{  "description": "VLL10 Test",  "name": "VLL10\_SH\_LX",  "status": "Archived",  "update\_time": "1435285609",  "revision": "1",  "hardware\_items": [{  "type": "MOTHERBOARD",  "name": "temp17",  "part\_number": "0",  "revision": "0",  "status": "Archived"  }, {  "type": "MOTHERBOARD",  "name": "temp18",  "part\_number": "0",  "revision": "1",  "status": "Archived"  }]  }, {  Another named hardware config  }] | | | |
| Named software config(Sub format used in named\_sw\_config\_list)  [{  "description": "TEST",  "name": "VLL10\_SH\_LX\_03",  "status": "Archived",  "update\_time": "1431673413",  "revision": "1",  "software\_items": [{  "type": "Software Bundle",  "primary": "true or false",  "file\_size": "343152",  "md5": "034fea74bc2b35b58b752727a290cb73e532ff59",  "name": "ValleyLab LS10 Software Package",  "part\_number": "0",  "regulatory\_exclusion": ["AF", "CN"],  "revision": "01.00.03\_bak",  "status": "In Production",  "uri": "url address",  "comparison\_order": "9",  "documents": [{  "file\_size": "11299",  "md5": "b86b8614f05f5caa81eaae00feae5e44136ee1f5",  "name": "VLS10\_Document",  "part\_number": "abc",  "revision": "2.0",  "status": "Limited Release",  "uri": "url address"  }]  }]  }, {  Another named software config  }] | | | |
| Named feature config(Sub format used in named\_fl\_config\_list)  [{  "description": "my feature test description",  "name": "feature test",  "status": "Archived",  "update\_time": "1435285609",  "revision": "1",  "feature\_items": [{  "name": "feature name",  "SKU": "sku",  "description": "desc",  "status": "Archived",  "regulatory\_exclusion": ["XA", "CN"]  }, {  "name": "feature name",  "SKU": "sku",  "description": "desc",  "status": "Archived",  "regulatory\_exclusion": ["XA", "CN"]  }]  }, {  Another named feature config  }] | | | |
| Software document list(Sub format used in software\_doc\_list)  [{  "name": "valleylab ls10 software package",  "revision": "01.00.04\_bak",  "part\_number": "0",  "documents": [{  "file\_size": "181957",  "md5": "13a78846b401a65eee39c6c3dd74b7c1718daed9",  "name": "LS10\_Document",  "part\_number": "0",  "revision": "1.0",  "status": "In Production",  "uri": "url address"  }]  }, {  Another software  }] | | | |
| Hardware document list(Sub format used in hardware\_doc\_list, this actually same as software\_doc\_list)  [{  "name": "temp4",  "revision": "0",  "part\_number": "0",  "documents": [{  "file\_size": "11299",  "md5": "b86b8614f05f5caa81eaae00feae5e44136ee1f5",  "name": "VLS10\_Document",  "part\_number": "abc",  "revision": "2.0",  "status": "Limited Release",  "uri": "url address"  }]  }, {  Another hardware  }] | | | |

#### Stat device (for device 2.0)

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/device/stat2 | POST | YES | Return device configuration if device exists |
| Request:  {  "device\_type\_guid": "3B682913-6D1E-4355-9E48-208EB7061A3D",  "serial\_number": "35B12P3001",  "country": "CN",  "region": "Asia"  } | | | |
| Response:  {  "timestamp": "1456802441753",  "device\_exist": "true or false",  "device\_type\_guid": "3B682913-6D1E-4355-9E48-208EB7061A3D ",  "serial\_number": "35B12P3001",  "country": "CN",  "facility": "facility",  "device\_address": "address",  "component\_checksum": "111",  "accessible": "true or false",  "last\_update": "2016-02-02 12:26:26",  "hardware\_list": hardware list, see sub JSON,  "software\_list": software list, see sub JSON,  "feature\_list": feature list, see sub JSON  } | | | |
| Hardware list(Sub format used in hardware\_list)  [{  "name": "Control Board",  "revision": "0",  "part\_number": "0",  "serial\_number": "xxx"  }, {  "name": "Control Board2",  "revision": "1",  "part\_number": "1",  "serial\_number": "yyy"  }] | | | |
| Software list(Sub format used in software\_list)  [{  "name": "Control",  "revision": "01.08.01",  "part\_number": "0"  }, {  "name": "Control2",  "revision": "01.02",  "part\_number": "1"  }] | | | |
| Feature list(Sub format used in feature\_list)  [{  "id": "Feature id",  "name": "Feature name",  "SKU": "sku",  "status ": "Active",  "license\_term": {  "type": "LimitedCount",  "limit": 10,  "used": 2,  "remaining": 8,  "start\_time": "",  "end\_time": "",  }  }, {  "id": "Feature id",  "name": "Feature name",  "SKU": "sku",  "status ": "Active",  "license\_term": {  "type": "LimitedDuration",  "limit": 0,  "used": 0,  "remaining": 0,  "start\_time": "2015-12-01 00:00:00",  "end\_time": "2016-12-01 00:00:00",  }  }]  Note:  License term type could be: LimitedCount, LimitedDuration, Unlimited | | | |

#### Stat device (for device 3.0)

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/device/stat3 | POST | YES | Return named configuration if device exists |
| Request:  {  "device\_type\_guid": "3B682913-6D1E-4355-9E48-208EB7061A3D ",  "serial\_number": "35B12P3001",  "country": "CN",  "region": "Asia"  } | | | |
| Response:  {  "timestamp": "1456802441753",  "device\_exist": "true or false",  "device\_type\_guid": "3B682913-6D1E-4355-9E48-208EB7061A3D ",  "serial\_number": "35B12P3001",  "country": "CN",  "facility": "facility",  "device\_address": "address",  "component\_checksum": "111",  "accessible": "true or false",  "last\_update": "2016-02-02 12:26:26",  "named\_hardware\_config\_list": hardware config list, see sub JSON,  "named\_software\_config\_list": software config list, see sub JSON  "feature\_list": same as [Stat device for 2.0](#_6.6_Stat_device)  } | | | |
| Named hardware config list(Sub format used in named\_hw\_config\_list)  [{  "type": "MOTHERBOARD",  "name": "Main PCBA",  "part\_number": "1077885",  "revision": "\_C",  "serial\_number": "D142800026"  }, {  "type": "MOTHERBOARD",  "name": "Main PCBA2",  "part\_number": "1077887",  "revision": "\_D",  "serial\_number": "D142800066"  }] | | | |
| Named software confg list(Sub format used in named\_sw\_config\_list)  [{  "type": "Software Bundle",  "name": "Host Bootloader",  "part\_number": "0",  "revision": "01.00"  }, {  "type": "Software Bundle",  "name": "Host Bootloader",  "part\_number": "0",  "revision": "01.00"  }] | | | |

#### Create device

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/device | PUT | YES | Add a device to GDMP system |
| Request:  {  "device\_type\_guid": "3B682913-6D1E-4355-9E48-208EB7061A3D ",  "serial\_number": "35B12P3001",  "country": "CN",  "region": "Asia",  "facility\_id":  } | | | |
| Response:  {  "timestamp": "1456802441753"  } | | | |

#### Register URL

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/regurl | GET | NO | Get register URL |
| Request:  None | | | |
| Response:  {  "timestamp": "1456802441753",  "url": "http://test.covidien.com/self/home"  } | | | |

#### Forgot password

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/forgotpassword | POST | NO | Forgot password |
| Request:  {  "username": "lye@insigmaus.com"  } | | | |
| Response:  {  "timestamp": "1456802441753"  } | | | |

#### Change password

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/changepassword | POST | YES | Change password |
| Request: (Password is plain text because System need to check the password strength)  {  "username": "lye@insigmaus.com",  "password": " old password, not md5 hash",  "new\_password": " new password, not md5 hash"  } | | | |
| Response:  {  "timestamp": "1456802441753"  } | | | |

#### SN re-program

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/snreprogram | POST | YES | SN re-program |
| Request:  {  "device\_type\_guid": "3B682913-6D1E-4355-9E48-208EB7061A3D ",  "serial\_number": "35B12P3001",  "old\_sn": "35B12P3001",  "new\_sn": "35B12P3009"  } | | | |
| Response:  If successful:  {  "timestamp": "1456802441753"  }  Otherwise:  {  "code": 409,  "msg": "Device Update Failure" or “No Such Device” or “Duplicate Device” or “Device Create Failure”  "detail": "ba1b1815-6368-4718-98fc-2b90115e16ac",  "timestamp" : 1456802441753  } | | | |

#### Download audit

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/download/audit | PUT | YES | Audit software download |
| Request:  {  "device\_type\_guid": "SCD 700",  "serial\_number": "35B12P3001",  "software\_list": [{  "type": "Business Rules",  "name": "SCDUBusinessRules",  "revision": "2.0"  }, {  "type": "Software Bundle",  "name": "SCDU",  "revision": "1.0"  }]  } | | | |
| Response:  {  "timestamp": "1456802441753"  } | | | |

#### Client upgrade

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/clientupgrade | PUT | YES | Audit when Agent/Client upgraded |
| Request:  {  "client\_type\_guid": "ADMINISTRATIVE\_AGENT",  "serial\_number": "ADMINISTRATIVE\_AGENT",  "type": "Business Rules or Software",  "name": "Software name",  "revision": "0",  "reason": "downloaded or upgraded"  } | | | |
| Response:  {  "timestamp": "1456802441753"  } | | | |

#### Device info (only for device 2.0)

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/device/config2 | POST | YES | Sync device config to GDMP, return all software/documents that related to the device type |
| Request:  {  "device\_type\_guid": "3B682913-6D1E-4355-9E48-208EB7061A3D ",  "serial\_number": "35B12P3001",  "country": "CN",  "region": "Asia",  "facility": "facility",  "device\_address": "address",  "hardware\_list": hardware list, see [Stat Device Response for 2.0](#_5.6_Stat_device),  "software\_list": software list, see [Stat Device Response for 2.0](#_5.6_Stat_device),  "feature\_list": same as [Stat device for 2.0](#_6.6_Stat_device)  } | | | |
| Response: (It will return software/document that related to this device by device type)  {  "timestamp": "1456802441753",  "component\_checksum": "111",  "software\_list": software list, see sub JSON,  "document\_list": document list, see sub JSON  } | | | |
| Software list(Sub format used in software\_list)  [{  "type": "Business Rules",  "file\_size": "3086",  "md5": "a61883315b1dd9e663d76bd45f99808ae90b4b7d",  "name": "SCDUBusinessRules",  "part\_number": "0",  "revision": "2.0",  "status": "In Production",  "language": "US English",  "comparison\_order": "1",  "uri": "https://rssqa-app.covidien.com/software/1390316402\_SCD700\_Updater\_3.config.zip",  "regulatory\_exclusion": ["XA", "CN"]  }] | | | |
| Document list (Sub format used in document\_list)  [{  "type": "Other",  "name": "DMP Overview and Contact Information",  "file\_size": "86474",  "md5": "9ec48aed70f15c07f86f95c4a4e14241982ab277",  "status": "In Production",  "uri": "url address of document",  "related\_hw\_list": [{  "name": "Control Board",  "revision": "0",  "part\_number": "0"  }],  "related\_sw\_list": [{  "name": "SCDUBusinessRules",  "revision": "2.0",  "part\_number": "0"  }]  }, {  Another document  }] | | | |

#### Sync device config (only for device 3.0)

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/device/config3 | POST | YES | Sync device config to GDMP |
| Request:  {  "device\_type\_guid": "3B682913-6D1E-4355-9E48-208EB7061A3D ",  "serial\_number": "35B12P3001",  "country": "CN",  "region": "Asia",  "facility": "facility",  "device\_address": "address",  "named\_hardware\_config\_list": hardware config list, see [Stat Device Response for 3.0](#_5.7_Stat_device),  "named\_software\_config\_list": software config list, see [Stat Device Response for 3.0](#_5.7_Stat_device)  "feature\_list": same as [Stat device for 2.0](#_6.6_Stat_device)  }  Sample:  {  "country": "ZZ",  "region": "NONE",  "requestTime": null,  "facility": "NA",  "device\_type\_guid": "5768C9CB-8ACE-4421-B1C6-071D131B935F",  "serial\_number": "C2016D4802",  "device\_address": "NA",  "named\_hardware\_config\_list": [{  "hardware\_items": [{  "type": "HARDWARE",  "name": "Signia Power Pack",  "revision": "1",  "part\_number": "NA",  "serial\_number": null  }]  }],  "named\_software\_config\_list": [{  "software\_items": [{  "type": "Software Bundle",  "name": "Software Blob",  "revision": "2016.03.01-15\_49\_50",  "part\_number": "ENG\_0301\_RELEASE"  }, {  "type": "Software Bundle",  "name": "Power Pack Application",  "revision": "2016.03.01-15\_45\_55",  "part\_number": "NA"  }, {  "type": "Software Bundle",  "name": "Power Pack Bootloader",  "revision": "2016.02.12-20\_29\_58",  "part\_number": "NA"  }, {  "type": "Software Bundle",  "name": "FPGA",  "revision": "2015.12.21-17\_05\_10",  "part\_number": "NA"  }]  }],  "feature\_list": []  } | | | |
| Response:  {  "timestamp": "1456802441753"  } | | | |

#### Client app info

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/clientapp | POST | YES | Check if there is new client app on Server |
| Request:  {  "client\_type\_guid": "61e08b77-df3c-4735-9f3b-0b42efb7bdcf",  "serial\_number": "xxx",  "software\_list": [{  "name": "SCDUInstaller",  "part\_number": "0",  "revision": "1.0.3"  }],  "business\_rule\_list": [{  "name": "SCDUBusinessRules",  "part\_number": "0",  "revision": "2.0"  }]  } | | | |
| Response:  {  "timestamp": "1456802441753",  "new\_version": "true or false",  "new\_software\_list": [{  "comparison\_order": "4",  "name": "SCDUInstaller",  "revision": "1.3.0",  "file\_size": "13120",  "language": "US English",  "md5": "98ada351e9d8ac2a6fe18789afc87f738fbb2492",  "part\_number": "0",  "status": "In Production",  "uri": "https://rssqa-intapp.covidien.com/software/SCD700\_Installer.zip"  }],  "new\_business\_rule\_list": [{  "comparison\_order": "2",  "name": "SCDUBusinessRules",  "revision": "3.0",  "file\_size": "3110",  "language": "US English",  "md5": "98ada351e9d8ac2a6fe18789afc87f738fbb2492",  "part\_number": "0",  "status": "In Production",  "uri": "https://rssqa-intapp.covidien.com/software/SCD700\_Updater.config.zip"  }]  } | | | |

#### Device software upgrade ACK

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/device/software/upgrade | PUT | YES | Acknowledge for client upgrade device software |
| Request:  {  "device\_type\_guid": "3B682913-6D1E-4355-9E48-208EB7061A3D ",  "serial\_number": "35B12P3001",  "timestamp": "1456802441753",  "timezone": "UTC+8",  "detail": "not attempted",  "localtime": "2/2/2016 12:29:11 PM"  "software": {  "name": "Control",  "part\_number": "0",  "revision": "01.08.01"  }  } | | | |
| Response:  {  "timestamp": "1456802441753"  } | | | |

#### Device feature license upgrade ACK

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/device/license/upgrade | PUT | YES | Acknowledge for client upgrade device feature license |
| Request:  {  "device\_type\_guid": "3B682913-6D1E-4355-9E48-208EB7061A3D ",  "serial\_number": "35B12P3001",  "timestamp": "1456802441753",  "timezone": "UTC+8",  "detail": "not attempted",  "localtime": "2/2/2016 12:29:11 PM"  "feature\_license\_file\_id": 1  } | | | |
| Response:  {  "timestamp": "1456802441753"  } | | | |

#### Connection check

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/connection/check | GET | NO | Check if Server is up |
| Request:  None | | | |
| Response:  Will return server current timestamp in ms  {"timestamp":1464104785869}  HTTP status code shall be 200 - OK | | | |

#### Upload log file

This is not totally JSON based API, in order to upload binary data; we’ll use multipart media type instead of JSON.

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/device/log/upload | POST | YES | Upload device log file to Server |
| Request:  (This is the JSON message in the Form Data, Server can use "JSONMsg" to get it. For binary data, Server can use "content" to get the uploaded file stream )  {  "device\_type\_guid": "3B682913-6D1E-4355-9E48-208EB7061A3D",  "serial\_number": "FT192DSAF23",  "md5": "43f84168f65a1977e0169f06dc1edc6f",  "signature": "DEADBEEF",  "chunk\_file\_index": "0",  "chunk\_file\_size": 1050,  "chunk\_size": 10000,  "original\_file\_name": "existing\_activity\_type.txt",  "original\_file\_md5": "ac76a408d45bfd0364c8593a8298ff65",  "split": true,  "chunk\_count": "2",  "schedule\_job\_GUID": "guid",  "overwrite": true  } | | | |
| Response:  {  "timestamp": 1464377652526,  "chunk\_received": [  0,  1  ],  "committed": true, // only set true if all chunks are received and log file has been merged  "message": null  }  If checksum verification failed, could return  {  "timestamp": 1464377743630,  "chunk\_received": [  1,  0  ],  "committed": false,  "message": "Checksum Verification Failure"  } | | | |
| Note: the schedule\_job\_GUID is only for Common Client when the log file is uploaded by schedule, not manually | | | |

#### Download file

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/download | POST | YES | Download file range |
| Request:  The data range in the http header. For example, ask for second 1024 bytes of file, it will be Range: bytes=1024-2047. To make Server easier to implement, Agent will only ask for downloading the whole file from a certain start position. So all requests will be Range: bytes=x-, x is the start position and no end position. | | | |
| Request:  {  "id" : 1,  "file\_name" : “file name”,  "type" : "SOFTWARE"  }  Response:  {  Use Multipart Form Data to return the stream. Set field key name to “content”.  }  If something is wrong, response’s content-type is still application/json with API failure info, for example:  {  "timestamp": 1460495444152,  "code": 401,  "msg": "Unauthorized",  "detail": "2291463f-7df6-4871-bf27-bd671a117f35"  }  HTTP status code is not 200 either.  If everything ok, return 200  If range is invalid (including case like requested start pos > actual file size), return 400  If file is not found, return 404  If IO exception, return 500  Refer to download request handler - com.medtronic.gdmp.server.rest.resources.CommonResourceHandler | | | |

#### Download agent permanent certificate

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/certificate | GET | YES | Get permanent certificate for agent |
| Request:  None | | | |
| Response:  {  "ssl\_certificate": "Base64 string of X.509 certificate",  "ssl\_certificate\_key": "Base64 string of private key",  "crypto\_certificate": "Base64 string of X.509 certificate",  "crypto\_certificate\_key": "Base64 string of private key"  } | | | |
| Certificate sample (X.509 certificate):  -----BEGIN CERTIFICATE-----  MIIDDTCCAfUCCQCOQipFgALmWDANBgkqhkiG9w0BAQUFADCBiTELMAkGA1UEBhMC  Q04xETAPBgNVBAgMCFpoZWppYW5nMREwDwYDVQQHDAhIYW5nemhvdTESMBAGA1UE  CgwJSW5zaWdtYXVzMQwwCgYDVQQLDANEZXYxEDAOBgNVBAMMB0dXLVJvb3QxIDAe…  -----END CERTIFICATE-----  Private Key sample (PKCS#1 format):  -----BEGIN RSA PRIVATE KEY-----  MIICXAIBAAKBgQDJG0jtwIVwGe8TWkW9RtNynJA3N2g8JWfrHjF4aWl/gvI1gzyv  B3lV+eVtjjpTbOE2sqiS9vcgIfJ0CcRqV09OQLglOv0Oj3K38gj0WUXkYr/Z83cm  yPNy/3ntho54SJPyFq51tyn7lEOh25ICh8uM/pkbikga7Vt9wwObBEFVBQIDAQAB  AoGAEuvEQso4Q8HM4o89vH8rab8dOnS2E+KZEl/225NfX6XcT/iW/wi++73VPieB  iVPXJKROg/VEbvuCqjQ5DKBGT7Wv4IdEHlH8T7tWrDGpJcvyvJVaMeGOJLBpFD…  -----END RSA PRIVATE KEY----- | | | |

### Server APIs for Common Client

#### Get country list

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/countries | GET | YES | Get all countries |
| Request:  None | | | |
| Response:  {  "timestamp": 1464106735304,  "country\_list": [{  "code": "CA",  "name": "Canada"  }, {  "code": "ST",  "name": "Sao Tome and Principe"  }, {  "code": "VE",  "name": "Venezuela, Bolivarian Republic of"  }, {  "code": "GW",  "name": "Guinea-Bissau"  }]  } | | | |

#### Get facility list

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/facilities | GET | YES | Get all facilities that associated to current user |
| Request:  None | | | |
| Response:  {  "timestamp": 1464106739719,  "facility\_list": [{  "name": "facility1",  "id": 1,  "address": "1 one st, one city, one state"  }, {  "name": "facility2",  "id": 2,  "address": "2 two st, one city, one state"  }]  } | | | |

#### Get device list

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/facility/{facility\_id}/devices | GET | YES | Get all devices that associated to the facility |
| Request:  None | | | |
| Response:  {  "timestamp": 1464108585037,  "devices": [{  "serial\_number": "35B1500991",  "device\_type": "ValleyLab LS10",  "CoT": "Vessel Sealing"  }, {  "serial\_number": "35B1500228",  "device\_type": "ValleyLab LS10",  "CoT": "Vessel Sealing"  }]  } | | | |

#### Get device type list

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/devicetypes | POST | YES | Get all device types that those facilities actually have devices. |
| Request:  {  "facility\_id\_list": [1, 2]  } | | | |
| Response:  {  "timestamp": 1466446814763,  "device\_type\_list": [  {  "CoT": "Vessel Sealing",  "device\_type": "Valleylab LS10",  "device\_type\_guid": "3B682913-6D1E-4355-9E48-208EB7061A3D",  "toggles": {  "FEATURE\_LICENSE\_AUTO\_UPDATE": true,  "DEVICE\_LOG\_AUTO\_CLEANUP": false  }  }  ]  }  Check com.medtronic.gdmp.common.domain.biz.DeviceTypeConfigurationToggle for a list of toggles names | | | |

#### Get software list

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/{device\_type\_guid}/software | GET | YES | Get all software that associated to the device type |
| Request:  None | | | |
| Response:  [see the software list in the response of get HW/SW configuration](#_6.4_Get_Hardware/software)  {  "timestamp": 1464968798094,  "software\_list": [  {  "name": "LS10\_SW\_1",  "type": "Software Bundle",  "md5": "AAAAAAAAAAAAAAAAA",  "status": "In Production",  "fileId": 1,  "expiration": null,  "hash": "c4ca4238a0b923820dcc509a6f75849b",  "language": "eng",  "part\_number": "PS01",  "revision": "0.1",  "file\_size": 1024,  "comparison\_order": 1,  "regulatory\_exclusion": [  "IR",  "IL",  "KP"  ]  },  {  "name": "LS10\_SW\_8",  "type": "Software Bundle",  "md5": "RRRRRRRRRRRRRRRRRR",  "status": "In Production",  "fileId": 34,  "expiration": null,  "hash": "6f4922f45568161a8cdf4ad2299f6d23",  "language": "eng",  "part\_number": "PS08",  "revision": "0.1",  "file\_size": 1024,  "comparison\_order": 1,  "regulatory\_exclusion": [  "IR",  "IL",  "KP"  ] }  ] | | | |

#### Download feature license

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/{device\_type\_guid}/{serial\_number}/license | GET | YES | Get latest feature license of a device |
| Request:  None | | | |
| Response:  {  "timestamp": "1456802441753",  "device\_type": "Valleylab LS10",  "serial\_number": "35B1500228",  "license\_file\_timestamp": "as CC only want latest license file",  "md5": "md5 string of license file",  "applied": "true or false",  "fileSize": 10240,  "fileId": "the feature license file url on Server "  }  Server probably should mark the license file as applied when get license upgrade ACK  Server probably should mark the license file as not applied when system creates license file with new sales data.  Sample response:  {  "timestamp": 1464980217129,  "license": {  "md5": "FFFFFFFFFFFFFFFFF",  "applied": true,  "fileId": 21,  "device\_type": "Valleylab LS10",  "serial\_number": "FT192DSAF23",  "license\_file\_timestamp": 1462946306000  }  } | | | |

#### Download feature license for facilities

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/device/licenses | POST | YES | Get latest feature licenses of facilities |
| Request:  Provide facility id list in request  {  "facility\_id\_list": [  1, 2  ]  } | | | |
| Response:  {  "timestamp": "1456802441753",  "license\_list": [{  "device\_type": "SCD 700",  "serial\_number": "12345",  "file\_time\_stamp": "as CC only want latest license file",  "md5": "md5 string of license file",  "applied": "true or false",  "fileSize": 10240,  "uri": "the feature license file url on Server "  }, {  "device\_type": "SCD 700",  "serial\_number": "678910",  "file\_time\_stamp": "as CC only want latest license file",  "md5": "md5 string of license file",  "applied": "true or false",  "fileSize": 10240,  "uri": "the feature license file url on Server "  }]  }  As discuss on 02/01/2016 feature license meeting, license will be sold to hospital and Server probably should get all licenses of hospital that the facility belongs to.  Sample response:  {  "problem\_sn": [  "FT192DSAF24"  ],  "problem\_ref": [  "11cbd266-8c07-4434-ae2d-86003f328875"  ],  "license\_list": [  {  "md5": "FFFFFFFFFFFFFFFFF",  "applied": true,  "fileId": 21,  "device\_type": "Valleylab LS10",  "serial\_number": "FT192DSAF23",  "license\_file\_timestamp": 1462946306000  }  ]  }  If any device has problem in getting feature license, the response will provide their SN and exception UUID for troubleshooting. | | | |

#### Schedule Log Retrieve Acknowledge

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/device/schedule/log | PUT | YES | Schedule Job Acknowledge for retrieving device log file |
| Request:  {  "device\_type\_guid": "3B682913-6D1E-4355-9E48-208EB7061A3D ",  "serial\_number": "35B12P3001",  "timestamp": "1456802441753",  "timezone": "UTC+8",  "detail": "run at 2/2/2016 12:29:11 PM once",  "localtime": "2/2/2016 12:29:11 PM"  "job\_guid": "Vital Sync Schedule Job GUID"  } | | | |
| Response:  {  "timestamp": "1456802441753"  } | | | |

#### Schedule Software Upgrade Acknowledge

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/device/schedule/software | PUT | YES | Schedule Job Acknowledge for upgrading software |
| Request:  {  "device\_type\_guid": "3B682913-6D1E-4355-9E48-208EB7061A3D ",  "serial\_number": "35B12P3001",  "timestamp": "1456802441753",  "timezone": "UTC+8",  "detail": "run at 2/2/2016 12:29:11 PM once",  "localtime": "2/2/2016 12:29:11 PM"  "software": {  "name": "Control",  "part\_number": "0",  "revision": "01.08.01"  }  "job\_guid": "Vital Sync Schedule Job GUID"  } | | | |
| Response:  {  "timestamp": "1456802441753"  } | | | |

#### Schedule Latest Software Upgrade Acknowledge

|  |  |  |  |
| --- | --- | --- | --- |
| API End Point | HTTP Method | Authenticate | Purpose |
| /medtronic/rest/agent/api/device/schedule/latestsoftware | PUT | YES | Schedule Job Acknowledge for upgrading latest software |
| Request:  {  Same as schedule software upgrade acknowledge  } | | | |
| Response:  {  "timestamp": "1456802441753"  } | | | |