

EyeHub: Service APIs

Technology Strategy Board (TSB) Internet Of Things (IoT) Demonstrator

Prepared by Flexeye Technology Ltd
Draft Version: 2.2

Table of Contents

- [1. Hub Entities](#)
 - [1.1 Hub Entity Types](#)
 - [1.2 Hub Entity Objects](#)
 - [1.2.1 Device Manager Object](#)
 - [1.2.1.1 DM Permission Object](#)
 - [1.2.1.2 DM Permission Event Object](#)
 - [1.2.2 Location Manager Object](#)
 - [1.2.3 Person Manager Object](#)
 - [1.2.4 Device Object](#)
 - [1.2.4.1 Device Tag Object](#)
 - [1.2.4.2 Device Permission Object](#)
 - [1.2.4.3 Device IO Event Object](#)
 - [1.2.4.4 Device Update Event Object](#)
 - [1.2.5 Location Object](#)
 - [1.2.6 Person Object](#)
 - [1.2.7 Application \(Service\) Object](#)
 - [1.2.8 Timer Object](#)
 - [1.2.9 Notification Object](#)
 - [1.2.10 Subscription Object](#)
 - [1.2.11 Personal Device Object](#)
 - [1.2.12 Location Tag Object](#)
- [2. Common Behaviours](#)
 - [2.1 Imperative Terms In Specifications](#)
 - [2.2. Transport Protocol](#)
 - [2.3. Media Types](#)
 - [2.4. Rest API Interface URI](#)
 - [2.5. Requests](#)
 - [2.6 Standard HTTP Status Codes](#)
 - [2.7 Interaction Diagram](#)
- [3. Service APIs](#)
 - [3.1 Hub](#)
 - [3.1.1 Authenticate](#)
 - [3.1.2 Login](#)
 - [3.1.3 Logout](#)
 - [3.1.4 Get People](#)
 - [3.1.5 Create Person](#)

- [3.1.6 Get Locations](#)
 - [3.1.7 Create Location](#)
 - [3.1.8 Get Events](#)
 - [3.1.9 Log Event](#)
 - [3.1.10 Get Timers](#)
 - [3.1.11 Start Timer](#)
 - [3.1.12 Register Device Manager](#)
 - [3.1.13 Deregister Device Manager](#)
 - [3.1.14 Register Application](#)
 - [3.1.15 Deregister Application](#)
 - [3.1.16 Register Location Manager](#)
 - [3.1.17 Deregister Location Manager](#)
 - [3.1.18 Register Person Manager](#)
 - [3.1.19 Deregister Person Manager](#)
 - [3.1.20 Subscribe](#)
 - [3.1.21 Unsubscribe](#)
 - [3.2 Person Manager](#)
 - [3.2.1 Authenticate](#)
 - [3.2.2 Login](#)
 - [3.2.3 Logout](#)
 - [3.2.4 Get Users](#)
 - [3.2.5 Create User](#)
 - [3.2.5.1 Register User](#)
 - [3.3 Location Manager](#)
 - [3.3.1 Get Locations](#)
 - [3.3.2 Create Location](#)
 - [3.4 Person](#)
 - [3.4.1 Get Person](#)
 - [3.4.1.5 Get Person Last Status](#)
 - [3.4.2 Update Person](#)
 - [3.4.2.1 Change Password](#)
 - [3.4.3 Delete Person](#)
 - [3.4.4 Get Trustees](#)
 - [3.4.5 Add Trustee](#)
 - [3.4.6 Remove Trustee](#)
 - [3.4.7 Set Default Trustee](#)
 - [3.4.8 Get Locations](#)
 - [3.4.8.1 Search Campus Location](#)
 - [3.4.9 Add Location](#)
-

- [3.4.10 Remove Location](#)
 - [3.4.11 Add Device](#)
 - [3.4.12 Remove Device](#)
 - [3.4.13 Get Events](#)
 - [3.4.14 Log Input Event](#)
 - [3.4.15 Log Output Event](#)
 - [3.4.16 Start Timer](#)
 - [3.4.16.5 Get Timers](#)
 - [3.4.17 Get Notifications](#)
 - [3.4.18 Send Notification \(to one device\)](#)
 - [3.4.18.5 Send Notification \(to all devices of a users\)](#)
 - [3.4.19 Accept Trustee Invitation](#)
 - [3.4.20 Reject Trustee Invitation](#)
 - [3.4.21 Delete Notification](#)
 - [3.4.22 Safety Alert](#)
 - [3.4.23 Add Personal Device](#)
 - [3.4.24 Delete Personal Device](#)
 - [3.5.1 Get Location](#)
 - [3.5.2 Update Location](#)
 - [3.5.3 Delete Location](#)
 - [3.5.4 Get Events](#)
 - [3.5.5 Log Output Event](#)
 - [3.5.6 Log Input Event](#)
 - [3.6 Device Manager](#)
 - [3.6.1 Get Device Manager](#)
 - [3.6.2 Update Device Manager](#)
 - [3.6.3 Get Devices](#)
 - [3.6.4 Create Device](#)
 - [3.6.5 Get Events](#)
 - [3.6.6 Device IO Event \(In\)](#)
 - [3.7 Device](#)
 - [3.7.1 Get Device](#)
 - [3.7.2 Update Device](#)
 - [3.7.3 Delete Device](#)
 - [3.7.4 Get Sub Devices](#)
 - [3.7.5 Get Events](#)
 - [3.7.6 Device Output Event](#)
 - [3.7.7 Device Input Event](#)
 - [3.7.8 Permit Devices](#)
-

[3.8 Application \(Service\)](#)

[3.8.1 Get Application](#)

[3.8.2 Register Device](#)

[3.8.3 De-Register Device](#)

[3.8.4 Get Events](#)

[3.8.5 Get Users](#)

[3.8.6 Add User](#)

[3.8.7 Authenticate User](#)

[3.8.8 Login User](#)

[3.8.9 Logout User](#)

[3.9 Timer](#)

[3.9.1 Get Timer](#)

[3.9.2 Pause Timer](#)

[3.9.3 Resume Timer](#)

[3.9.4 Stop Timer](#)

[3.9.5 Delete Timer](#)

[3.9.6 Expire Timer](#)

[3.9.7 Update Timer](#)

[3.10 Notification](#)

[3.10.1 Get Notification](#)

[3.10.2 Delete Notification](#)

[4. Appendices](#)

[4.1 This Document](#)

[4.1.1 Audience](#)

[4.1.2 Scope](#)

[4.1.3 Document Purpose](#)

[4.2 Notification Message Templates - myGuardian applications.](#)

[4.3 Administrator Signatures](#)

[4.4 Interim Collection Signatures](#)

[4.4.1 Search Locations](#)

[4.4.2 Search User Trustees](#)

[4.4.3 Search User Notifications](#)

[4.4.4 Search User Locations](#)

[4.4.5 Search Application Users](#)

[4.5 Team](#)

[4.6 Version Control](#)

1. Hub Entities

1.1 Hub Entity Types

A Hub Entity is the EyeHub digital representation of an IoT entity, which may represent a device, location, person, application or any other “thing” and is identified by URI.

1.2 Hub Entity Objects

This section defines the common elements of the Hub Entity Objects and other API parameters(e.g. Timers)

These properties are common amongst all entity objects:

Property Name	Definition	Property Value
"uri"	Uniform Resource Identifier for the entity item within the Hub	URI as a JSON string
"inEvents"	Array of input Events	JSON array of zero or more Event objects
"outEvents"	Array of output Events	JSON array of zero or more Event objects
"inOutEvents"	Entire array of input and output Events, including events occurring on the sub resources, e.g. if the resource is a Device, this property store the events for the Device and all it's sub devices. (GET requests only)	JSON array of zero or more Event objects
"represents"	A device that exists within a location, e.g. a sensor device on the wall of a room location, has a "represents" relationship to the location.	JSON Resource object
"representedBy"	A location that has one or more devices that exist within, e.g. a room location which has many fixed sensor devices, has a "representedBy" relationship to	JSON array of zero or more Resource objects

	multiple devices.	
"server"	The resource hostname if it has a reachable endpoint.	JSON string
"scheme"	E.g. "http". Used in conjunction with server if the resource has a reachable endpoint.	JSON string
"port"	Used in conjunction with server if the resource has a reachable endpoint.	JSON string
"path"	The path to an API library, where applicable, for a resource.	JSON string
"apiCallBkFunc"	The API function to be used for a callback from the hub to a resource.	JSON string
"mySubs"	List of Subscriptions the entity holds	JSON array of zero or more Subscription objects
"imageUrl"	URL endpoint to an image representing the resource.	JSON string

1.2.1 Device Manager Object

A Device Manager object is a JSON object.

A Device Manager object **MUST** contain the following properties:

Property Name	Definition	Property Value
"uri"	Uniform Resource Identifier for the Device Manager item within the Hub	URI as a JSON string

A Device Manager object **MAY** contain the following properties:

Property Name	Definition	Property Value
"ownerUri"	Uniform Resource Identifier for the	URI as a JSON string

	Device Manager item endpoint on the Owner side	
"description"	Text description of the Device Manger item	JSON string
"isRegistered"	Enumerated string value that defines the registered status of Device Manager= {(1,TRUE),(0, FALSE)}	JSON string
"dateRegistered"	Date when the Device Manger was registered with the Hub	ISO 8601 date as JSON string
"lastSynchronized"	Most recent date of all devices' events	ISO 8601 date as JSON string
"devices"	List of Device items managed	JSON array of zero or more Device objects
"managers"	List of Location Managers	JSON array of zero or more Location Manager objects
"permissions"	List of permissions for all managed Devices - defines Allow and/or Deny rules that control what Devices can <i>enter</i> the Hub.	JSON array of DM Permission objects
"activeDevices"	Number of active devices managed by Device Manager (GET requests only)	JSON Number
"totalDevices"	Total number of devices managed by Device Manager (GET requests only)	JSON Number
"numEvents"	Total number of events managed by Device Manager (GET requests only)	JSON Number

The same Device URI MUST NOT appear more than once in the same Device Manager's devices list, i.e. a set of distinct URIs.

1.2.1.1 DM Permission Object

A DM Permission object is a JSON object.

A DM Permission object MUST contain the following properties:

Property Name	Definition	Property Value
"type"	Enumerated string value that defines the type of permission = {ALLOW, DENY}	JSON string
"devices"	List of Device items that have the permission defined permission. Wildcarding or groups of devices are fine. But we need to "identify" the devices that are allowed.	JSON array of one or more Device ids, which may include wildcard characters.

1.2.1.2 DM Permission Event Object

A DM Permission Event object is a JSON object, which encapsulates an update to a set of permissions for one or more Devices managed by a Device Manager.

Note: this event is used to incrementally update a set of permissions - so overlay/merge rules apply with respect to the complete set of Device permissions for a Device Manager.

A DM Permission Event object MUST contain the following properties:

Property Name	Definition	Property Value
"datetime"	Date and time of the input/output (I/O) event when originated from the source. E.g. "2013-06-11T13:15:30Z"	ISO 8601 date as JSON string
"permissions"	List of permissions for all managed Devices - defines Allow and/or Deny rules that control what Devices can <i>enter</i> the Hub.	JSON array of DM Permission objects

.B. I/O permissions are at Device level - the DM-level just controls which Devices can **enter**.

1.2.2 Location Manager Object

A Location Manager object is a JSON object.

A Location Manager object **MUST** contain the following properties:

Property Name	Definition	Property Value
"uri"	Uniform Resource Identifier for the Location Manager item within the Hub	URI as a JSON string

A Location Manager object **MAY** contain the following properties:

Property Name	Definition	Property Value
"name"	Name of the Location Manager	JSON string
"ownerUri"	Uniform Resource Identifier for the Location Manager item endpoint on the Owner side	URI as a JSON string
"isRegistered"	Enumerated string value that defines the registered status of Location Manager= {TRUE, FALSE} Enum value for FALSE is "0" and TRUE is "1"	JSON string
"dateRegistered"	Date of the Location Manager registration.	ISO 8601 date as JSON string
"lastSync"	Most recent date of all locations' events	Date value in restricted ISO 8601 format as JSON string
"locations"	List of Location items managed	JSON array of zero or more Location objects
"managers"	List of Location Managers	JSON array of zero or more Location Manager objects
"permissions"	List of permissions for all managed Locations - defines Allow and/or Deny rules that control what Locations can <i>enter</i> the Hub.	JSON array of DM Permission objects

"activeLocations"	Number of active locations managed by Location Manager (GET requests only)	JSON Number
"totalLocations"	Total number of locations managed by Location Manager (GET requests only)	JSON Number

The same Location URI MUST NOT appear more than once in the same Location Manager's locations list, i.e. a set of distinct URIs.

1.2.3 Person Manager Object

A Person Manager object is a JSON object.

A Person Manager object MUST contain the following properties:

Property Name	Definition	Property Value
"uri"	Uniform Resource Identifier for the Person Manager item within the Hub	URI as a JSON string

A Person Manager object MAY contain the following properties:

Property Name	Definition	Property Value
"name"	Name of Person Manager item	JSON string
"ownerUri"	Uniform Resource Identifier for the Person Manager item endpoint on the Owner side	URI as a JSON string
"isRegistered"	Enumerated string value that defines the registered status of Person Manager= {TRUE, FALSE} Enum value for FALSE is "0" and TRUE is "1"	JSON string
"dateRegistered"	Date of Person Manager registration.	ISO 8601 date as JSON string

"pmUsers"	List of People/Users managed by Person Manager Item	JSON array of zero or more User/Person objects
"managers"	List of Location Managers	JSON array of zero or more Location Manager objects
"permissions"	List of permissions for all managed People - defines Allow and/or Deny rules that control which people can <i>enter</i> the Hub.	JSON array of PM Permission objects
"numUsers"	Total number of users managed by Person Manager (GET requests only)	JSON Number

The same Person URI MUST NOT appear more than once in the same Person Manager's people list, i.e. a set of distinct URIs.

1.2.4 Device Object

A Device object is a JSON object.

A Device object MUST contain the following properties:

Property Name	Definition	Property Value
"uri"	Uniform Resource Identifier for the Device item within the Hub.	URI as JSON string

A Device object MAY contain the following properties:

Property Name	Definition	Property Value
"name"	Name of Device object	JSON string
"description"	Defines description of the Device item	JSON string
"deviceOwner"	Defines owner of the Device	JSON Person object

"entryDate"	Registration date of the device	ISO 8601 date as JSON string
is		
"deviceType"	Device Type(Sensor, Multi-Sensor) managed by the Device	JSON Device Type object
"lastKnownLocation"	Recently used location source Ex: CCSR / 20BA02	JSON string
"lastActiveDate"	Last date the Device had an event	ISO 8601 date as JSON string
"dms"	Unique identifier for the Device item within the scope of the Device Manager - defined by the Owner.	JSON string
"deviceTags"	An array of Device Tag objects describing the Device.	JSON array of Device Tag objects
"location"	Location managed by the Device	JSON Location object
"managers"	List of Location Managers	JSON array of zero or more Location Manager objects
"subDevices"	List of sub-Devices that are composited by this Device	JSON array of zero or more Device objects
"subscribed"	List of names whose values indicate whether there are subscribers to the Device in the Hub and what events they subscribe to. The value "NONE" is used to indicate no subscribers. The value "ALL" is used to indicate there are one or more subscribers to every event supported by the Device. The value can be a link to a list of	JSON array of one or more JSON strings The values "ALL" and "NONE" (not case-sensitive) are reserved words. Remaining values are dependent on the particular Device I/O interface.

	subscribers.	
"permissions"	List of permissions for the Device - defines Allow and or Deny rules that control access to the I/O interface.	JSON array of Device Permission objects
"ownerParentId"	Unique identifier for the parent Device item within the scope of the Device Manager - defined by the Owner.	JSON string
"numEvents"	Total number of events managed by Device (GET requests only)	JSON Number
"numEventsAll"	Total number of events generated by Device and Hub (GET requests only)	JSON Number
"numSubDevices"	Total number of sub Devices managed by Device (GET requests only)	JSON Number

1.2.4.1 Device Tag Object

A Device Tag object is a JSON object.

A Device Tag is contained by (private to) a Device.

A Device Tag object MUST contain the following properties:

Property Name	Definition	Property Value
"rel"	A relationship between the Device and a target classification/categorisation entity. E.g. "http://purl.oclc.org/NET/ssnx/ssn#SensingDevice".	URI of relationship as JSON string
"val"	The entity to which the <code>rel</code> property applies. E.g. "http://dbpedia.org/page/Photoresistor".	URI of entity as JSON string

1.2.4.2 Device Permission Object

A Device Permission object is a JSON object.

A Device Permission object MUST contain the following properties:

Property Name	Definition	Property Value
"iotype"	Enumerated string value that defines the type of I/O permission = {NONE, INPUT_ONLY, OUTPUT_ONLY, INPUT_AND_OUTPUT}	JSON string
"iosubscribers"	List of subscribers that have the permission defined permission.	JSON array of one or more subscriber ids.

1.2.4.3 Device IO Event Object

A Device IO Event object is a JSON object.

A Device IO Event object MUST contain the following properties:

Property Name	Definition	Property Value
"timeStamp"	Date and time of the input/output (I/O) event when originated from the source. E.g. "2013-06-11T13:15:30Z"	Date value in restricted ISO 8601 format* as JSON string
"payload"	The data pertaining to the Device I/O event.	Event payload as JSON string

* For details of the restricted ISO 8601 format see note <http://www.w3.org/TR/NOTE-datetime>

A Device IO Event object MAY contain the following properties:

Property Name	Definition	Property Value
"id"	Unique Identifier for the Event	JSON string
"source"	Defines the path when event is originated from the source.	JSON String

"type"	Enumerated string value that defines the type of the event = {(0,Output),(1,Input),(2,Both)}	Name of event type as JSON string
--------	--	-----------------------------------

1.2.4.4 Device Update Event Object

A Device Update Event object is a JSON object.

A Device Update Event object MUST contain the following properties:

Property Name	Definition	Property Value
"datetime"	Date and time of the expect event when originated from the source. E.g. "2013-06-11T13:15:30Z"	Date value in restricted ISO 8601 format* as JSON string

* For details of the restricted ISO 8601 format see note <http://www.w3.org/TR/NOTE-datetime>

A Device Update Event object MUST contain **one** or more (and MAY contain all) of the following properties:

Property Name	Definition	Property Value
"subscribed"	List of names whose values indicate whether there are subscribers to the Device in the Hub and what events they subscribe to. The value "NONE" is used to indicate no subscribers. The value "ALL" is used to indicate there are one or more subscribers to every event supported by the Device.	JSON array of one or more JSON strings The values "ALL" and "NONE" (not case-sensitive) are reserved words. Remaining values are dependent on the particular Device I/O interface.
"permissions"	List of permissions for the Device - defines Allow and or Deny rules that control access to the I/O interface.	JSON array of Device Permission objects

1.2.5 Location Object

A Location object is a JSON object.

A Location object **MUST** contain the following properties:

Property Name	Definition	Property Value
"uri"	Uniform Resource Identifier for the Location item within the Hub.	URI as JSON string

A Location object **MAY** contain the following properties:

Property Name	Definition	Property Value
"name"	Name of Location object	JSON string
"timezone"	Defines time zone used for the Location	JSON String
"qrCode"	QR Code identifies the type of data it holds about the location.	JSON String
"lms"	Unique identifier for the Location item within the scope of the Location Manager - defined by the Owner.	JSON string
"dateRegistered"	Stores location registration date	ISO 8601 date as JSON string
"lastSync"	Date of the last location event.	ISO 8601 date as JSON string
"managers"	List of Location Managers	JSON array of zero or more Location Manager objects
"subLocations"	List of sub-Locations that are composited by this Location	JSON array of zero or more Location objects
"devices"	List of devices that belong to the location	JSON array of zero or more Device objects
"subscribed"	List of names whose values indicate	JSON array of one or more JSON

	<p>whether there are subscribers to the Location in the Hub and what events they subscribe to.</p> <p>The value "NONE" is used to indicate no subscribers.</p> <p>The value "ALL" is used to indicate there are one or more subscribers to every event supported by the Location.</p> <p>The value can be a link to a list of subscribers.</p>	<p>strings</p> <p>The values "ALL" and "NONE" (not case-sensitive) are reserved words. Remaining values are dependent on the particular Location I/O interface.</p>
"ownerParentId"	Unique identifier for the parent Location item within the scope of the Location Manager - defined by the Owner.	JSON string
"activeDevices"	Number of active devices managed by Location (GET requests only)	JSON Number
"activeSubLoc"	Number of active sub locations managed by Location (GET requests only)	JSON Number
"numDevices"	Total number of devices managed by Location (GET requests only)	JSON Number
"numEvents"	Total number of events managed by Location (GET requests only)	JSON Number
"numSubLocDevices"	Total number of sub location devices managed by Location (GET requests only)	JSON Number
"numSubLocSubDevices"	Total number of sub location's sub devices managed by Location (GET requests only)	JSON Number

"numSubLocations"	Total number of sub locations managed by Location (GET requests only)	JSON Number
"tags"	List of tags that exist within the location.	JSON array of zero or more Location Tag objects
"addr_building"	Address of the nearest building/landmark to the location	JSON String
"addr_road"	Defines the road number / street name of the location	JSON String
"addr_other"	Defines other details that identifies the location ex: country code,IP Address,Network operator	JSON String
"addr_town"	Defines the town name of the location	JSON String
"addr_country"	Defines the country name of the location	JSON String
"addr_postcode"	Defines the postal code/zip code of the location	JSON String
"lat"	Defines the latitude value of the location	JSON Number
"long"	Defines the longitude value of the location	JSON Number

1.2.6 Person Object

A Person object is a JSON object.

A Person object MUST contain the following properties:

Property Name	Definition	Property Value
"id"	Unique Identifier for the Person item within the Hub.	JSON string
"firstName"	Defines the first name of the person	JSON string
"lastName"	Defines the last name of the person	JSON string
"emailId"	Defines the email address of the person	JSON string
"password"	Defines the password of the person	JSON string

A Person object MAY contain the following properties:

Property Name	Definition	Property Value
"name"	Name of Person	JSON string
"pmRef"	Unique identifier for the Person item within the scope of the Person Manager - defined by the Owner.	JSON string
"address"	Defines the address of the person	JSON string
"gender"	Enumerated string value that defines the gender of the Person = {(0,Male), (1,Female)}	JSON string
"phoneNumber"	Defines contact number of the person	JSON string
"trustees"	List of trustees managed by the user	JSON array of zero or more trustee objects.
"defaultTrustee"	Default trustee managed by the user	JSON Person object

"devices"	List of devices from which events are generated	JSON array of zero or more Device objects.
"timers"	List of timers associated with the person	JSON array of zero or more timer objects.
"locations"	List of locations managed by the user	JSON array of zero or more Locations objects.
"notifications"	List of notifications generated by the user	JSON array of zero or more Notification objects
"trusteeInvitations"	List of trustee invitations managed by the user	JSON array of one or more Person objects
"rejectedTrustees"	List of rejected trustees managed by the user	JSON array of one or more Person objects
"type"	Person Type(Staff, Student) managed by the user	JSON Person Type object
"subscribed"	<p>List of names whose values indicate whether there are subscribers to the person in the Hub and what events they subscribe to.</p> <p>The value "NONE" is used to indicate no subscribers.</p> <p>The value "ALL" is used to indicate there are one or more subscribers to every event supported by the Person.</p> <p>The value can be a link to a list of subscribers.</p>	<p>JSON array of one or more JSON strings</p> <p>The values "ALL" and "NONE" (not case-sensitive) are reserved words. Remaining values are dependent on the particular Person I/O interface.</p>

1.2.7 Application (Service) Object

An Application (Service) object is a JSON object.

An Application (Service) object MUST contain the following properties:

Property Name	Definition	Property Value
"id"	Unique Identifier for the Application (Service)	JSON string

An Application (Service) object MAY contain the following properties:

Property Name	Definition	Property Value
"name"	Name of Application (Service)	JSON string
"description"	Defines description of the Application (Service)	JSON string
"users"	List of users who uses the Application (Service)	JSON array of zero or more user objects.
"managers"	List of Location Managers	JSON array of zero or more Location Manager objects
"isRegistered"	Enumerated string value that defines the Application registered status= {TRUE, FALSE} Enum value for FALSE is "0" and TRUE is "1"	JSON string
"dateRegistered"	Application registration date	ISO 8601 date as JSON string
"devices"	List of Device items registered to the Application (Service)	JSON array of zero or more references to Device objects
"numUsers"	Total number of users managed by Application (GET requests only)	JSON Number
"notifications"	List of notifications that exist across all of the users of the application.	JSON array of zero or more Notification objects.

1.2.8 Timer Object

A Timer object is a JSON object.

A Timer object **MUST** contain the following properties:

Property Name	Definition	Property Value
"duration"	The number of seconds the timer will tick (run for), if not interrupted by a status changing operation, until it changes to the expired state.	JSON Number

A Timer object **MAY** contain the following properties:

Property Name	Definition	Property Value
"id"	Unique Identifier for the Timer object	JSON string
"notifyCoverage"	Enumerated string value that defines the coverage of notification sent when a timer's state changes to expired. Possible values: {(1,DefaultTrustee), (2,AllTrustees)}.	JSON string
"startTime"	The Timer start time E.g. "2013-06-11T13:15:30Z"	Date value in restricted ISO 8601 format* as JSON string
"state"	Enumerated string value that defines the state of timer= {(0,Ticking), (1,Paused), (2,Stopped), (3,Expired)}	JSON string
"lastRestartTime"	The Timer resume time E.g. "2013-06-11T16:45:20Z"	Date value in restricted ISO 8601 format* as JSON string
"lat"	The last recorded latitude of the participant of the timer.	JSON Number
"long"	The last recorded longitude of the participant of the timer.	JSON Number

"batteryLevel"	The last recorded device battery level of the participant of the timer.	JSON Number
"defaultTrustee"	The default trustee for the timer object. This defaults to the default trustee of the timer's user, but can be set to a different value whilst creating a timer (signature 3.4.16) or after creation (signature 3.4.16.5).	JSON Person object
"trustees"	The one or many trustees associated with the timer. This defaults to the trustees of the timer's user, but can be set to different values whilst creating a timer (signature 3.4.16) or after creation (signature 3.4.16.5).	JSON Person object
"startTagLocation"	The starting tag location object for the timer.	JSON Tag Location object
"endTagLocation"	The starting tag location object for the timer.	JSON Tag Location object
"startLocation"	The starting location object for the timer. Automatically set from the Tag Location's associated Location.	JSON Location object
"endLocation"	The ending location object for the timer. Automatically set from the Tag Location's associated Location.	JSON Location object
"startGPS"	The starting GPS location from Google Maps	JSON String
"endGPS"	The ending GPS location from Google Maps	JSON String
"checkInLocation"	A location the that has been reached during the course of a trip/journey and acknowledged using a QR code or NFC Tag.	JSON Location object

"currentLocation"	The existing location of the participant of the timer.	JSON Location object
"doStartNotify"	Stores 1.0 (true) if the hub will automatically send a notification when a timer is created, otherwise store 0.0 (false) and nothing is performed by the hub.	JSON Number
"doPauseNotify"	Stores 1.0 (true) if the hub will automatically send a notification when a timer is paused, otherwise store 0.0 (false) and nothing is performed by the hub.	JSON Number
"doStopNotify"	Stores 1.0 (true) if the hub will automatically send a notification when a timer is stopped, otherwise store 0.0 (false) and nothing is performed by the hub.	JSON Number
"logMessage"	The message content that is sent when the Timer expires	JSON string

1.2.9 Notification Object

A Notification object is a JSON object.

A Notification object MUST contain the following properties:

Property Name	Definition	Property Value
"id"	Unique Identifier for the Notification object	JSON string
"medium"	Enumerated string value that defines the type of notification= {(0,Email), (1,SMS), (2,ApplePushNotification), (3,GoogleCloudMessaging)}	JSON string

"message"	Defines the Notification message	JSON string
-----------	----------------------------------	-------------

A Notification object MAY contain the following properties:

Property Name	Definition	Property Value
"timeStamp"	Date and time of the notification when originated from the source. E.g. "2013-06-11T13:15:30Z"	Date value in restricted ISO 8601 format* as JSON string
"subject"	The subject for the Notification based on the type	JSON string
"title"	The title for the Notification message.	JSON string
"notifyPerson"	The User/Person object when notification is originated from the source	JSON User object
"status"	Enumerated string value that defines the status of notification= {(0,Not Sent), (1,Sent)}	JSON string
"sender"	The text description for the sender when notification is originated from the source	JSON string
"timer"	Respective Timer object if any	JSON Timer object
"notifyType"	A string which can be used to store the type of notification. e.g. "InvitationAcceptedNotification", "TimerRunOutNotification" etc	JSON string

1.2.10 Subscription Object

A Subscription object is a JSON object.

A Subscription object MUST contain the following properties:

Property Name	Definition	Property Value
"subscribeEntity"	Defines the entity you are subscribing to.	JSON Resource Object

A Subscription object MAY contain the following properties:

Property Name	Definition	Property Value
"isForwarding"	Boolean that controls if the hub will make a callback to an end point with any events that are created. Possible values are True or False. Defaults to True.	JSON string
"type"	Enumerated string value that defines the coverage of subscription. Possible values are: 0.0: All the entities' output events, including any output events it has in it's subscribed events list. 1.0: Subscribe to only the entities direct output events. Defaults to 0.	JSON string

1.2.11 Personal Device Object

A Personal Device object is a JSON object.

A Personal Device object MUST contain the following properties:

Property Name	Definition	Property Value
"regId"	Application instance id (for APN this is a device token and for GCM this is a registration Id).	JSON string

"medium"	Enumerated string value that defines the type of notification= {(0,Email), (1,SMS), (2,ApplePushNotification), (3,GoogleCloudMessaging)}	JSON string
----------	--	-------------

A Personal Device object MAY contain the following properties:

Property Name	Definition	Property Value
"imei"	Unique identifier for a mobile device.	JSON string

1.2.12 Location Tag Object

A Location Tag object is a JSON object.

A Location Tag object MUST contain the following properties:

Property Name	Definition	Property Value
"id"	Unique identifier for the tag.	JSON string

A Location Tag object MAY contain the following properties:

Property Name	Definition	Property Value
"name"	Name of Tag.	JSON string
"lat"	Defines the latitude value of the location.	JSON Number
"long"	Defines the longitude value of the location.	JSON Number
"location"	The associated location where the tag resides.	JSON Location Object

"areaId"	The area name where the tag resides.	JSON string
----------	--------------------------------------	-------------

2. Common Behaviours

2.1 Imperative Terms In Specifications

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document, and all other documents that comprise the specification of the Hub API, are to be interpreted as described in "Key words for use in RFCs to Indicate Requirement Levels" (RFC 2119).

2.2. Transport Protocol

The Hub API is based on the Hypertext Transfer Protocol, version 1.1 (RFC 2616). Each request will be authenticated using HTTP Basic Authentication (RFC 2617). Therefore, requests sent from clients on the public Internet (and not on a secure channel such as a VPN) **MUST** use the https protocol.

2.3. Media Types

In the Hub API, entity representations and request bodies are normally encoded in JSON, as specified in RFC4267.

Each type of Entity has its own media-type, which matches the pattern `application/vnd.com.flexeye.iot.rm.v1+json`, where "iot.rm.v1" represents the portion of the identifier unique to a particular representation format for the Hub service.

The Hub API implementation **MUST** provide representations of all entities available in JSON.

The Hub API implementation **MUST** accept requests from clients encoded in JSON.

2.4. Rest API Interface URI

https://<hostname>:<port>/v1/<AppName_Default>/<Composite>/<Attribute>

2.5. Requests

2.5.1 Request Headers

The JSON request headers must have the following information to be accepted via the REST API:

Header Field Name	Header Value
Accept:	application/json
Accept-Charset:	UTF-8
Authorization:	Basic + Base 64 encode of 'username:password'
Content-Type:	application/json;charset=UTF-8

2.5.2 GET

2.5.2.1 Collections

GET accepts the following collection (array) parameters to filter the collection list:

Query Parameter Name	Parameter Value	Description
count	A positive integer.	This parameter specifies the maximum number of items to return. By default, 20 items are returned.
start	A positive integer.	This parameter specifies the item to start with in a collection.

		By default, starts at the first item in the collection.
sort	asc or dec	<p>This parameter specifies the sort order for resource items, which can be either ascending or descending by key reference.</p> <p>By default, the sort order is ascending.</p>
q	A string	<p>This parameter specifies the query string that would be used to filter a collection.</p> <p>By default, the filter returns 20 items as determined by the count parameter.</p>

Examples:

Retrieve the information from the fifth item of the list of items for an instance, where the list page items is 10:

- GET /jCountries?start=5&count=10

Retrieve information about the list of items in descending order by key:

- GET /jCountries?sort=dec

Items

GET accepts the following element parameters to filter the attributes for an item:

Query Parameter Name	Parameter Value	Description
fields	A comma-separated list of attribute names.	<p>This parameter filters the attributes that are returned.</p> <p>To include a subset of the members of an object property, include the member name (as object.member) instead of the object property name.</p> <p>By default, all properties are returned.</p>

Examples:

Retrieve the name, sum, and date attributes for the item with the 4.0 key:

- GET /jCountries/4.0?fields=name,sum,date

2.5.2.2 Collection Queries

GET can filter down results from a collection by using the query q= parameter.

The query syntax is: (<item>[Keyword]<value>)

Parentheses are optional. If used, they can determine logical groupings for (<item>[Keyword]<value>) tuples.

Below is a list of keywords that the REST API accepts:

Keyword	Description
[and]	logical and (&&)
[or]	logical or ()
[eq]	equals (==)
[gt]	greater than (>)
[ge]	greater than or equal to (>=)
[lt]	less than (<)
[le]	less than or equal to (<=)
[ne]	not equal (!=)

Examples

Retrieve information about the list of items whose sum is 1500:

- GET /jCountries?q=sum[eq]1500

Retrieve information about a list of items whose sum is greater than 1500 and not equal to NaN:

- GET /jCountries?q=(sum[gt]1500)[and](sum[ne]NaN)

Retrieve information about a list of items whose date is greater than August 31, 2013:

- GET /jCountries?q=date[gt]"2013-08-31T0:00:00.000Z"
-

Retrieve information about a list of items whose boolean value is true:

- GET /jCountries?q=bool[eq]1.0

Retrieve information about a list of items whose color is not equal to brown:

- GET /jCountries?q=color[ne]"#A52A2A"

Retrieve information about a list of items whose sum is equal to 1500 or equal to 1200:

- GET /jCountries?q=(sum[eq]1500)[or](sum[ne]1200)

Retrieve information about a list of items whose name is greater than Jamaica:

- GET /jCountries?q=Name[gt]"Jamaica"

Below is a list of String specific functions:

Function	Description
lowerCase(String)	Converts the string to lowerCase
upperCase(String)	Converts the string to upperCase
properCase(String)	Converts the string to properCase
indexOf(String,pattern)	Returns the position of the given string
left(String,index)	Returns the string from left upto number specified in index
right(String,index)	Returns the string from right upto number specified in index
substring(String,start,end)	Returns the string from start position to end position
sentenceCase(String)	Converts the string to properCase

Examples:

Retrieve information about a list of items whose name has jam:

- GET /jCountries?q=indexOf(Name,"jam")[gt]0
-

Retrieve information about a list of items whose name has JAM: This can be used in conjunction with either lowerCase or upperCase

- GET /jCountries?q=indexOf(lowerCase(Name),lowerCase("JAM"))[gt]0
- GET /jCountries?q=indexOf(upperCase(Name),upperCase("JAM"))[gt]0

Collection Query Note

The system uses **NaN** and **null** as values for items (attributes). These values may return a larger subset of results than desired. To filter out these items, make sure that you set that item to **<item>[ne]NaN** or **<item>[ne>null**.

2.5.3 POST

2.5.3.1 Operations

Operations can be run via a POST request and operations are indicated (where applicable) in section 3. If an operation is successful, POST returns a 200 (OK). The POST request does not accept a body for operations, however parameters can be passed in the query string portion of the URL e.g.

"?param1=value1¶m2=value2" below:

POST

`https://hub.flexeye.com/v1/iot_Default/some_operation?param1=value1¶m2=value2`

2.6 Standard HTTP Status Codes

Unless otherwise specified, to indicate failure, each service **MUST** return a valid HTTP status code, as specified in RFC2616, which is one of the following:

Code	Meaning
200	"OK"
201	"Created"
204	"No Content"
400	"Bad Request" (eg. malformed input)
401	"Unauthorized"
403	"Forbidden"
404	"Not Found"
500	"Internal Server Error"
501	"Not Implemented"

2.7 Interaction Diagram

Diagram below show various options for the developer applications to interact with the Hub.

Interaction between Application and EyeHub

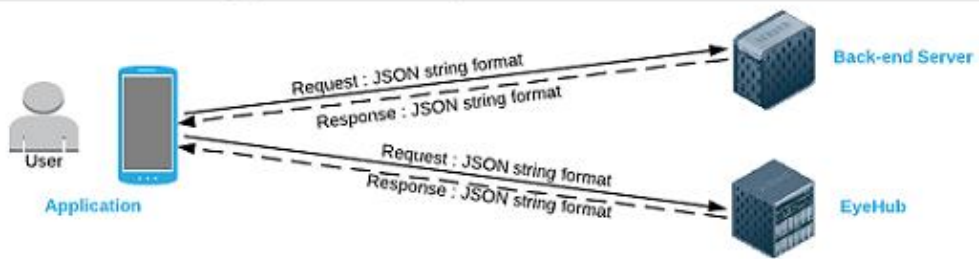


Figure 1 : Interaction between Application and EyeHub along with Back-end Server

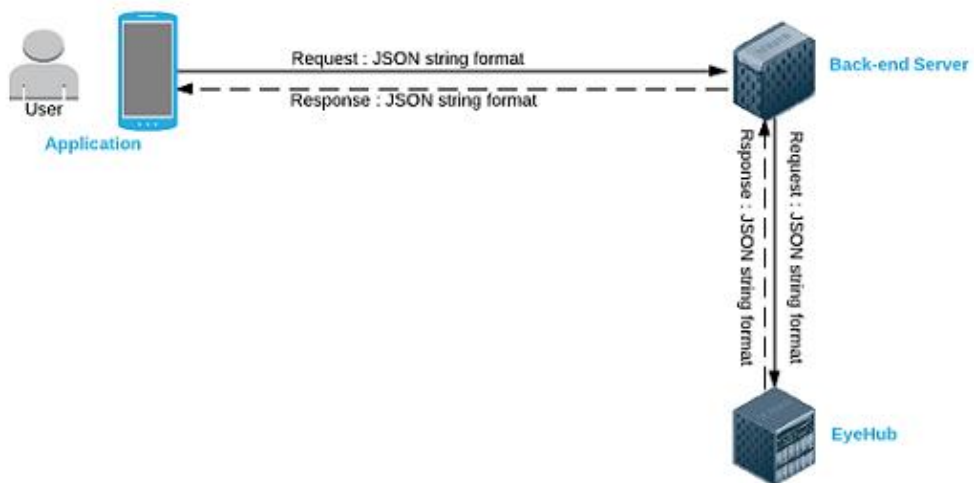


Figure 2 : Interaction between Application and EyeHub through Back-end Server

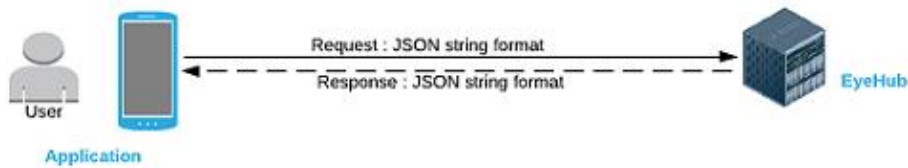


Figure 3 : Interaction between Application and EyeHub directly

3. Service APIs

3.1 Hub

3.1.1 Authenticate

Description: Authenticates a Person managed by the Hub's default Person Manager (see also *Person Manager* below).

Synopsis: POST (Operation)

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: username, password

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Authenticates a user named "user121" managed by the Hub's default Person Manager.

POST

`https://hub.flexeye.com/v1/iot_Default/authenticate?username=<username>&password=<password>`

Host: `hub.flexeye.com`

Authorization: `Basic authentication`

Accept: `application/json`

Content-Length: `nnn`

Content-Type: `application/json`

Example Response:

`HTTP/1/1 200 OK`

Content-Type: `application/vnd.com.flexeye.iot+json`

Content-Length: `nnn`

```
{
  "results" : "Authentication Successful",
  "operation" : "authenticate"
}
```

3.1.2 Login

Description: Authenticates a Person managed by the Hub's default Person Manager and establishes a session for that Person with the Hub.

N.B. Should use secure channel for passing the password, i.e. https

Synopsis: POST

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: username, password

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: Session ID

Response Status: 200, 400, 401, 403, 404

Example Request: Authenticates a user named "user121" and establishes a session.

POST

https://hub.flexeye.com/v1/iot_Default/login?username=<username>&password=<password>

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
{
  "results": "sessionID: 24e0945aad6f0869a77ccab063d7fdfd"
}
```

3.1.3 Logout

Description: Closes an open session with the Hub.

Synopsis: POST

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: sessionID

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Closes an opened session "user1" with the Hub.

POST https://hub.flexeye.com/v1/iot_Default/logout?sessionID=<sessionID>

Host: hub.flexeye.com
Authorization: Basic authentication
Accept: application/json
Content-Length: nnn
Content-Type: application/json

Example Response:

HTTP/1/1 200 OK
Content-Type: application/vnd.com.flexeye.iot+json
Content-Length: nnn

```
{  
    "results" : "True"  
}
```

3.1.4 Get People

Description: Returns a list of Persons for the Hub (across all Person Managers).

Synopsis: GET

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: If no query parameters are passed all users are returned.

Optional Query parameters: Any property listed in section 1.2.6.

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: Array of Users

Response Status: 200, 400, 401, 403, 404

Example Request: Returns a list of people in the Hub whose first name is "John"

GET https://hub.flexeye.com/v1/iot_Default/allPeople
Query: firstName=John
Host: hub.flexeye.com
Authorization: Basic authentication
Accept: N/A
Content-Length: N/A
Content-Type: N/A

Example Response:

HTTP/1/1 200 OK
Content-Type: application/vnd.com.flexeye.iot+json
Content-Length: nnn

```
[  
    {
```

```
    "allPeople" : [ {  
      "uri" :  
      "https://hub.flexeye.com/v1/iot_Default/allPeople/John_Person_1",  
      "key" : "John_Person_1"  
    }, {  
      "uri" :  
      "https://hub.flexeye.com/v1/iot_Default/allPeople/John_Person_2",  
      "key" : "John_Person_2"  
    } ],  
    "allPeopleCursor" : {  
      "count" : "0",  
      "nextUri" : null,  
      "prevUri" : null  
    }  
  }  
]
```

3.1.5 Create Person

Description: Creates a new Person managed by the Hub's default Person Manager (see also *Person Manager* below).

Synopsis: POST

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: Any property listed in section 1.2.6

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Creates a new Person named "user1" under Default Person Manager

POST https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{  
  "id" : "user1",  
  "firstName" : "fname",  
  "lastName" : "sname",  
  "emailId" : "email@mail.com",  
  "phoneNumber" : "1234567891"  
}
```

Example Response:

HTTP/1/1 201 CREATED

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
{
  "uri" :
  "https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/u
ser1",
  "id" : "user1",
  "firstName" : "fname",
  "lastName" : "sname",
  "emailId" : "email@mail.com",
  "phoneNumber" : "1234567891"
}
```

3.1.6 Get Locations

Description: Returns a list of Locations for the Hub (across all Location Managers).

Synopsis: GET

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: If no query parameters are passed all locations are returned.

Optional Query parameters: Any property listed in section 1.2.5

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: array of locations

Response Status: 200, 400, 401, 403, 404

Example Request: Returns list of all locations in the Hub

GET https://hub.flexeye.com/v1/iot_Default/allLocations

Query: addr_postcode=GU24HJ

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: N/A

Content-Length: N/A

Content-Type: N/A

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
  {
    "allLocations" : [ {
      "uri" : "https://hub.flexeye.com/v1/iot_Default/allLocations/CCSR",
      "key" : "CCSR"
    }, {
      "uri" :
"https://hub.flexeye.com/v1/iot_Default/allLocations/FlexeyeLoc",
      "key" : "FlexeyeLoc"
    } ],
    "allLocationsCursor" : {
      "count" : "0",
      "nextUri" : null,
      "prevUri" : null
    }
  }
]
```

]

3.1.7 Create Location

Description: Creates a new Location managed by the Hub's default Location Manager (see also *Location Manager* below).

Synopsis: POST

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: lat, long , addr_building, addr_road, addr_other, addr_town, addr_county, addr_postcode.

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Creates a new location named "location1" under Default Location Manager

POST: https://hub.flexeye.com/v1/iot_Default/lms/dlm/locations

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{
  "lat" : "90E",
  "long" : "87S",
  "addr_building" : "bridgehouse",
  "addr_road" : "millmead",
  "addr_other" : "Near George Abbott",
  "addr_country" : "UK",
  "addr_postcode" : "GU24HJ"
}
```

Example Response:

HTTP/1/1 201 CREATED

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
{
  "uri" :
  "https://hub.flexeye.com/v1/iot_Default/lms/dlm/locations/location1",
  "name" : "location1",
  "lat" : "90E",
  "long" : "87S",
```

```
"addr_building" : "bridgehouse",
"addr_road" : "millmead",
"addr_other" : "Near George Abbott",
"addr_country" : "UK",
"addr_postcode" : "GU24HJ"
}
```

3.1.8 Get Events

Description: Returns a list of Events for the Hub.

Synopsis: GET

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: If no query parameters are passed all events are returned.

Optional Query parameters: Any property listed in section 1.2.4.3

limit(number of events to return(default: 100, max:1000), interval (start-end date-time),

datetime(Retrieve events which occurred after/before/on the specified date,number of events),

specific source events(device ID/user ID/location ID/sub device ID/sub location ID,number of events

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: Array of Events

Response Status: 200, 400, 401, 403, 404

Example Request: Get the list of Events for the Hub.

GET: https://hub.flexeye.com/v1/iot_Default/allEvents

Query: locationID=CCSR

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: N/A

Content-Length: N/A

Content-Type: N/A

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
  {
    "allEvents" : [ {
      "uri" : "https://hub.flexeye.com/v1/iot_Default/allEvents/Event1",
      "key" : "Event1"
    }, {
      "uri" : "https://hub.flexeye.com/v1/iot_Default/allEvents/Event2",
```

```
        "key" : "Event2"
      } ],
      "allEventsCursor" : {
        "count" : "14735",
        "nextUri" :
        "https://hub.flexeye.com/v1/iot_Default/allEvents?count=20&start=20",
        "prevUri" : null
      }
    }
  ]
}
```

3.1.9 Log Event

Description: Logs a generic event for the Hub

Synopsis: POST

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: timeStamp, payload

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: logs an event at "21-Aug-2013-12:12:12"

POST: https://hub.flexeye.com/v1/iot_Default/hubEvents

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{
  "timeStamp" : "2013-11-07T16:45:46.307Z",
  "payload" : "test description"
}
```

Example Response:

HTTP/1/1 201 CREATED

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
  {
    "timeStamp" : "2013-11-07T16:45:46.307Z",
```

```
    "type" : "NaN",
    "payload" : "test description"
    "source" : ""
  ]
```

3.1.10 Get Timers

Description: Returns a list of Timers for the Hub.

Synopsis: GET

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: If no query parameters are passed all members are returned.

Optional Query parameters: Any property listed in section 1.2.8

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: array of Timer Objects

Response Status: 200, 400, 401, 403, 404

Example Request: Get all the Timers in the Hub.

GET: https://hub.flexeye.com/v1/iot_Default/timersAll

Query: state=3.0

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: N/A

Content-Length: N/A

Content-Type: N/A

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
  {
    "timers" : [ {
      "uri" : "https://hub.flexeye.com/v1/iot_Default/timers/T-394",
      "key" : "T-394"
    }, {
      "uri" : "https://hub.flexeye.com/v1/iot_Default/timers/T-400",
      "key" : "T-400"
    } ],
    "timersCursor" : {
      "count" : "0",
      "nextUri" : null,
      "prevUri" : null
    }
  }
]
```

```
    }  
  }  
]
```

3.1.11 Start Timer

Description: Creates and starts a Timer with the given duration for the Hub. Option to specify an on-Expiry handler which will perform an action if the Timer is allowed to expire (i.e. not Stopped - see Stop Timer below).

Synopsis: POST

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: duration, notifyType, notifyCoverage

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: Timer object

Response Status: 200, 400, 401, 403, 404

Example Request: Creates and starts a Timer named "T-453" with the given duration(3000 sec) for the Hub. If the Timer is not stopped , after the given duration(3000 sec),on-expiry handler is invoked and notification will be sent to default trustee (s) (see 3.4.18 sendNotification section)

POST https://hub.flexeye.com/v1/iot_Default/timers

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{
  "notifyType" : "0.0",
  "notifyCoverage" : "1.0",
  "duration" : "3000",
  "logMessage" : "Message to send on timer expiry specified here"
}
```

Example Response:

HTTP/1/1 201 CREATED

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
{
  "duration" : "3000",
  "state" : 0.0,
  "id" : "Timer_450",
  "notifyCoverage" : "DefaultTrustee",
  "logMessage" : "Message to send on timer expiry specified here",
  "notifyType" : "Email",
  "startTime" : "2013-11-07T17:02:04.261Z",
}
```

```
"timersCursor" : {  
  "count" : "0",  
  "nextUri" : null,  
  "prevUri" : null  
},  
"uri" : "https://hub.flexeye.com/v1/iot_Default/timers/Timer_450"  
}
```

3.1.12 Register Device Manager

Description: Register a new Device Manager on the Hub, providing a suitable URI for the Hub to access the Owner Device Manager API.

Synopsis: POST

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: N/A

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Register a new Device Manager "DM1" on the Hub

POST https://hub.flexeye.com/v1/iot_Default/dms

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{
  "id" : "DM2",
  "name" : "DM2 Test",
}
```

Example Response:

HTTP/1/1 201 CREATED

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
  {
    "name" : "DM2 Test",
    "id" : "DM2",
    "uri" : "https://hub.flexeye.com/v1/iot_Default/dms/DM2"
  }
]
```

3.1.13 Deregister Device Manager

Description: Remove an existing Device Manager from the Hub - marked as deregistered

Synopsis: PUT

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: isRegistered

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Deregister an existing Device Manager "DM2" on the Hub

PUT https://hub.flexeye.com/v1/iot_Default/dms/DM2

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{
    "isRegistered" : "False"
}
```

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
    {
        "name" : "DM2 Test",
        "id" : "DM2",
        "uri" : "https://hub.flexeye.com/v1/iot_Default/dms/DM2",
        "isRegistered" : 0.0
    }
]
```

3.1.14 Register Application

Description: Register a new Application on the Hub, optionally providing a suitable URI for the Hub to callback the Application API.

Synopsis: POST

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: applicationId

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Register a new Application "MyGuardian" on the Hub

POST https://hub.flexeye.com/v1/iot_Default/applications

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{
    "id" : "myGuardianTest"
}
```

Example Response:

HTTP/1/1 201 CREATED

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
    {
        "name" : "myGuardianTest",
        "id" : "myGuardianTest",
        "isRegistered" : 1.0,
        "uri" :
        "https://hub.flexeye.com/v1/iot_Default/applications/myGuardianTest"
    }
]
```

3.1.15 Deregister Application

Description: Remove an existing Application from the Hub - marked as deregistered

Synopsis: PUT

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: isRegistered

Request Message Body: Name/value pairs with media type “application/json”

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Deregister an existing application on the Hub

PUT https://hub.flexeye.com/v1/iot_Default/applications/MyGuardian

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{
    "isRegistered" : "False"
}
```

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
    {
        "name" : "myGuardianTest",
        "id" : "myGuardianTest",
        "isRegistered" : 0.0,
        "uri" :
        "https://hub.flexeye.com/v1/iot_Default/applications/myGuardianTest"
    }
]
```

3.1.16 Register Location Manager

Description: Register a new Location Manager on the Hub, providing a suitable URI for the Hub to access the Owner Location Manager API.

Synopsis: POST

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: locationManagerId

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Register a new Location Manager "LM1" on the Hub

POST https://hub.flexeye.com/v1/iot_Default/lms

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{
  "id" : "LM1"
}
```

Example Response:

HTTP/1/1 201 CREATED

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
  {
    "name" : "LM1",
    "id" : "LM1",
    "isRegistered" : 1.0,
    "uri" : "https://hub.flexeye.com/v1/iot_Default/lms/LM1"
  }
]
```

3.1.17 Deregister Location Manager

Description: Remove an existing Location Manager from the Hub - marked as deregistered

Synopsis: PUT

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: isRegistered

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Deregister an existing Location Manager "LM1" on the Hub

```
PUT https://hub.flexeye.com/v1/iot_Default/lms/LM1
Host: hub.flexeye.com
Authorization: Basic authentication
Accept: application/json
Content-Length: nnn
Content-Type: application/json
```

```
{
    "isRegistered" : "False"
}
```

Example Response:

```
HTTP/1/1 200 OK
Content-Type: application/vnd.com.flexeye.iot+json
Content-Length: nnn
[
    {
        "name" : "LM1",
        "id" : "LM1",
        "isRegistered" : 0.0,
        "uri" : "https://hub.flexeye.com/v1/iot_Default/lms/LM1"
    }
]
```

3.1.18 Register Person Manager

Description: Register a new Person Manager on the Hub, providing a suitable URI for the Hub to access the Owner Person Manager API.

Synopsis: POST

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: personManagerId

Request Message Body: Name/value pairs with media type “application/json”

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Register a new Person Manager "PM1" on the Hub

```
POST https://hub.flexeye.com/v1/iot_Default/pms
Host: hub.flexeye.com
Authorization: Basic authentication
Accept: application/json
Content-Length: nnn
Content-Type: application/json
```

```
{
  "id" : "PM3"
}
```

Example Response:

HTTP/1/1 201 CREATED

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
  {
    "name" : "PM3",
    "id" : "PM3",
    "isRegistered" : 1.0,
    "uri" : "https://hub.flexeye.com/v1/iot_Default/pms/PM3"
  }
]
```

3.1.19 Deregister Person Manager

Description: Remove an existing Person Manager from the Hub - marked as deregistered

Synopsis: PUT

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: isRegistered

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Deregister an existing Person Manager "PM1" on the Hub.

PUT https://hub.flexeye.com/v1/iot_Default/pms/PM3

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{
  "isRegistered" : "False"
}
```

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
  {
    "name" : "PM3",
    "id" : "PM3",
    "isRegistered" : 0.0,
    "uri" : "https://hub.flexeye.com/v1/iot_Default/pms/PM3"
  }
]
```

3.1.20 Subscribe

Description: Create a subscription to another entity.

Synopsis: POST (Operation)

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: subscribeEntity

Optional Query parameters: Any property listed in section 1.2.10

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Application "AppABC" subscribes to device entity with ID "Device_123".

POST

https://hub.flexeye.com/v1/iot_Default/applications/AppABC?subscribe&entityRef=Device_123

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{
    "entityRef" : "Device_123"
}
```

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
    {
        "results" : "Successfully subscribed",
        "operation" : "subscribe"
    }
]
```

3.1.21 Unsubscribe

Description: Deactivates a subscription.

Synopsis: POST (Operation)

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: subscribeEntity

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Deactivates the subscription application "AppABC" has with resource with ID "Device_123".

POST

https://hub.flexeye.com/v1/iot_Default/applications/AppABC?unsubscribe&entityRef=App_123

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{
    "entityRef" : "Device_123"
}
```

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
    {
        "results" : "Successfully unsubscribed",
        "operation" : "unsubscribe"
    }
]
```

3.1.22 Get Location Tag

Description: Returns a location tag.

Synopsis: GET

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: N/A

Optional Query parameters: Any property listed in section 1.2.12

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: Single Location Tag Object

Response Status: 200, 400, 401, 403, 404

Example Request: Get the Tag in the Hub with id of "52050b4dd4574".

GET: https://hub.flexeye.com/v1/iot_Default/tags/52050b4dd4574

Host: hub.flexeye.com
Authorization: Basic authentication
Accept: N/A
Content-Length: N/A
Content-Type: N/A

Example Response:

HTTP/1/1 200 OK
Content-Type: application/vnd.com.flexeye.iot+json
Content-Length: nnn

```
[
  {
    "id" : "52050b4dd4574",
    "location" : {
      "uri" :
"http://hub.flexeye.com/v1/iot_Default/tags/52050b4dd4574/location"
    },
    "lat" : "51.24207678",
    "long" : "-0.583975524",
    "areaId" : "uni_of_surrey_campus",
    "name" : "Millenium House 84",
    "uri" : "http://hub.flexeye.com/v1/iot_Default/tags/52050b4dd4574"
  }
]
```


3.2 Person Manager

3.2.1 Authenticate

Description: Used for authenticating a Person managed by the Person Manager.

Synopsis: POST (Operation)

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: username, password

Request Message Body: Name/value pairs with media type “application/json”

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Authenticate a person named “user121” in Default Person Manager.

POST https://hub.flexeye.com/v1/iot_Default/pms/dpm/authenticateUser

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{
    "username" : "user121",
    "password" : "xxxxxx"
}
```

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
{
    "results" : "True",
    "operation" : "authenticateUser"
}
```

3.2.2 Login

Description: Authenticates a Person managed by the Person Manager and establishes a session for that Person with the Hub.

Synopsis: POST (Operation)

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: username, password

Request Message Body: Name/value pairs with media type “application/json”

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Authenticates and opens a session for the user “user121”.

```
POST https://hub.flexeye.com/v1/iot_Default/pms/dpm/loginUser
Host: hub.flexeye.com
Authorization: Basic authentication
Accept: application/json
Content-Length: nnn
Content-Type: application/json
```

```
{
    "username" : "user121",
    "password" : "xxxxxxx"
}
```

Example Response:

```
HTTP/1/1 200 OK
Content-Type: application/vnd.com.flexeye.iot+json
Content-Length: nnn
```

```
{
    "results" : "True",
    "operation" : "loginUser"
}
```

3.2.3 Logout

Description: Closes an open session with the Hub for a Person managed by the Person Manager.

Synopsis: POST (Operation)

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: sessionId

Request Message Body: Name/value pairs with media type “application/json”

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Closes an opened session for sessionId “24e0945aad6f0869a77ccab063d7fdfd”

```
POST https://hub.flexeye.com/v1/iot_Default/pms/dpm/logoutUser
Host: hub.flexeye.com
Authorization: Basic authentication
```

```
Accept: application/json
Content-Length: nnn
Content-Type: application/json
```

```
{
    "sessionID" : "24e0945aad6f0869a77ccab063d7fdfd"
}
```

Example Response:

```
HTTP/1/1 200 OK
Content-Type: application/vnd.com.flexeye.iot+json
Content-Length: nnn
```

```
{
    "results" : "True",
    "operation" : "logoutUser"
}
```

3.2.4 Get Users

Description: Returns a list of Persons managed by the Person Manager.

Synopsis: GET

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: If no query parameters are passed all users are returned.

Optional Query parameters: Any property listed in section 1.2.6

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: Array of Users

Response Status: 200, 400, 401, 403, 404

Example Request: Returns the list of all people managed by "dpm"

```
GET https://hub.flexeye.com/v1/iot_Default/pms/dpm/pmUsers
Query: firstName=John
Host: hub.flexeye.com
Authorization: Basic authentication
Accept: N/A
Content-Length: N/A
Content-Type: N/A
```

Example Response:

```
HTTP/1/1 200 OK
```

Content-Type: application/vnd.com.flexeye.iot+json
Content-Length: nnn

```
[
  {
    "pmUsers" : [ {
      "uri" :
      "https://hub.flexeye.com/v1/iot_Default/pms/dpm/pmUsers/John_Person_1",
      "key" : "John_Person_1"
    }, {
      "uri" :
      "https://hub.flexeye.com/v1/iot_Default/pms/dpm/pmUsers/John_Person_2",
      "key" : "John_Person_2"
    } ],
    "pmUsersCursor" : {
      "count" : "0",
      "nextUri" : null,
      "prevUri" : null
    }
  }
]
```

3.2.5 Create User

Description: Creates a new Person managed by the Person Manager.

Synopsis: POST

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: Any property listed in section 1.2.6.

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Creates a new Person named "user121" under Person Manager "dpm"

POST https://hub.flexeye.com/v1/iot_Default/pms/dpm/pmUsers

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{
  "firstName" : "user121",
  "lastName" : "user121",
  "emailId" : "user121@flexeye.com",
  "id" : "user121",
  "phoneNumber" : "1234567891"
}
```

Example Response:

HTTP/1/1 201 CREATED

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
{
  "uri" :
  "https://hub.flexeye.com/v1/iot_Default/pms/dpm/pmUsers/user121",
  "firstName" : "user121",
  "lastName" : "user121",
  "emailId" : "user121@flexeye.com",
  "id" : "user121",
  "phoneNumber" : "1234567891"
}
```

3.2.5.1 Register User

Description: Creates a new Person managed by the Person Manager.

Synopsis: POST (Operation)

Request Headers: Accept, Authorization, Content-Length, Content-Type

Query Parameters: (Mandatory) username, emailAddress, telephoneNumber, passcode, confirmPasscode. (Optional) forename, surname, fullName.

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Creates a new Person named "user121" under Person Manager "pm1"

POST

https://hub.flexeye.com/v1/iot_Default/pms/pm1/register?fullName=user121_name
&emailAddress=user121@gmail.com&telephoneNumber=01483100200&username=user121&
passcode=PASSWORD&confirmPasscode=PASSWORD&forename=user121_fn&surname=user121_sn

Host: hub.flexeye.com

Authorization: Basic authentication
Accept: application/json
Content-Length: nnn
Content-Type: application/json

Example Response:

```
HTTP/1/1 201 CREATED
Content-Type: application/vnd.com.flexeye.iot+json
Content-Length: nnn
{
    "results" : "Created Successfully",
    "operation" : "register"
}
```

3.3 Location Manager

3.3.1 Get Locations

Description: Returns a list of Locations managed by the Location Manager.

Synopsis: GET

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: If no query parameters are passed all locations are returned.

Optional Query parameters: Any property listed in section 1.2.5

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: array of locations

Response Status: 200, 400, 401, 403, 404

Example Request: Returns the list of locations managed by location manager "CCSR_LM" whose postcode is "GU24HJ".

GET https://hub.flexeye.com/v1/iot_Default/lms/CCSR_LM/locations

Query: addr_postcode=GU24HJ

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: N/A

Content-Length: N/A

Content-Type: N/A

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
  {
    "locations" : [ {
      "uri" :
      "https://hub.flexeye.com/v1/iot_Default/lms/CCSR_LM/locations/CCSR",
      "key" : "CCSR"
    } ],
    "locationsCursor" : {
      "count" : "0",
      "nextUri" : null,
      "prevUri" : null
    }
  }
]
```


3.3.2 Create Location

Description: Creates a new Location managed by the Location Manager.

Synopsis: POST

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: Any property listed in section 1.2.5

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Create a location "location11" under location manager "CCSR_LM"

POST: https://hub.flexeye.com/v1/iot_Default/lms/CCSR_LM/locations

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{
  "name" : "location1",
  "lat" : "90E",
  "long" : "87S",
  "addr_building" : "hayshouse",
  "addr_road" : "millmead",
  "addr_other" : "Near White house",
  "addr_town" : "Guildford",
  "addr_county" : "UK",
  "addr_postcode" : "GU24HJ"
}
```

Example Response:

HTTP/1/1 201 CREATED

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
  {
    "name" : "location1",
    "id" : "location1",
    "isRegistered" : 1.0,
    "isActiveNew" : 1.0,
    "lat" : "90E",
    "long" : "87S",
  }
]
```

```
    "addr_building" : "hayshouse",
    "addr_road" : "millmead",
    "addr_other" : "Near White house",
    "addr_town" : "Guildford",
    "addr_county" : "UK",
    "addr_postcode" : "GU24HJ",
    "displayName" : "",
    "uri" :
    "https://hub.flexeye.com/v1/iot_Default/lms/CCSR_LM/locations/location1"
  }
]
```

3.4 Person

3.4.1 Get Person

Description: Returns the current state of the Person.

Synopsis: GET

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: N/A

Optional Query parameters: Any property listed in section 1.2.6

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Get the current state of person "user121" from the "myGuardian" application list of users.

GET

`https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/user121`

Host: `hub.flexeye.com`

Authorization: Basic authentication

Accept: N/A

Content-Length: N/A

Content-Type: N/A

Example Response:

HTTP/1/1 200 OK

Content-Type: `application/vnd.com.flexeye.iot+json`

Content-Length: `nnn`

```
[
  {
    "id" : "user121",
    "name" : null,
    "address" : null,
    "gender" : "NaN",
    "uri" :
      "https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/u
      ser121"
  }
]
```

3.4.1.5 Get Person Last Status

Description: Returns the person's last recorded location and device status.

Synopsis: GET

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: N/A

Optional Query parameters: N/A

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Get the last status of person "pmr2" -which is the last reported latitude, longitude, and battery level of their mobile phone.

GET

`https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2/last_status`

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: N/A

Content-Length: N/A

Content-Type: N/A

Example Response:

HTTP/1.1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
  {
    "lat" : 90.0,
    "long" : 90.0,
    "battery_level" : 100.0,
    "uri" :
      "http://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2/last_status"
  }
]
```

3.4.2 Update Person

Description: Updates the state (properties) of a Person.

Synopsis: PUT

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: Any property listed in section 1.2.6

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Updates first name, last name, email Id and phone number for the person with id "pmr2" in the hub.

PUT https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{
  "firstName" : "updatedFN",
  "lastName" : "updatedLN",
  "emailId" : "eN@f.cm",
  "phoneNumber" : "1234567891"
}
```

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
  {
    "id" : "user121",
    "phoneNumber" : "1234567891",
    "emailId" : "eN@f.com",
    "firstName" : "updatedFN",
    "lastName" : "updatedLN",
    "uri" :
    "https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2"
  }
]
```

3.4.2.1 Change Password

Description: Change Password of a Person.

Synopsis: POST

Request Headers: Accept, Authorization, Content-Length, Content-Type

Query Parameters: oldPassword, newPassword, repeatNewPassword

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Updates an existing person " user121" in the hub.

POST

https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/user121/changePassword?oldPassword=OLD_PASSWORD_HERE&newPassword=NEW_PASSWORD_HERE&repeatNewPassword=NEW_PASSWORD_HERE

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
  {
    "results" : "Password changed successfully",
    "operation" : "changePassword"
  }
]
```

Note: Not Applicable for the LDAP Users

3.4.3 Delete Person

Description: Used to delete a Person from the Hub.

Synopsis: DELETE

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: N/A

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Deletes a person named "pmr2" under the Hub

DELETE

```
https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2
Host: hub.flexeye.com
Authorization: Basic authentication
```

Example Response:

```
HTTP/1/1 200 OK
Content-Type: application/vnd.com.flexeye.iot+json
Content-Length: nnn
```

3.4.4 Get Trustees

Description: Returns a list of the trustees of the Person.

Synopsis: GET

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: If no query parameters are passed all members are returned.

Optional Query parameters: Any property listed in section 1.2.6

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: array of trustees

Response Status: 200, 400, 401, 403, 404

Example Request: Get the list of trustees of the Person named "pmr2" where their email is "gs@flexeye.com"

```
GET
https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2/tru
stees
Query: emailId=gs@flexeye.com
Host: hub.flexeye.com
Authorization: Basic authentication
Accept: N/A
Content-Length: N/A
Content-Type: N/A
```

Example Response:

```
HTTP/1/1 200 OK
Content-Type: application/vnd.com.flexeye.iot+json
Content-Length: nnn
```

```
[
  {
    "trustees" : [ {
      "uri" :
        "http://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pm
```

```
    r2/trustees/gs",
    "key" : "gs"
  } ],
  "trusteesCursor" : {
    "count" : "0",
    "nextUri" : null,
    "prevUri" : null
  }
}
```

3.4.5 Add Trustee

Description: Looks for the specified user within the application's users list and, if found, automatically sends a trustee invitation message to all personal devices of that user.

Synopsis: PUT

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: `proposedTrustees`

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: User "pmr2" invites "ds1@flexeye.com" to be one of their trustees.

PUT https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2

Host: hub.flexeye.com/

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{
  "proposedTrustees" : "ds1@flexeye.com"
}
```

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
  {
    "id" : "pmr2",
    "name" : null,
    "phoneNumber" : "",
```



```
    "emailId" : "pmr2@flexeye.com",
    "proposedTrustees" : [ {
      "uri" :
"https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2/pr
oposedTrustees/TestUser01",
      "key" : "TestUser01"
    }, {
      "uri" :
"https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2/pr
oposedTrustees/ds1@flexeye.com",
      "key" : "ds1@flexeye.com"
    } ],
    "proposedTrusteesCursor" : {
      "count" : "0",
      "nextUri" : null,
      "prevUri" : null
    },
    "firstName" : "pmr2_FN",
    "lastName" : "pmr2_LN",
    "address" : null,
    "gender" : "NaN",
    "uri" :
"https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2"
  }
]
```

3.4.6 Remove Trustee

Description: Removes a trustee from the list of trustees of the Person.

Synopsis: POST (Operation)

Request Headers: Accept, Authorization, Content-Length, Content-Type

Query Parameters: personID

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Remove a trustee named "ds1@flexeye.com" from the "pmr2" trustee's list

POST

https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2/rem
oveTrustee?personID=ds1@flexeye.com

Host: hub.flexeye.com

Authorization: Basic authentication

Example Response:

```
HTTP/1/1 200 OK
Content-Type: application/vnd.com.flexeye.iot+json
Content-Length: nnn
[ {
    "results": "Trustee with name ds1@flexeye.com is successfully removed",
    "operation": "removeTrustee"
}]
```

3.4.7 Set Default Trustee

Description: Sets the default trustee for the Person.

Synopsis: PUT

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: defaultTrustee

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Set the default trustee for person "pmr" to "gs@flexeye.com".

```
PUT https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr
Host: hub.flexeye.com
Authorization: Basic authentication
Accept: application/json
Content-Length: nnn
Content-Type: application/json
```

```
{
    "defaultTrustee" : "gs@flexeye.com"
}
```

Example Response:

```
HTTP/1/1 200 OK
Content-Type: application/vnd.com.flexeye.iot+json
Content-Length: nnn

[
    {
        "id" : "user121",
        "name" : null,
        "defaultTrustee" : {
            "uri" :
"https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/user121/defaultTrustee"
```

```
    },
    "uri" :
"https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/user121"
  }
]
```

3.4.8 Get Locations

Description: Returns a list of locations for the Person.

Synopsis: GET

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: If no query parameters are passed all locations are returned.

Optional Query parameters: Any property listed in section 1.2.5

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: array of locations

Response Status: 200, 400, 401, 403, 404

Example Request: Get the list of locations associated with the user "pmr2", where the location country is UK.

GET

https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2/locations

Query: addr_country=UK

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: N/A

Content-Length: N/A

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
  {
    "locations" : [ {
      "uri" :
"https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2/locations/location1",
      "key" : "location1"
    }, {
      "uri" :
```

```
"https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/p
mr2/locations/TestLocation1",
  "key" : "TestLocation1"
} ],
"locationsCursor" : {
  "count" : "0",
  "nextUri" : null,
  "prevUri" : null
}
}
```

3.4.8.1 Search Campus Location

Description: Search a Campus Location.

Synopsis: POST (Operation)

Request Headers: Accept, Authorization, Content-Length, Content-Type

Query parameters: locationName

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: array of locations

Response Status: 200, 400, 401, 403, 404

Example Request: Search for campus locations whose name include the value "Ballast00002913".

POST

https://hub.flexeye.com/v1/iot_Default/searchCampusLocation?locationName=Ballast00002913

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: N/A

Content-Length: N/A

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
  {
    "operation" : "searchCampusLocation",
    "results" : [ {
      "id" : "00002913",
      "areaId" : null,
```

```
    "name" : "Ballast00002913",
    "lat" : "52.1815699937106",
    "long" : "0.178138911724091",
    "tags" : {
      "tags" : [ ],
      "tagsCursor" : {
        "count" : "0",
        "nextUri" : null,
        "prevUri" : null
      }
    },
    "uri" :
    "http://hub.flexeye.com/v1/iot_Default/allLocationCampuses/002b
    9de02eff498e6bc38b3ff3f833dc"
  }
]
```

3.4.9 Add Location

Description: Adds a location to the list of locations for the Person.

Synopsis: PUT

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: locations

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Add "CCSR" to the locations associated with person "pmr2".

PUT https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{
    "locations" : "CCSR"
}
```

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
  {
    "id" : "pmr2",
    "name" : null,
    "phoneNumber" : "1234567891",
    "emailId" : "eN@f.com",
    "firstName" : "updatedFN",
    "lastName" : "updatedLN",
    "locations" : [ {
      "uri" :
      "https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/p
mr2/locations/TestLocation1",
      "key" : "TestLocation1"
    }, {
      "uri" :
      "https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/p
mr2/locations/CCSR",
      "key" : "CCSR"
    } ],
    "locationsCursor" : {
      "count" : "0",
      "nextUri" : null,
      "prevUri" : null
    },
    "uri" :
    "https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/p
mr2"
  }
]
```

3.4.10 Remove Location

Description: Removes a location from the list of locations for the Person.

Synopsis: DELETE

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: N/A

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Removes "CCSR" location from the locations associated with person "user121".

DELETE

https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/user121/locations/CCSR

Host: hub.flexeye.com

Authorization: Basic authentication

Example Response:

HTTP/1/1 204 No Content

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

3.4.11 Add Device

Description: Adds a device to the list of devices for the Person.

Synopsis: PUT

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: devices

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Adds device with ID "135" to the user121's devices list

PUT https://hub.flexeye.com/v1/iot_Default/allPeople/user121

Host: hub.flexeye.com

Authorization: Basic authentication

```
Accept: application/json
Content-Length: nnn
Content-Type: application/json
```

```
{
    "devices" : "135"
}
```

Example Response:

```
HTTP/1/1 200 OK
Content-Type: application/vnd.com.flexeye.iot+json
Content-Length: nnn
```

```
[
    {
        "id" : "user121",
        "name" : null,
        "phoneNumber" : "1234567891",
        "emailId" : "eN@f.com",
        "firstName" : "updatedFN",
        "lastName" : "updatedLN",
        "devices" : [ {
            "uri" :
            "https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/u
ser121/devices/135",
            "key" : "135"
        } ],
        "devicesCursor" : {
            "count" : "0",
            "nextUri" : null,
            "prevUri" : null
        },
        "uri" :
        "https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/u
ser121"
    }
]
```

3.4.12 Remove Device

Description: Removes a device from the list of devices for the Person.

Synopsis: DELETE

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: N/A

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Removes device with ID "135" from the user's (user121) devices list.

DELETE https://hub.flexeye.com/v1/iot_Default/allPeople/user121/devices/135

Host: hub.flexeye.com

Authorization: Basic authentication

Example Response:

HTTP/1/1 204 No Content

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

3.4.13 Get Events

Description: Returns a list of Events for the Person.

Synopsis: GET

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: If no query parameters are passed all events are returned.

Optional Query parameters: Any property listed in section 1.2.4.3

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: Array of Events

Response Status: 200, 400, 401, 403, 404

Example Request: Get the list of events for the person named "user121" where the type is 1.0 (input)

GET https://hub.flexeye.com/v1/iot_Default/allPeople/user121/inOutEvents

Query: type=1.0

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: N/A

Content-Length: N/A

Content-Type: N/A

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
  {
    "events" : [ {
      "uri" :
      "https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/ser121/inOutEvents/30538",
      "key" : "30538"
    } ],
    "eventsCursor" : {
      "count" : "0",
      "nextUri" : null,
      "prevUri" : null
    }
  }
]
```

3.4.14 Log Input Event

Description: Logs an input event for the Person

Synopsis: POST

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: Any property listed in section 1.2.4.3

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: logs an event with timestamp of "2013-09-25T14:30:34.000Z" and type of 1.0 (Input Event) for the person "pmr2".

POST

https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pm2/events

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{
  "timeStamp" : "2013-09-25T14:30:34.000Z",
  "payload" : "10",
  "type" : "1.0"
}
```

Example Response:

HTTP/1/1 201 CREATED

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
  {
    "timeStamp" : "2013-11-08T09:00:34.000Z",
    "type" : 1.0,
    "payload" : "10",
    "displayTimeStamp" : "2013-11-08T09:00:34.000Z",
    "source" : null,
    "id" : "Event_user121_38",
    "isUpdated" : "NaN",
    "uri" :
    "https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2/events/Event_user121_38"
  }
]
```

3.4.15 Log Output Event

Description: Logs an output event for the Person

Synopsis: POST

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: Any property listed in section 1.2.4.3

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: logs an event with timestamp of "2013-09-25T14:30:34.000Z" and a type of 0.0 (Output event) for the person "pmr2".

POST

https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2/events

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{
  "timeStamp" : "2013-09-25T14:30:34.000Z"
```

```
    "payload" : "10",  
    "type" : "0.0"  
}
```

Example Response:

HTTP/1/1 201 CREATED

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[  
  {  
    "timeStamp" : "2013-11-08T09:00:34.000Z",  
    "type" : 0.0,  
    "payload" : "10",  
    "displayTimeStamp" : "2013-11-08T09:00:34.000Z",  
    "source" : null,  
    "id" : "Event_user121_39",  
    "isUpdated" : "NaN",  
    "uri" :  
    "https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/p  
mr2/events/Event_user121_39"  
  }  
]
```

3.4.16 Start Timer

Description: Creates and starts a Timer with the given duration. Duration (seconds) must be passed.

When the timer expires, a boilerplate “expiry” notification is sent to a default trustee or all trustees - see section 4.2.

The properties below are assigned the following default values:-

“notifyCoverage” defaults to 1.0 - ‘DefaultTrustee’ only.

“defaultTrustee” is defaulted to the “defaultTrustee” defined on the user.

“trustees” is defaulted to the “trustees” defined on the user.

“state” to 1.0 (ticking).

“startTime” set to the time of timer creation.

(Note: these three properties can be amended on or after the creation of a timer).

Synopsis: POST

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: duration

Optional Query parameters: Any property listed in section 1.2.8

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body:

Response Status: 200, 400, 401, 403, 404

Example Request: Creates a Timer for the Person "pmr2", with a duration of 3000 seconds and notification on expiry to be sent to all trustees defined on the timer (notifyCoverage = 2.0).

POST

https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2/timers

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{
  "notifyCoverage" : "2.0",
  "duration" : "3000",
  "onPauseNotify" : "1.0"
}
```

Example Response:

HTTP/1/1 201 CREATED

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
  {
    "duration" : "3000",
    "state" : 0.0,
    "id" : "Timer_789",
    "notifyCoverage" : "2.0",
    "onPauseNotify" : "1.0",
    "uri" :
    "https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2/timers/Timer_789"
  }
]
```

3.4.16.5 Get Timers

Description: Returns a list of timers for the Person.

Synopsis: GET

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: If no query parameters are passed all members of the collection are returned.

Optional Query parameters: Any property listed in section 1.2.8

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: Timer Object

Response Status: 200, 400, 401, 403, 404

Example Request: Get all the Timers for the Person "user121" where the timer state is expired.

GET: `https://hub.flexeye.com/v1/iot_Default/allPeople/user121/timers`

Query: `state=3.0`

Host: `hub.flexeye.com`

Authorization: `Basic authentication`

Accept: `N/A`

Content-Length: `N/A`

Content-Type: `N/A`

Example Response:

`HTTP/1/1 200 OK`

Content-Type: `application/vnd.com.flexeye.iot+json`

Content-Length: `nnn`

```
[
  {
    "timers" : [ {
      "uri" :
      "https://hub.flexeye.com/v1/iot_Default/allPeople/user121/timers/Timer_
      user121_1",
      "key" : "Timer_user121_1"
    }, {
      "uri" :
      "https://hub.flexeye.com/v1/iot_Default/allPeople/user121/timers/Timer_
      user121_2",
      "key" : "Timer_user121_2"
    } ],
    "timersCursor" : {
      "count" : "0",
      "nextUri" : null,
      "prevUri" : null
    }
  }
]
```

```
    }  
  }  
]
```

3.4.17 Get Notifications

Description: Returns a list of Events for the Person.

Synopsis: GET

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: If no query parameters then returns all notifications;

Optional Query parameters: Any property listed in section 1.2.9

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: TBD

Response Status: 200, 400, 401, 403, 404

Example Request: Get all the notifications for the Person named "user121" where the notification type is 0.0 (Email).

GET

https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/user121/notifications

Query: type=0.0 (Email)

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: N/A

Content-Length: N/A

Content-Type: N/A

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[  
  {  
    "notifications" : [ {  
      "uri" :  
"https://hub.flexeye.com/v1/iot_Default/allPeople/user121/notifications  
/N-1077",  
      "key" : "N-1077"  
    }, {  
      "uri" :  
"https://hub.flexeye.com/v1/iot_Default/allPeople/user121/notifications  
/N-1078",  
      "key" : "N-1078"  
    } ],  
  },  
]
```

```
    "notificationsCursor" : {  
      "count" : "0",  
      "nextUri" : null,  
      "prevUri" : null  
    }  
  }  
]
```


3.4.18 Send Notification (to one device)

Description: Used to send a notification to one Device.

Synopsis: POST (Operation)

Request Headers: Accept, Authorization, Content-Length, Content-Type

Query Parameters: registrationId, msg, msgTemplate, timerId, senderText

(see section 4.2 for possible values to pass for the “msgTemplate” parameter)

Request Message Body: Name/value pairs with media type “application/json”

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Send a Notification to a Person’s device.

POST

https://hub.flexeye.com/v1/iot_Default/allPeople/user1/sendMessageToDevice?registrationId=0000123&msg=Notification_message_here

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
{
  "results" : "True",
  "operation" : "sendMessageToDevice"
}
```

3.4.18.5 Send Notification (to all devices of a users)

Description: Used to send a notifications to the one or more devices associated with a person.

Synopsis: POST (Operation)

Request Headers: Accept, Authorization, Content-Length, Content-Type

Query Parameters: msg, notificationType, timerId, msgTemplate, senderText

(see section 4.2 for possible values to pass for the “msgTemplate” parameter)

Request Message Body: Name/value pairs with media type “application/json”

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Send a Notification to all devices of a Person, setting the notification type to “Invite Trustee notification” and timer ID to “Timer464”.

```
POST
https://hub.flexeye.com/v1/iot_Default/allPeople/user1/sendMessageToAllDevices?msg=NotificationMessageHere&notificationType=inviteTrustee&timerId=Timer_464
Host: hub.flexeye.com
Authorization: Basic authentication
Accept: application/json
Content-Length: nnn
Content-Type: application/json
```

Example Response:

```
HTTP/1/1 200 OK
Content-Type: application/vnd.com.flexeye.iot+json
Content-Length: nnn
```

```
{
  "results" : "True",
  "operation" : "sendMessageToAllDevices"
}
```

3.4.19 Accept Trustee Invitation

Description: Used to add a trustee after they accept their trustee invitation.

Synopsis: POST (Operation)

Request Headers: Accept, Authorization, Content-Length, Content-Type

Query Parameters: `personId`

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: User "ds1@flexeye.com" has agreed trustee invitation from user "pmr2" and so is added to pmr2's trustees list.

POST

`https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2/acceptTrustee?personId=ds1@flexeye.com`

Host: `hub.flexeye.com`

Authorization: `Basic authentication`

Accept: `application/json`

Content-Length: `nnn`

Content-Type: `application/json`

Example Response:

`HTTP/1/1 200 OK`

Content-Type: `application/vnd.com.flexeye.iot+json`

Content-Length: `nnn`

```
{
  "results" : "True",
  "operation" : "acceptTrustee"
}
```

3.4.20 Reject Trustee Invitation

Description: Used to reject trustee invitation. The invited trustee is not be added to the user's trustees list, but they are added to the user's rejectedTrustees list.

Synopsis: `POST (Operation)`

Request Headers: Accept, Authorization, Content-Length, Content-Type

Query Parameters: `personId`

Request Message Body: Name/value pairs with media type “application/json”

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: The user “ds1@flexeye.com” has rejected the invite to be a trustee for user “pmr2”.

POST

https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2/rejectTrustee?personId=ds1@flexeye.com

Host: hub.flexeye.com

Authorization: **Basic authentication**

Accept: application/json

Content-Length: **nnn**

Content-Type: application/json

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
{
  "results" : "True",
  "operation" : "rejectTrustee"
}
```

3.4.21 Delete Notification

Description: Used to delete a Notification object from a Person.

Synopsis: DELETE

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: N/A

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Deletes a notification with ID "N1" from user121.

DELETE

https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/user121/notifications/N1

Host: hub.flexeye.com

Authorization: Basic authentication

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

3.4.22 Safety Alert

Description: Sends a safety alert message, using a defined template (see section 4.2), to all devices of all trustees of the user.

Synopsis: POST (Operation)

Request Headers: Accept, Authorization, Content-Length, Content-Type

Query Parameters: N/A

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Sends a safety alert message to all the user's trustees.

POST

https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2/safetyAlert

Host: hub.flexeye.com

Authorization: Basic authentication

```
Accept: application/json
Content-Length: nnn
Content-Type: application/json
```

Example Response:

```
HTTP/1/1 200 OK
Content-Type: application/vnd.com.flexeye.iot+json
Content-Length: nnn
```

```
{
  "results" : "True",
  "operation" : "safetyAlert"
}
```

3.4.23 Add Personal Device

Description:

Synopsis: POST

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: regId, medium.

Request Message Body: Name/value pairs with media type “application/json”

Response Headers: Content-Length, Content-Type

Response Message Body:

Response Status: 200, 400, 401, 403, 404

Example Request: Adds a new personal device to the user “pmr2” which uses Google Cloud Messaging as the notification medium.

POST

https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2/myPersonalDevices

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{
  "regId" : "000123",
  "medium" : "3.0"
}
```

Example Response:

HTTP/1/1 201 CREATED

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
  {
    "regId" : "000123",
    "medium" : "3.0",
    "uri" :
    "https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2/myPersonalDevices/000123"
  }
]
```

3.4.24 Delete Personal Device

Description: Removes a User's Personal Device.

Synopsis: DELETE

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: N/A

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Removes a User's Personal Device identified by "000123".

DELETE

https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2/myPersonalDevices/000123

Host: hub.flexeye.com

Authorization: Basic authentication

Example Response:

HTTP/1/1 204 No Content

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

3.5 Location

3.5.1 Get Location

Description: Retrieves the current state of the Location.

Synopsis: GET

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: N/A

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: Location object

Response Status: 200, 400, 401, 403, 404

Example Request: Retrieves the current state of the location "FlexeyeLoc".

GET https://hub.flexeye.com/v1/iot_Default/allLocations/FlexeyeLoc

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: N/A

Content-Length: N/A

Content-Type: N/A

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
  {
    "name" : "FlexeyeLoc",
    "id" : "FlexeyeLoc",
    "addr_building" : "HaysHouse",
    "addr_road" : "Hays Road",
    "addr_other" : "Mead",
    "addr_town" : "Guildford",
    "addr_country" : "UK",
    "addr_postcode" : "GU 123",
    "displayName" : null,
    "lat" : "90S",
    "long" : "84N",
    "uri" :
      "https://hub.flexeye.com/v1/iot_Default/allLocations/FlexeyeLoc"
  }
]
```

3.5.2 Update Location

Description: Updates the state of the Location.

Synopsis: PUT

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: Any property listed in section 1.2.5

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Updates the state of the location "location1".

PUT https://hub.flexeye.com/v1/iot_Default/allLocations/location1

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{
  "lat" : "90W",
  "long" : "89S",
  "addr_building" : "updatedbuilding",
  "addr_road" : "updatedroad",
  "addr_other" : "updatedother",
  "addr_town" : "updatedtown",
  "addr_county" : "UK",
  "addr_postcode" : "GU2 4HJ"
}
```

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
  {
    "name" : "location1",
    "id" : "location1",
    "addr_building" : "updatedbuilding",
    "addr_road" : "updatedRoad",
    "addr_other" : "updatedOther",
    "addr_town" : "updatedtown",
    "addr_country" : "UK",
    "addr_postcode" : "GU2 4HJ",
    "lat" : "90W",
    "long" : "89S",
```

```
    "uri" :  
    "https://hub.flexeye.com/v1/iot_Default/allLocations/location1"  
  }  
]
```

3.5.3 Delete Location

Description: Removes a Location from a location manager's locations.

Synopsis: DELETE

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: N/A

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Removes the Location "location1" from the locations associated with location manager "CCSR_LM".

DELETE https://hub.flexeye.com/v1/iot_Default/lms/CCSR_LM/locations/location1

Host: hub.flexeye.com

Authorization: Basic authentication

Example Response:

HTTP/1/1 204 No Content

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

3.5.4 Get Events

Description: Returns a list of Events for the Location.

Synopsis: GET

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: If no query parameters are passed all events are returned.

Optional Query parameters: Any property listed in section 1.2.4.3

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: Array of Events

Response Status: 200, 400, 401, 403, 404

Example Request: Get all input Events for the location "CCSR".

GET https://hub.flexeye.com/v1/iot_Default/allLocations/CCSR/inOutEvents
Query: type=1.0 (Input)
Host: hub.flexeye.com
Authorization: Basic authentication
Accept: N/A
Content-Length: N/A
Content-Type: N/A

Example Response:

HTTP/1/1 200 OK
Content-Type: application/vnd.com.flexeye.iot+json
Content-Length: nnn

```
[
  {
    "inOutEvents" : [ {
      "uri" :
      "https://hub.flexeye.com/v1/iot_Default/allLocations/CCSR/inOutEvents/35",
      "key" : "35"
    }, {
      "uri" :
      "https://hub.flexeye.com/v1/iot_Default/allLocations/CCSR/inOutEvents/41",
      "key" : "41"
    }, {
      "uri" :
      "https://hub.flexeye.com/v1/iot_Default/allLocations/CCSR/inOutEvents/42",
      "key" : "42"
    } ],
    "inOutEventsCursor" : {
      "count" : "0",
      "nextUri" : null,
      "prevUri" : null
    }
  }
]
```

3.5.5 Log Output Event

Description: Logs an output event for the Location.

Synopsis: POST

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: N/A

Optional Query parameters: Any property listed in section 1.2.4.3

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Logs an output event under the "CCSR" location with a timestamp of "25-Sep-2013-14-30-34".

POST https://hub.flexeye.com/v1/iot_Default/allLocations/CCSR/outEvents

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{
  "timeStamp" : "2013-08-21T18:42:12.000Z"
  "payload" : "10.413"
}
```

Example Response:

HTTP/1/1 201 CREATED

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
  {
    "timeStamp" : "2013-08-21T18:42:12.000Z",
    "type" : 0.0,
    "payload" : "10.413",
    "id" : "80",
    "uri" :
    "https://hub.flexeye.com/v1/iot_Default/allLocations/CCSR/outEvents/80"
  }
]
```

3.5.6 Log Input Event

Description: Logs an input event for the Location.

Synopsis: POST

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: N/A

Optional Query parameters: Any property listed in section 1.2.4.3

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Logs an input event under the “CCSR” location with a timestamp of "25-Sep-2013-14-30-34".

POST https://hub.flexeye.com/v1/iot_Default/allLocations/CCSR/inEvents

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{
  "timeStamp" : "2013-08-21T18:42:12.000Z"
  "payload" : "10.413"
}
```

Example Response:

HTTP/1/1 201 CREATED

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
  {
    "timeStamp" : "2013-08-21T18:42:12.000Z",
    "type" : 1.0,
    "payload" : "10.413",
    "id" : "81",
    "uri" :
    "https://hub.flexeye.com/v1/iot_Default/allLocations/CCSR/inEvents/81"
  }
]
```

3.6 Device Manager

3.6.1 Get Device Manager

Description: Get a Device Manager object.

Synopsis: POST

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: N/A

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Retrieve Device Manager "CCSR_DM".

GET https://hub.flexeye.com/v1/iot_Default/dms/CCSR_DM

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: N/A

Content-Length: N/A

Content-Type: N/A

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
  {
    "name" : "CCSR_DM",
    "id" : "CCSR_DM",
    "owner" : {
      "uri" : "https://hub.flexeye.com/v1/iot_Default/dms/CCSR_DM/owner"
    },
    "uri" : "https://hub.flexeye.com/v1/iot_Default/dms/CCSR_DM",
    "ownerUri" : "CCSR_DM",
    "activeDevices" : 190.0,
    "totalDevices" : 190.0,
    "isRegisteredSubscription" : "NaN"
  }
]
```

]

3.6.2 Update Device Manager

Description: Updates an existing Device Manager on the Hub

Synopsis: PUT

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: Any property listed in section 1.2.1

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Updates the Device Manager "DM1".

PUT https://hub.flexeye.com/v1/iot_Default/dms/DM1

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{
    "description" : "updated description"
}
```

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
{
    "uri" : "https://hub.flexeye.com/v1/iot_Default/dms/DM1",
    "id" : "DM1",
    "description" : "updated description"
}
```

3.6.3 Get Devices

Description: Returns a list of Devices managed by the Device Manager.

Synopsis: GET

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: If no query parameters are passed all members of the collection are returned.

Optional Query parameters: Any Device property listed in section 1.2.4.

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: Array of Devices

Response Status: 200, 400, 401, 403, 404

Example Request: Get all active Devices managed by the Device Manager "CCSR_DM".

GET https://hub.flexeye.com/v1/iot_Default/dms/CCSR_DM/devices

Query: isActive=1.0 (true)

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: N/A

Content-Length: N/A

Content-Type: N/A

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
  {
    "deviceId" : "135",
    "locationID" : "CSSR",
    "deviceManagerID" : "CCSR_DM",
    "description" : "temperature device"
    "isActive" : "1.0"
  },
  {
    "deviceId" : "121",
    "locationID" : "CSSR",
    "deviceManagerID" : "CCSR_DM",
    "description" : "humidity device"
    "isActive" : "1.0"
  }
]
```

3.6.4 Create Device

Description: Creates a new Device managed by a Device Manager.

Synopsis: POST

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: N/A

Optional Query parameters: Any property listed in section 1.2.4

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Creates a new Device managed by the Device Manager "CCSR_DM". Sets the id and name to "device02", device type to "MultiSensor" and device owner to Joe Bloggs.

POST https://hub.flexeye.com/v1/iot_Default/dms/CCSR_DM/devices

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{
  "id " : "device02",
  "name" : "device02",
  "deviceType " : "MultiSensor",
  "description " : "This is our First Device"
  "deviceOwner " : "JoeBloggs"
}
```

Example Response:

HTTP/1/1 201 CREATED

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
{
  "uri" :
  "https://hub.flexeye.com/v1/iot_Default/dms/CCSR_DM/devices/device02",
  "id " : "device02",
  "name" : "device02",
  "deviceType " : "MultiSensor",
  "description " : "This is our First Device"
  "deviceOwner" : {
    "uri" :
    "https://hub.flexeye.com/v1/iot_Default/dms/CCSR_DM/devices/device02/de
viceOwner"
  }
}
```

3.6.5 Get Events

Description: Returns a list of Events for the Device Manager.

Synopsis: GET

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: If no query parameters are passed all events are returned.

Optional Query parameters: Any property listed in section 1.2.4.3

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: Array of Events

Response Status: 200, 400, 401, 403, 404

Example Request: Returns a list of output Events for the Device Manager "CCSR_DM".

GET https://hub.flexeye.com/v1/iot_Default/dms/CCSR_DM/inOutEvents

Query: type=0.0 (Output)

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: N/A

Content-Length: N/A

Content-Type: N/A

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
  "inOutEvents" : {
    "inOutEvents" : [ {
      "uri" : "https://hub.flexeye.com/v1/iot_Default/dms/CCSR_DM/30492",
      "key" : "30492"
    } ],
    "inOutEventsCursor" : {
      "count" : "0",
      "nextUri" : null,
      "prevUri" : null
    }
  }
]
```

3.6.6 Device IO Event (In)

Description : Device IO Event (in) receives an input event to an existing Device from the Hub. It's an Owner Device Manager API that is called by the Hub service.

Synopsis: POST

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: N/A

Request Message Body: Device Input Event

Response Headers: Content-Length, Content-Type

Response Message Body: Name/value pairs with media type "application/json"

Response Status: 200, 400, 401, 403, 404

Example Request: Send input event to Device with MAC address "00-14-22-01-23-45" for Device Manager named "eseye01".

POST /eseyedmpath/inputs

Host: eseyehost.com

Authorization: Basic xxxxxxxxxxxxxxxxxxxx

Accept: application/vnd.com.flexeye.iot.dm+json

Content-Length: nnn

Content-Type: application/vnd.com.flexeye.iot.dm+json

```
{
  "timeStamp" : "2013-09-25T13:30:34.000Z",
  "type" : 1.0,
  "payload" : "10.1234",
  "source" : "00-14-22-01-23-45"
}
```

Example Response:

HTTP/1/1 200 OK

Content-Type: application/json

Content-Length: nnn

```
{
  "result" : "Device input event processed description/confirmation TBD"
}
```

3.7 Device

3.7.1 Get Device

Description: Retrieves the current state of the Device.

Synopsis: GET

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: N/A

Optional Query parameters: N/A

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: device object

Response Status: 200, 400, 401, 403, 404

Example Request: Get the current state of the Device "135".

GET https://hub.flexeye.com/v1/iot_Default/allDevices/135

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: N/A

Content-Length: N/A

Content-Type: N/A

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
{
  "name" : "135",
  "id" : "135",
  "owner" : {
    "uri" : null
  },

  "subDevices" : {
    "subDevices" : [ {
      "uri" :
"https://hub.flexeye.com/v1/iot_Default/allDevices/135/reading_light",
      "key" : "reading_light"
    }, {
      "uri" :
```

```
"https://hub.flexeye.com/v1/iot_Default/allDevices/135/reading_mic",
  "key" : "reading_mic"
}, {
  "uri" :
"https://hub.flexeye.com/v1/iot_Default/allDevices/135/reading_PIR",
  "key" : "reading_PIR"
}, {
  "uri" :
"https://hub.flexeye.com/v1/iot_Default/allDevices/135/reading_temp",
  "key" : "reading_temp"
}, {
  "uri" :
"https://hub.flexeye.com/v1/iot_Default/allDevices/135/reading_watts",
  "key" : "reading_watts"
} ],
"subDevicesCursor" : {
  "count" : "0",
  "nextUri" : null,
  "prevUri" : null
}
},
"uri" : "https://hub.flexeye.com/v1/iot_Default/allDevices/135",
"macAddress" : "135",
"isActive" : 1.0,
"description" : "updatedDesc"
}
]
```

3.7.2 Update Device

Description: Updates the current state of the Device.

Synopsis: PUT

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: Any property listed in section 1.2.4

Optional Query parameters: N/A

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Updates the current state of the Device "135".

PUT https://hub.flexeye.com/v1/iot_Default/allDevices/135

Host: hub.flexeye.com

Authorization: Basic authentication

```
Accept: application/json
Content-Length: nnn
Content-Type: application/json
```

```
{
  "description" : "This is an updated description"
}
```

Example Response:

```
HTTP/1/1 200 OK
Content-Type: application/vnd.com.flexeye.iot+json
Content-Length: nnn
```

```
[
  {
    "name" : "135",
    "id" : "135",
    "uri" : "https://hub.flexeye.com/v1/iot_Default/allDevices/135",
    "description" : "This is an updated description"
  }
]
```

3.7.3 Delete Device

Description: Deletes a Device

Synopsis: DELETE

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: N/A

Request Message Body: Name/value pairs with media type “application/json”

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Deletes the Device "F_D010" found under the “Flexeye_DM” device manager.

```
DELETE https://hub.flexeye.com/v1/iot_Default/dms/Flexeye_DM/devices/F_D010
Host: hub.flexeye.com
Authorization: Basic authentication
```

Example Response:

```
HTTP/1/1 204 No Content
Content-Type: application/vnd.com.flexeye.iot+json
Content-Length: nnn
```

3.7.4 Get Sub Devices

Description: Returns a list of Sub devices for a Device.

Synopsis: GET

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: If no query parameters are passed all events are returned.

Optional Query parameters: Any property listed in section 1.2.4

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: Array of sub devices

Response Status: 200, 400, 401, 403, 404

Example Request: Get all Sub devices for the Device "135" whose location is "CCSR".

GET https://hub.flexeye.com/v1/iot_Default/allDevices/135/subDevices

Query: location=CCSR

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: N/A

Content-Length: N/A

Content-Type: N/A

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[
  "subDevices" : {
    "subDevices" : [ {
      "uri" :
      "https://hub.flexeye.com/v1/iot_Default/allDevices/135/reading_light",
      "key" : "reading_light"
    }, {
      "uri" :
      "https://hub.flexeye.com/v1/iot_Default/allDevices/135/reading_mic",
      "key" : "reading_mic"
    }, {
      "uri" :
      "https://hub.flexeye.com/v1/iot_Default/allDevices/135/reading_PIR",
      "key" : "reading_PIR"
    }, {
      "uri" :
      "https://hub.flexeye.com/v1/iot_Default/allDevices/135/reading_temp",
      "key" : "reading_temp"
    }, {
```

```
        "uri" :  
        "https://hub.flexeye.com/v1/iot_Default/allDevices/135/reading_watts",  
        "key" : "reading_watts"  
    } ],  
    "subDevicesCursor" : {  
        "count" : "0",  
        "nextUri" : null,  
        "prevUri" : null  
    }  
}  
  
]
```

3.7.5 Get Events

Description: Returns a list of Events for the Device.

Synopsis: GET

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: If no query parameters are passed all events are returned.

Optional Query parameters: Any property listed in section 1.2.4.3

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: Array of Events

Response Status: 200, 400, 401, 403, 404

Example Request: Get all input Events for the Device "135".

GET https://hub.flexeye.com/v1/iot_Default/allDevices/135/inOutEvents

Query: type=1.0 (Input)

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: N/A

Content-Length: N/A

Content-Type: N/A

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
[  
    "inOutEvents" : {  
        "inOutEvents" : [ {  
            "uri" :  
            "https://hub.flexeye.com/v1/iot_Default/allDevices/135/30493",  
            "key" : "30493"  
        }  
    ]  
}
```



```
    } ],  
    "inOutEventsCursor" : {  
      "count" : "0",  
      "nextUri" : null,  
      "prevUri" : null  
    }  
  }  
]
```

3.7.6 Device Output Event

Description: Send an output event from an existing Device for a registered Device on the Hub.

Synopsis: POST

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: payload

Optional Query parameters: Any property listed in section 1.2.4.3

Request Message Body: Device IO Event

Response Headers: Content-Length, Content-Type

Response Message Body: Name/value pairs with media type "application/json"

Response Status: 200, 400, 401, 403, 404

Example Request: Send an output event from an existing Device for a registered Device Manager on the Hub.

POST https://hub.flexeye.com/v1/iot_Default/dms/CCSR_DM/devices/135/outEvents

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{  
  "timeStamp" : "2013-09-25T14:30:34.000Z",  
  "payload" : "10.143"  
}
```

Example Response:

HTTP/1/1 201 CREATED

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
{  
  "timeStamp" : "2013-09-25T13:30:34.000Z",  
  "type" : 0.0,  
  "payload" : "10.143",  
}
```

```
"device" : {  
  "uri" :  
  "https://localhost:8181/v1/iot_Default/dms/CCSR_DM/devices/135/outEvents/3054  
9/device"  
  },  
  "id" : "30549",  
  "uri" :  
  "https://localhost:8181/v1/iot_Default/dms/CCSR_DM/devices/135/outEvents/3054  
9"  
}
```

3.7.7 Device Input Event

Description: Send an input event from an existing Device for a registered Device on the Hub

Synopsis: POST

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: payload

Optional Query parameters: Any property listed in section 1.2.4.3

Request Message Body: Device IO Event

Response Headers: Content-Length, Content-Type

Response Message Body: Name/value pairs with media type "application/json"

Response Status: 200, 400, 401, 403, 404

Example Request: Receive an input event to an existing Device from the Hub.

POST https://hub.flexeye.com/v1/iot_Default/allDevices/125/inEvents
Host: hub.flexeye.com
Authorization: **Basic authentication**
Accept: application/json
Content-Length: **nnn**
Content-Type: application/json

```
{  
  "timeStamp" : "2013-09-25T14:30:34.000Z",  
  "payload" : "10.1234"  
}
```

Example Response:

HTTP/1/1 201 CREATED
Content-Type: **application/vnd.com.flexeye.iot+json**
Content-Length: **nnn**

```
{  
  "timeStamp" : "2013-09-25T13:30:34.000Z",  
  "type" : 1.0,  
}
```

```
"payload" : "10.1234",
"device" : {
  "uri" :
    "https://localhost:8181/v1/iot_Default/allDevices/125/inEvents/30546/device"
},
"id" : "30546",
"uri" :
  "https://localhost:8181/v1/iot_Default/allDevices/125/inEvents/30546"
}
```

3.7.8 Permit Devices

Description: Use 3.7.2 Update Device to change permissions

3.8 Application (Service)

3.8.1 Get Application

Description: Retrieves the current state of the Application (Service).

Synopsis: GET

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: N/A

Optional Query parameters: Any property listed in section 1.2.7

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Get the current state of the Application (Service) named "myGuardian"

GET https://hub.flexeye.com/v1/iot_Default/applications/myGuardian

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: N/A

Content-Length: N/A

Content-Type: N/A

Example Response

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
{
  "name" : "myGuardian",
  "id" : "myGuardian",
  "owner" : {
    "uri" :
"https://localhost:8181/v1/iot_Default/applications/myGuardian/owner"
  },
  "numUsers" : 22.0,
  "uri" : "https://localhost:8181/v1/iot_Default/applications/myGuardian"
}
```

3.8.2 Register Device

Description: Register an existing Device with an Application (Service).

Synopsis: POST

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: id

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Registers a Device named "225" with an Application (Service) named "myGuardian"

POST https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/devices

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{
  "id" : "225"
}
```

Example Response:

HTTP/1/1 201 CREATED

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
{
  "devices" : [ {
    "uri" :
"https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/device
s/134",
    "key" : "134"
  }, {
    "uri" :
"https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/device
s/225",
    "key" : "225"
  } ],
  "devicesCursor" : {
    "count" : "0",
    "nextUri" : null,
    "prevUri" : null
  }
}
```

3.8.3 De-Register Device

Description: De-register a device from an Application (Service).

Synopsis: PUT

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: Register flag

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: De-Registers a Device named "135" from an Application (Service) named "myGuardian"

PUT

https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/devices/135

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{
  "isRegistered": "false"
}
```

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
{
  "name" : "135",
  "id" : "135",
  "lastSynchronised_t" : "2013-11-08T10:50:15.244Z",
  "statusTV" : "NaN",
  "lastKnownLocation" : "CCSR / 20BA02",
  "isRegistered" : 0.0,
  "description" : "updatedDesc",
  "uri" :
  "https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/devices/135"
}
```

3.8.4 Get Events

Description: Returns a list of Events for an Application's device.

Synopsis: GET

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: If no query parameters are passed all events are returned.

Optional Query parameters: Any property listed in section 1.2.4.3

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: Array of Events

Response Status: 200, 400, 401, 403, 404

Example Request: Get all output Events for the "myGuardian" device "135".

GET

https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/devices/135/inOutEvents

Query: type=0.0 (Output)

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: N/A

Content-Length: N/A

Content-Type: N/A

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
{
  "inOutEvents" : [ {
    "uri" :
    "https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/device
s/135/inOutEvents/30493",
    "key" : "30493"
  }, {
    "uri" :
    "https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/device
s/135/inOutEvents/30539",
    "key" : "30539"
  } ],
  "inOutEventsCursor" : {
    "count" : "0",
    "nextUri" : null,
    "prevUri" : null
  }
}
```

3.8.5 Get Users

Description: Returns a list of users for the Application.

Synopsis: GET

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: If no query parameters are passed all users are returned.

Optional Query parameters: Any property listed in section 1.2.3

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Type, Content-Language, Date, Expires, Last-Modified, Cache-Control, Content-Length

Response Message Body: Array of Users

Response Status: 200, 400, 401, 403, 404

Example Request: Get all Users for the Application "myGuardian" whose first name is "John"

GET https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users

Query: firstName=John

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: N/A
Content-Length: N/A
Content-Type: N/A

Example Response:

HTTP/1/1 200 OK
Content-Type: application/vnd.com.flexeye.iot+json
Content-Length: nnn

```
[
  {
    "users" : [ {
      "uri" :
      "https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/
      JohnSmith",
      "key" : "JohnSmith"
    }, {
      "uri" :
      "https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/
      JohnDoe",
      "key" : "JohnDoe"
    } ],
    "usersCursor" : {
      "count" : "0",
      "nextUri" : null,
      "prevUri" : null
    }
  }
]
```

3.8.6 Add User

Description: Adds a Person as a user of the Application. If the Person does not exist then creates a new Person managed by the Hub's default Person Manager (see also *Person Manager* below).

Synopsis: POST

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: Any property listed in section 1.2.6.

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Type, Content-Language, Date, Expires, Last-Modified, Cache-Control, Content-Length

Response Message Body: User object

Response Status: 200, 400, 401, 403, 404

Example Request: Adds a Person with id "user121" as a user of the Application (Service) named

"myGuardian".

POST https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{
  "id" : "user121",
  "firstName" : "test_firstName",
  "lastName" : "test_lastName",
  "emailId" : "test_emailId",
  "password" : "test_password"
}
```

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
{
  "id" : "user121",
  "authName" : "user121",
  "password" : "##9fb7fe1217aed442b04c0f5e43b5d5a7d3287097",
  "authenticator" : 1.0,
  "phoneNumber" : "",
  "emailId" : "test_emailId",
  "defaultTrustee" : {
    "uri" : null
  },
  "firstName" : "test_firstName",
  "lastName" : "test_lastName",
  "myPersonalDevices" : {
    "myPersonalDevices" : [ ],
    "myPersonalDevicesCursor" : {
      "count" : "0",
      "nextUri" : null,
      "prevUri" : null
    }
  },
  "uri" :
"http://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/user121"
}
```

3.8.7 Authenticate User

Description: Authenticates a user of the Application (by calling through to their Person Manager)

Synopsis: POST (Operation)

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: username, password

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Authenticates a user named "user121" of the Application (Service) named "myGuardian".

POST

https://hub.flexeye.com:8181/v1/iot_Default/applications/myGuardian/authenticateUser?username=<username>&password=<password>

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{
    "username" : "user121",
    "password" : "xxxxxx"
}
```

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
{
    "results" : "True",
    "operation" : "authenticateUser"
}
```

3.8.8 Login User

Description: Authenticates a user of the Application and establishes a session for that user with the Application.

Synopsis: POST (Operation)

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: username, password

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Authenticates a user named "user121" of the Application (Service) named "myGuardian" and establishes a session for that user with the Application.

POST

https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/user1/loginUser

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{
  "username" : "user121",
  "password" : "xxxxxx"
}
```

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
{
  "results" : "True",
  "operation" : "loginUser"
}
```

3.8.9 Logout User

Description: Closes an open session with the Application.

Synopsis: POST (Operation)

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: sessionID

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Closes an open session with the Application (Service) named "myGuardian".

EyeHub: Service APIs v2.2

```
POST
https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/user1/logoutUser
Host: hub.flexeye.com
Authorization: Basic authentication
Accept: application/json
Content-Length: nnn
Content-Type: application/json
```

```
{
  "sessionID" : "24e0945aad6f0869a77ccab063d7fdfd"
}
```

Example Response:

```
HTTP/1/1 201 OK
Content-Type: application/vnd.com.flexeye.iot+json
Content-Length: nnn
```

```
{
  "results" : "True",
  "operation" : "logoutUser"
}
```

3.9 Timer

3.9.1 Get Timer

Description: Retrieves the current state of the Timer.

Synopsis: GET

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: N/A

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Type, Content-Language, Date, Expires, Last-Modified, Cache-Control, Content-Length

Response Message Body: Timer object

Response Status: 200, 400, 401, 403, 404

Example Request: Get the current state of the Timer named "T-348".

GET

`https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/JoeBloggs/timers/T-348`

Host: `hub.flexeye.com`

Authorization: `Basic authentication`

Accept: `N/A`

Content-Length: `N/A`

Content-Type: `N/A`

Example Response:

`HTTP/1/1 200 OK`

Content-Type: `application/vnd.com.flexeye.iot+json`

Content-Length: `nnn`

```
{
  "duration" : "NaN",
  "state" : 2.0,
  "id" : "T-105",
  "notifyCoverage" : "NaN",
  "logMessage" : null,
  "notifyType" : "NaN",
  "startTime" : "2013-10-23T13:17:01.724Z",
  "lastRestartTime" : "NaN",
  "durationExpired" : "NaN",
  "notificationPeople" : null,
  "uri" :
  "https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/JoeBloggs/timers/T-348"
}
```

3.9.2 Pause Timer

Description: Pauses a Timer (operation).

Synopsis: POST

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: N/A

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Type, Content-Language, Date, Expires, Last-Modified, Cache-Control, Content-Length

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Pauses a Timer named "T-348".

POST

https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/JoeBloggs/timers/T-348/pause

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
{
  "results" : "True",
  "operation" : "stop"
}
```

3.9.3 Resume Timer

Description : Resumes a paused Timer (operation).

Synopsis: POST

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: N/A

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Type, Content-Language, Date, Expires, Last-Modified, Cache-

Control,Content-Length

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Resumes a paused Timer named "T-348"

POST

https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/JoeBloggs/timers/T-348/resume

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
{
  "results" : "True",
  "operation" : "resume"
}
```

3.9.4 Stop Timer

Description: Stops a running Timer before it expires. If called after the Timer has expired, then the signature has no effect.

Synopsis: POST (Operation)

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: N/A

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Type, Content-Language, Date, Expires, Last-Modified, Cache-Control, Content-Length

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Stops a running Timer named "T-348" before it expires.

```
POST
https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2/timers/T-348/stop
Host: hub.flexeye.com
Authorization: Basic authentication
Accept: application/json
Content-Length: nnn
Content-Type: application/json
```

Example Response:

```
HTTP/1/1 200 OK
Content-Type: application/vnd.com.flexeye.iot+json
Content-Length: nnn
```

```
{
  "results" : "True",
  "operation" : "stop"
}
```

3.9.5 Delete Timer

Description: Deletes a Timer

Synopsis: DELETE

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: N/A

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Deletes a Timer named "T-348"

DELETE

https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2/timers/T-348

Host: hub.flexeye.com

Authorization: Basic authentication

Example Response:

HTTP/1/1 204 No Content

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

3.9.6 Expire Timer

Description: Changes a Timer's state to expired before it expires automatically.

Synopsis: POST (Operation)

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: N/A

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Type, Content-Language, Date, Expires, Last-Modified, Cache-Control, Content-Length

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Changes the state of a Timer named "T-348" to expired.

POST

https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2/timers/T-348/expire

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
{  
  "results" : "True",  
  "operation" : "expire"  
}
```

3.9.7 Update Timer

Description: Updates an existing Timer object

Synopsis: PUT

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: Any property listed in section 1.2.8

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Updates the Timer "T-348".

PUT

https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2/timers/T-348

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

```
{
  "lat" : "-1.12345",
  "long" : "1.12345",
  "batteryLevel" : "1.5",
  "defaultTrustee" : "JohnSmith",
  "startTagLocation" : "UniS Library",
  "endTagLocation" : "Surrey Sports Park",
  "checkInLocation" : "Starbucks"
}
```

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
{
  "uri" :
  "https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2/timers/T-348",
  "id" : "T-348",
  "lat" : "-1.12345",
  "long" : "1.12345",
}
```

```
"batteryLevel" : "1.5",  
"defaultTrustee" : "JohnSmith",  
"startTagLocation" : "UniS Library",  
"endTagLocation" : "Surrey Sports Park"  
}
```

3.10 Notification

3.10.1 Get Notification

Description: Retrieves the current state of the Notification.

Synopsis: GET

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: N/A

Optional Query parameters: Any property listed in section 1.2.9

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: Notification object

Response Status: 200, 400, 401, 403, 404

Example Request: Get the current state of the Notification named "N-1070"

GET https://hub.flexeye.com/v1/iot_Default/allPeople/gs/notifications/N-1070

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: N/A

Content-Length: N/A

Content-Type: N/A

Example Response:

HTTP/1/1 200 OK

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
{
  "id" : "N-1070",
  "timeStamp" : "2013-09-24T00:00:00.000Z",
  "message" : "Started from Guildford",
  "type" : 1.0,
  "status" : 1.0,
  "uri" :
  "https://hub.flexeye.com/v1/iot_Default/allPeople/gs/notifications/N-1070"
}
```

3.10.2 Delete Notification

Description: Deletes a Notification

Synopsis: DELETE

Request Headers: Accept, Authorization, Content-Length, Content-Type

Request Parameters: N/A

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Deletes a Notification named "N-1070"

DELETE https://hub.flexeye.com/v1/iot_Default/allPeople/gs/notifications/N-1070

Host: hub.flexeye.com

Authorization: Basic authentication

Example Response:

HTTP/1/1 204 No Content

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

4. Appendices

4.1 This Document

4.1.1 Audience

This document is for the use of any stakeholder involved in the TSB IoT EyeHub project.

4.1.2 Scope

This document covers the service APIs for the Hub resources i.e. as Location Manager, Person Manager and Person, Location, Device, etc. as described in *EyeHub Specification v0.9* document.

4.1.3 Document Purpose

The purpose of this document is to specify the EyeHub software components that will be delivered as part of WP1 of the TSB IoT EyeHub demonstrator project. As such the document will go through several stages during the course of the project:

- 1. Draft:** The first version, or draft version, is compiled after requirements have been discovered, recorded, classified, and prioritized, analysed and mapped to original proposed Hub components and applications for managing IoT devices, locations and people.
- 2. Proposed:** The draft document is then proposed as a Hub Service API specification for the project. The proposed document should be reviewed by several parties, who may comment on any content and any priorities, either to agree, to disagree, or to identify missing material. Readers include end-users, developers, project managers, and any other stakeholders. The document may be amended and re-proposed several times before moving to the next stage.
- 3. Approved:** The proposed document is accepted by representatives of each party of stakeholders as an appropriate input for the WP1 milestone MS1 Initial Hub Specification Completed of the EyeHub project. The integration team then use the document for the delivery of the Entity external API to the Hub system.

4.2 Notification Message Templates - myGuardian applications.

The following templates exist for the body portions of notification messages sent from the hub to a user of the myGuardian application. The following replacements are made within these message bodies:

<trustor> is the name of a user that has trustees.

<trustee> is the name of the trustee.

Message Template Id (used as a parameter for 3.4.18 and 3.4.18.5)	Message Content
TrusteeInvitation	Trustee Invitation: You have been invited to be a trustee for <trustor>
TrusteeInvitationAcceptance	Invitation Accepted: <trustee> has accepted your invitation to be your trustee
TrusteeInvitationRejection	Invitation Rejected: <trustee> has rejected your invitation to be your trustee
TrusteeRemoval	Trustee Removed: You were removed from the list of trustee of <trustor>
InformJourneyTrustee	Journey Trustee: You have been invited to be a journey trustee for <trustor>
TimerExpiry	Your trustor <trustor> has undertaken a journey but not reached or checked into their destination within the expected journey time.
TimerStart	Your trustor <trustor> has started a journey.
TimerPause	Your trustor <trustor> has paused a journey they are on.
TimerStop	Your trustor <trustor> has stopped a journey they are on.

SafetyAlert	Your trustor <trustor> has raised a safety alert.
-------------	---

4.3 Administrator Signatures

The following signatures require an administrator user to be used for the basic authentication of the REST request. Signatures outside this list can be run using a specific user who is created for use with an application with the hub.

3.1 Hub

- 3.1.5 Create Person
- 3.1.7 Create Location
- 3.1.12 Register Device Manager
- 3.1.13 Deregister Device Manager
- 3.1.14 Register Application
- 3.1.15 Deregister Application
- 3.1.16 Register Location Manager
- 3.1.17 Deregister Location Manager
- 3.1.18 Register Person Manager
- 3.1.19 Deregister Person Manager

3.2 Person Manager

- 3.2.5 Create User
- 3.2.5.1 Register User

3.3 Location Manager

- 3.3.2 Create Location

4.4 Interim Collection Signatures

4.4.1 Search Locations

Description: Returns the Hub locations (including all the properties that exist on the location class), filtering the list based on occurrences of two optionally passed parameters with the location id string and the location name string.

Synopsis: POST (Operation)

Request Headers: Accept, Authorization, Content-Length, Content-Type

Query Parameters: (Optional, case insensitive) idFilter, nameFilter.

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Searches the hub locations for members whose "id" property contains (anywhere within it) the text "CCSR".

POST https://hub.flexeye.com/v1/iot_Default/searchLocations?idFilter=CCSR

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

Example Response:

HTTP/1/1 201 CREATED

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
{
  "operation" : "searchLocations",
  "results" : [ {
    "name" : "CCSR",
    "id" : "CCSR",
    "owner" : {
      "uri" : null
    },
    "lat" : "51.243587899999999",
    "long" : "-0.5881084999999975",
    "uri" : "http://hub.flexeye.com/v1/iot_Default/allLocations/CCSR"
  } ]
}
```

4.4.2 Search User Trustees

Description: Returns a user's trustees (including all the properties that exist on the trustee (Person) object), filtering the list based on occurrences of two optionally passed parameters with the trustee's id string and the trustee's name string.

Synopsis: POST (Operation)

Request Headers: Accept, Authorization, Content-Length, Content-Type

Query Parameters: (Optional, case insensitive) idFilter, nameFilter.

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Searches the trustees of user "pmr2" whose "id" property contains (anywhere within it) the text "gs".

POST

https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2/searchTrustees?idFilter=gs

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

Example Response:

HTTP/1/1 201 CREATED

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
{
  "id" : "",
  "name" : null,
  "authName" : "",
  "password" : "##0f37b93b7a6bcc71004969ff58b3a9537c9485d0",
  "authenticator" : 1.0,
  "security" : 3.0,
  "authMethod" : {
    "uri" : null
  },
  "uri" :
"http://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2/"
}
```

4.4.3 Search User Notifications

Description: Returns a user's notifications (including all the properties that exist on the notification object), filtering the list based on id and medium.

Synopsis: POST (Operation)

Request Headers: Accept, Authorization, Content-Length, Content-Type

Query Parameters: (Optional, case insensitive) idFilter, mediumFilter (for possible values see "medium" under 1.2.9)

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Searches the notifications of user "pmr2" whose "id" property contains (anywhere within it) the text "N-1777" and whose medium is 3.0 (Google Cloud Messaging).

POST

https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2/searchNotifications?idFilter=N-1777&mediumFilter=3.0

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

Example Response:

HTTP/1/1 201 CREATED

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
{
  "operation" : "searchNotifications", "results" : [ {
    "timeStamp" : "2014-01-15T11:06:49.560Z",
    "message" : "SafetyAlert",
    "medium" : 3.0,
    "status" : "NaN",
    "id" : "N-1777",
    "notifyPerson" : {
      "uri" : null
    },
    "sender" : null,
    "subject" : null,
    "timer" : {
      "uri" : null
    },
  },
```

```
    "type" : null,  
    "notifyType" : {  
      "uri" : "http://hub.flexeye.com/v1/iot_Default/applications/myGuardia  
n/users/pmr2/notifications/002d5e062eff498e75a24d48dbcf1514/notifyType"  
    },  
    "uri" : "http://hub.flexeye.com/v1/iot_Default/applications/myGuardian/  
users/pmr2/notifications/002d5e062eff498e75a24d48dbcf1514"  
  } ]  
}
```

4.4.4 Search User Locations

Description: Returns a user's locations (including all the properties that exist on the location object), filtering the list based on occurrences of two optionally passed parameters with the location's id string and the location's name string.

Synopsis: POST (Operation)

Request Headers: Accept, Authorization, Content-Length, Content-Type

Query Parameters: (Optional, case insensitive) idFilter, nameFilter

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Searches the locations of user "pmr2" whose "id" property contains (anywhere within it) the text "AmazonLoc"

POST

https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2/searchLocations?idFilter=amazonloc

Host: hub.flexeye.com

Authorization: Basic authentication

Accept: application/json

Content-Length: nnn

Content-Type: application/json

Example Response:

HTTP/1/1 201 CREATED

Content-Type: application/vnd.com.flexeye.iot+json

Content-Length: nnn

```
{
  "operation" : "searchLocations",
  "results" : [ {
    "name" : "AmazonLoc",
    "id" : "AmazonLoc",
    "owner" : {
      "uri" : null
    },
    "uri" : "http://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/pmr2/locations/AmazonLoc"
  } ]
}
```

4.4.5 Search Application Users

Description: Returns an application's users (including all the properties that exist on the user object), filtering the list based on occurrences of two optionally passed parameters with the user's id string and the user's name string.

Synopsis: POST (Operation)

Request Headers: Accept, Authorization, Content-Length, Content-Type

Query Parameters: (Optional, case insensitive) idFilter, nameFilter

Request Message Body: Name/value pairs with media type "application/json"

Response Headers: Content-Length, Content-Type

Response Message Body: True/False

Response Status: 200, 400, 401, 403, 404

Example Request: Searches the users list of the "myGuardian" application, whose "id" property contains (anywhere within it) the text "pmr2"

POST

`https://hub.flexeye.com/v1/iot_Default/applications/myGuardian/searchUsers?idFilter=pmr2`

Host: `hub.flexeye.com`

Authorization: `Basic authentication`

Accept: `application/json`

Content-Length: `nnn`

Content-Type: `application/json`

Example Response:

HTTP/1/1 201 CREATED

Content-Type: `application/vnd.com.flexeye.iot+json`

Content-Length: `nnn`

```
{
  "operation" : "searchUsers",
  "results" : [ {
    "id" : "pmr2",
    "name" : null,
    "authenticator" : 1.0,
    "security" : 3.0,
    "uri" :
      "http://hub.flexeye.com/v1/iot_Default/applications/myGuardian/users/002baafa
      2eff498e6bc38b3fc39f0e22"
  } ]
}
```


4.5 Team

Role	Name	Organisation
Author	Gabriel Falana	Flexeye
Reviewer	Divyang Sharma	Flexeye
Reviewer	Richie Saville	Flexeye
Reviewer/Approver	Alex Gluhak	University of Surrey
Reviewer/Approver	Georgios Michalakidis	Manageplaces

4.6 Version Control

Version	Status	Date	Author	Description of Change(s)
0.1	Draft	19/07/2013	Gabriel Falana	New document
0.2	Draft	23/07/2013	Gabriel Falana	Updated with the API for Timer
0.3	Draft	26/07/2013	Patrick Rivet	Added APIs for Application and updated description of Person API
0.4	Draft	29/07/2013	Patrick Rivet	Updated descriptions of Hub and Location API
0.5	Draft	31/07/2013	Patrick Rivet	Added the operation for Location Manager, Person Manager entities.
0.6	Draft	2/8/2013	Richie Saville	Added missing signatures, sections, updated descriptions; added FIXMEs and Comments.
0.7	Draft	9/8/2013	Richie Saville	Moved users and auth signatures onto Application object.
0.8	Draft	16/8/2013	Patrick Rivet	Added JSON description for all Entity Objects
0.9	Draft	06/9/2013	Richie Saville	REST update of signatures
0.91	Draft	29/10/2013	Patrick Rivet	Added Rest API Interface URI and requests section.
1.0	Release	12/11/2013	Patrick Rivet	Corrected REST request and response examples. Removed FIXMEs (maintained in previous version)
1.1	Release	15/11/2013	Patrick Rivet	Updated the Resource definition (section

				<p>1.2) with properties: server, scheme, host, port, path, apiCallBkFunc, represents and representedBy.</p> <p>Updated REST request path for Device IO Input (section 3.7.7) to “/inEvent” instead of “/inputs”.</p>
1.2	Release	18/11/2013	Patrick Rivet	<p>Event creation was changed from “events” to “inEvents” and “outEvent”, and “allEvents” accessed via new property “inOutEvents”.</p>
1.3	Release	29/11/2013	Divyang Sharma	<p>1. Corrections made to the description of Device IO Event (In) and Device IO Event (Out).</p> <p>2. Added to the Device Manager section, the Device manager Owner specific function Device IO Event (In).</p> <p>3. Updated the examples and response messages for subscribe and unsubscribe APIs.</p>
1.4	Release	09/12/2013	Patrick Rivet	<p>(1) Section 1.2.9: Notification “type” renamed to “medium” and added “type” back in as a string field within the “might contain” table.</p> <p>(2) 3.4.5 Parameter changed from “trustees” to “proposedTrustees”.</p> <p>(3) 3.4.19 and 3.4.20: Changed operation names changed from “accept” and “reject” to “acceptTrustee” and “rejectTrustee” respectively.</p> <p>(4) Added signatures 3.9.6, 3.9.7.</p>
1.5	Release	13/12/2013	Patrick Rivet	<p>Added definition and signatures for a person’s mobile devices; sections 3.4.23,</p>

				3.4.24 and 1.2.11. Updated 3.4.18 and added 3.4.18.5.
1.6	Release	16/12/2013	Patrick Rivet	<p>Updated:-</p> <p>1.2.5 Location; “QRTag” renamed to “qrTag”.</p> <p>3.9.7 - example request parameters</p> <p>3.9.4 - example request URI</p> <p>1.2.8 Timer Object: added “startLocation”, “endLocation”, “startTagLocation” & “endTagLocation”.</p> <p>Signature 3.4.16 changed to 3.4.16.5</p> <p>3.4.18.5: Changed Query Parameters to be: msg.</p> <p>1.2.8: Added “logMessage” as mandatory and moved “notifyType” to optional (as this is governed by the trustee’s person device notification type).</p> <p>Added:-</p> <p>3.4.2.1 Change Password</p> <p>3.2.5.1 Register User</p> <p>3.1.22 Get Location Tag</p> <p>1.2.12 Location Tag Object</p> <p>3.4.1.5 Get Person Last Status</p>
1.7	Release	24/12/2013	Vamsi Nama	<p>1. Modified section 2.5.2.1 for the string function list</p> <p>2. Modified 3.4.5 Add Trustee</p> <p>3. Modified 3.4.6 Remove Trustee</p> <p>4. Modified 3.4.7 Set Default Trustee</p>
1.8	Release	24/12/2013	Vamsi Nama	<p>1. Added 3.4.8.1 Search Campus Location</p>

1.9	Release	02/Jan/2014	Pat Rivet	<p>Added Appendix 4.2: Notification Message Templates.</p> <p>3.4.4: Updated example request.</p> <p>3.4.5: Changed the signature to automatically send notification.</p> <p>Across all sections apart from 3.2, altered the example request URLs to use path relative to /iot_Default/applications/ instead of /iot_Default/pms/</p> <p>Added Appendix 4.3 Admin Signatures.</p> <p>1.2.11: Updated “imei” property as optional.</p> <p>3.8.6: Updated Example request to show mandatory properties need to be passed when creating a user.</p> <p>1.2.6: Updated mandatory properties. Removed “uri” and “phoneNumber”, added “id”, “emailId” and “password”.</p>
2.0	Release	09/Jan/2014	Pat Rivet	<p>3.4.14: Updated signature path from “inEvents” to “events”. Example request now shows setting type to an input event (type=1.0).</p> <p>3.4.15: Updated signature path from “outEvents” to “events”. Example request now shows setting type to an output event (type=0.0).</p> <p>1.2.8: Updated with “checkInLocation” and</p>

				<p>“currentLocation”. Deprecated “logMessage”.</p> <p>3.4.16: Changed the example request, as only “notifyType” and “duration” are mandatory parameters.</p> <p>4.2 Added message templates “TimerExpiry” and “SafetyAlert”.</p> <p>3.4.22: Changed description and example request.</p>
2.1	Release	13/Jan/2014	Pat Rivet	<p>3.2.5.1: Modified the Query parameters: “phoneNumber” was changed to “telephoneNumber”. “password” was changed to “passcode”. “confirmPassword” was changed to “confirmPasscode”.</p> <p>3.4.15: Correction to the path used in the example request “Events” changed to “events”.</p> <p>3.4.8.1: Changed the request example.</p> <p>1.2.8: Timer Object: Amended the description of “duration” and “notifyCoverage”. Added properties: “onStartNotify”, “onPauseNotify” and “onStopNotify”.</p> <p>1.2.9 Notification Object: Added “id” property.</p>

				<p>Amended “payload” to “message” - as “payload” does not exist.</p> <p>Added “title” property.</p> <p>Amended “type” to “notifyType”.</p> <p>Deprecated the “notifyPeople” property.</p> <p>4.4 Interim Collection Signatures: Added this new appendix.</p> <p>3.4.16 Start Timer: Defined the default values set for properties.</p> <p>1.2 Hub Entity: definition updated with new property “imageUrl”.</p> <p>1.2.7 Application Object: Added “notifications” property.</p> <p>4.2: Notification Message Templates: Added message templates: “TimerStart”, “TimerPause” and “TimerStop”.</p>
2.2	Release	28/Jan/2014	Pat Rivet	<p>1.2.8: Removed deprecated properties; “notifyType” and “notificationPeople”.</p> <p>1.2.8: Restored property “logMessage” from deprecated to in use; moved to “May contain” table.</p> <p>1.2.9: Removed deprecated property; “notifyPeople”.</p> <p>1.2.5: Moved properties starting with</p>

				<p>“addr”, “lat” and “long” into the “may have” table.</p> <p>3.4.18: Query Parameters -added “timerId” and “senderText”.</p> <p>3.4.18.5: Query Parameters -added “senderText”.</p>
--	--	--	--	--

