EyeHub: Service APIs

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

Prepared by Flexeye Technology Ltd
Draft Version: 1.1

Table of Contents

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
"events"	List of Event items managed	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 multiple devices.	JSON array of zero or more Resource objects
"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	JSON string

	endpoint.	
"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

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 Device Manager item endpoint on the Owner side	URI as a JSON string
"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
"events"	List of Events managed by device manager	JSON array of zero or more Event 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 *enter* the Hub.	JSON array of DM Permission objects
"allEvents"	List of Events across the device manager's devices (GET requests only)	JSON array of zero or more Event 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 *enter* 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
"events"	List of Event items managed	JSON array of zero or more Event 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 *enter* 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	JSON string

	is "1"	
"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
"events"	List of Event items managed by Person Manager Item	JSON array of zero or more Event 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 *enter* 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
`"entryDate"`	Registration date of the device	ISO 8601 date as JSON string
`"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
`"events"`	List of events generated by the Device	JSON array of zero or more Event objects
`"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	JSON array of one or more JSON strings The values "ALL" and "NONE" (not case-sensitive) are reserved words. Remaining values are

	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 subscribers.	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
"ownerParentId"	Unique identifier for the parent Device item within the scope of the Device Manager - defined by the Owner.	JSON string
"allEvents"	List of Events generated by Device and Hub (GET requests only)	JSON array of zero or more Event objects
"hubRefEvents"	List of Events generated by Hub for this Device object (GET requests only)	JSON array of zero or more Event objects
"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 rel 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

`"type"`	Enumerated string value that defines the type of the event = {(0,Output),(1,Input),(2,Both)}	Name of event type as JSON string
`"payload"`	The data pertaining to the Device I/O event.	Event payload as JSON string
`"source"`	Defines the path when event is originated from the source.	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
`"location"`	Defines location of the Event object	JSON Location object
`"person"`	Defines Person of the Event object	JSON Person object
`"subDevice"`	Used to get Sub Device item(Event managed by Sub Device) (GET requests only)	JSON Device object
`"device"`	Used to get Device item(Event managed by Device) (GET requests only)	JSON Device object
`"dms"`	Used to get Device Manager item(Event managed by Device Manager's devices) (GET requests only)	JSON Device Manager object
`"personManager"`	Used to get Person Manager item(Event managed by Person Manager)	JSON Device Manager object

16

	(GET requests only)	

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
`"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

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"`	QRCode 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
`"events"`	List of events generated at the location	JSON array of zero or more Event objects
`"subscribed"`	List of names whose values indicate whether there are subscribers to the Location 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 Location. The value can be a link to a list of subscribers.	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 Location I/O interface.
`"ownerParentId"`	Unique identifier for the parent Location item within the scope of the Location Manager - defined by the Owner.	JSON string

`"allEvents"`	List of Events generated by Location and Hub (GET requests only)	JSON array of zero or more Event objects
`"hubRefEvents"`	List of Events generated by Hub for this Location object (GET requests only)	JSON array of zero or more Event objects
`"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

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
"uri"	Uniform Resource Identifier for the Person item within the Hub.	URI as 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
"phoneNumber"	Defines contact number of the person	JSON Number

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
"password"	Defines the password 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

`"events"`	List of events generated by the person	JSON array of zero or more Event objects.
`"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
`"trusteeInvitati ons"`	List of trustee invitations managed by the user	JSON array of one or more Person objects
`"rejectedTrustee s"`	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"`	List of names whose values indicate whether there are subscribers to the person 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 Person. The value can be a link to a list of subscribers.	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 Person I/O interface.
`"allEvents"`	List of Events generated by Person and Hub (GET requests only)	JSON array of zero or more Event objects

`"hubRefEvents"`	List of Events generated by Hub for this Person object (GET requests only)	JSON array of zero or more Event objects

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.
`"events"`	List of events generated by the user to the Application (Service)	JSON array of zero or more event 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
`"allEvents"`	List of Events generated by Application and Hub (GET requests only)	JSON array of zero or more Event objects
`"hubRefEvents"`	List of Events generated by Hub for this Application object (GET requests only)	JSON array of zero or more Event objects
`"numUsers"`	Total number of users managed by Application (GET requests only)	JSON Number

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"`	duration defines the interval between states of Timer	JSON Number
`"state"`	Enumerated string value that defines the state of timer= {(0,Ticking), (1,Paused), (2,Stopped), (3,Expired)}	JSON string
`notifyType`	Enumerated string value that defines the type of timer notification = {(0,Email), (1,SMS), (2,ApplePushNotification), (3,GoogleCloudMessaging)}	JSON string
`notifyCoverage`	Enumerated string value that defines the people coverage for timer notification = {(0,None),	JSON string

Property Name	Definition	Property Value
	(1,DefaultTrustee), (2,AllTrustees)}	

A Timer object MAY contain the following properties:

Property Name	Definition	Property Value
`"id"`	Unique Identifier for the Timer object	JSON string
"startTime"	The Timer start time E.g. "2013-06-11T13:15:30Z"	Date value in restricted ISO 8601 format* as JSON string
"lastRestartTime"	The Timer resume time E.g. "2013-06-11T16:45:20Z"	Date value in restricted ISO 8601 format* as JSON string
`"logmessage"`	The message content that is sent when the Timer expires	JSON string
`"notificationPeople"`	The uri of people to be notified about the Timer state	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
`"type"`	Enumerated string value that defines the type of notification= {(0,Email), (1,SMS), (2,ApplePushNotification), (3,GoogleCloudMessaging)}	JSON string
`"payload"`	Defines the Notification message	JSON string
`"notifyPeople"`	List of users to receive the notification(s)	JSON array of one or more users.

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
"notifyPerson"	Tthe 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

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

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.	This parameter filters the attributes that are returned. To include a subset of the members of an object property, include the member name (as object.member) instead of the object property name. By default, all properties are returned.

Examples:

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

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

2.5.2.1 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"

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 applicabe) 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&param2=value2" below:

```
POST
https://config28.flexeye.com/v1/iot_Default/some_operation?param1=value1&param2=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
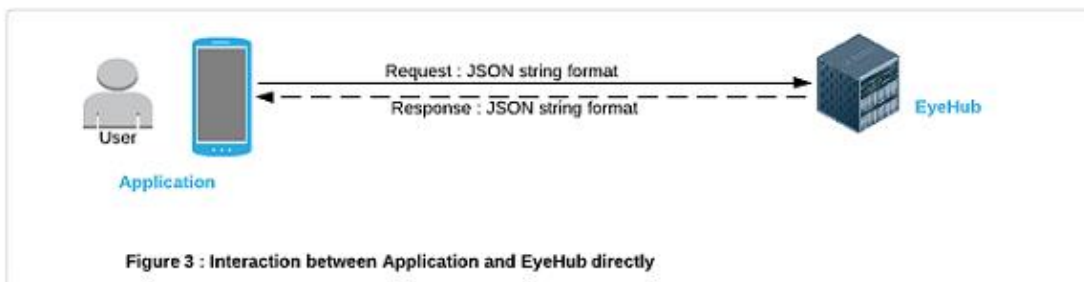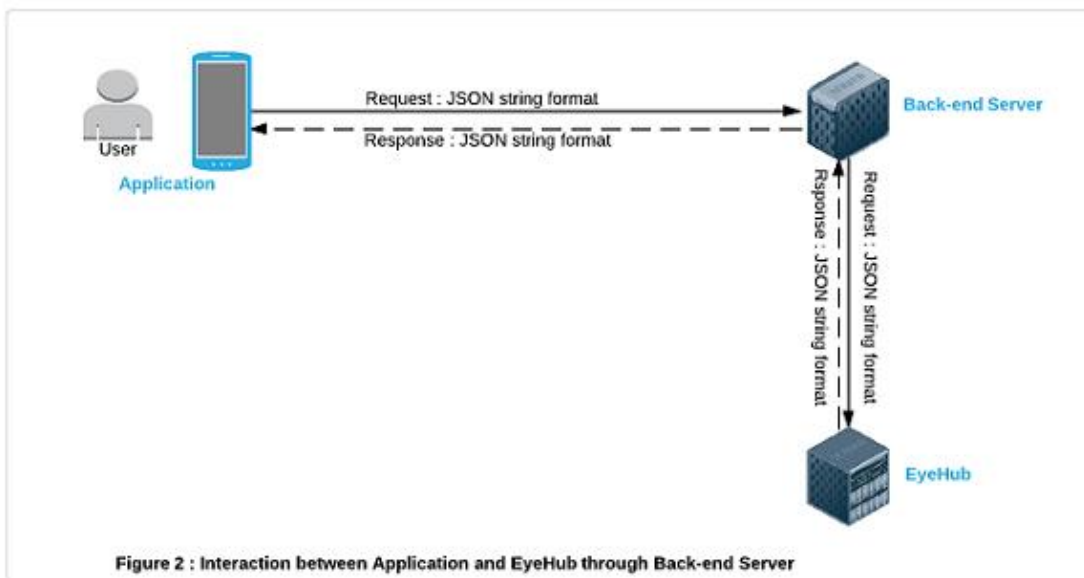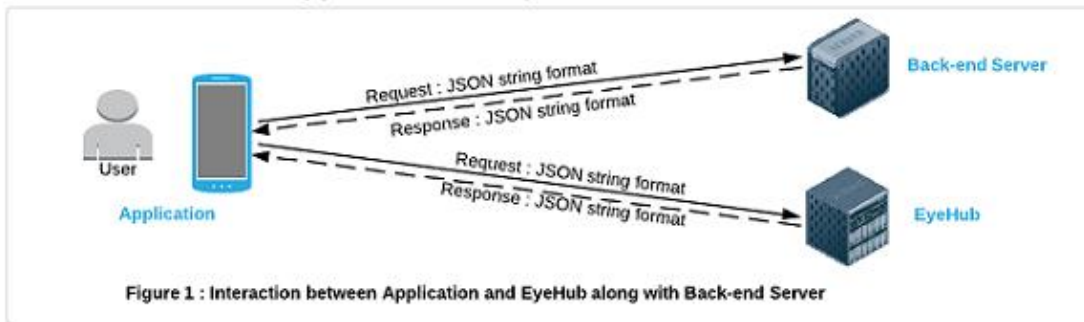
3. Service APIs

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
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://config28.flexeye.com/v1/iot_Default/authenticate
Host: config28.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

{
      "Result" : "True"
}
```

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://config28.flexeye.com/v1/iot_Default/login
Host: config28.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


{
     "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://config28.flexeye.com/v1/iot_Default/logout
```

```
Host: config28.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


{
     "Result" : "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://config28.flexeye.com/v1/iot_Default/allPeople
Query: firstName=John
Host: config28.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://config28.flexeye.com/v1/iot_Default/allPeople/John_Person_1",
        "key" : "John_Person_1"
      }, {
        "uri" :
"https://config28.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: id,firstName, lastName, emailId, password, phoneNumber
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://config28.flexeye.com/v1/iot_Default/pms/dpm/pmUsers
Host: config28.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://config28.flexeye.com/v1/iot_Default/pms/dpm/pmUsers/user1",
      "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://config28.flexeye.com/v1/iot_Default/allLocations
Query: addr_postcode=GU24HJ
Host: config28.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://config28.flexeye.com/v1/iot_Default/allLocations/CCSR",
        "key" : "CCSR"
      }, {
        "uri" :
"https://config28.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: name, 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://config28.flexeye.com/v1/iot_Default/lms/dlm/locations
Host: config28.flexeye.com
Authorization: Basic authentication
Accept: application/json
Content-Length: nnn
Content-Type: application/json


{
     "name" : "location1",
     "lat" : "90E",
     "long" : "87S",
     "addr_building" : "bridgehouse",
     "addr_road" : "millmead",
     "addr_other" : "Near George Abbort",
     "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://config28.flexeye.com/v1/iot_Default/lms/dlm/locations/location
     1",
```

```
        "name" : "location1",
        "lat" : "90E",
        "long" : "87S",
        "addr_building" : "bridgehouse",
        "addr_road" : "millmead",
        "addr_other" : "Near George Abbort",
        "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://config28.flexeye.com/v1/iot_Default/allEvents
Query: locationID=CCSR
Host: config28.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://config28.flexeye.com/v1/iot_Default/allEvents/Event1",
        "key" : "Event1"
      }, {
        "uri" :
    "https://config28.flexeye.com/v1/iot_Default/allEvents/Event2",
        "key" : "Event2"
      } ],
    "allEventsCursor" : {
        "count" : "14735",
        "nextUri" :
    "https://config28.flexeye.com/v1/iot_Default/allEvents?count=20&start=2
    0",
        "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: time, eventType, payLoad, deviceID (optional), locationID (optional), personID (optional), applicationID (optional)
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://config28.flexeye.com/v1/iot_Default/hubEvents
Host: config28.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",
    "device" : "135",
    "person" : "testP",
```

```
     "application" : "myGuardian"

}
```

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",
      "displayTimeStamp" : "2013-11-07T16:45:46.307Z",
      "source" : "135",
      "device" : {
      "uri" :
    "https://config28.flexeye.com/v1/iot_Default/hubEvents/30538/device"
      },
      "subDevice" : {
        "uri" : null
      },
      "dms" : {
        "uri" :
    "https://config28.flexeye.com/v1/iot_Default/hubEvents/30538/dms"
      },
      "id" : "30538",
      "isUpdated" : "NaN",
      "location" : {
        "uri" :
    "https://config28.flexeye.com/v1/iot_Default/hubEvents/30538/location"
      },
      "personManager" : {
        "uri" :
    "https://config28.flexeye.com/v1/iot_Default/hubEvents/30538/personMana
    ger"
      },
      "sourceType" : "NaN",
      "checkInTime" : "NaN",
      "checkOutTime" : "NaN",
      "application" : {
        "uri" :
    "https://config28.flexeye.com/v1/iot_Default/hubEvents/30538/applicatio
    n"
      },
      "person" : {
```

```
        "uri" :
   "https://config28.flexeye.com/v1/iot_Default/hubEvents/30538/person"
      },
      "payloadDesc" : null,
      "isMostRecent" : "NaN",
      "uri" : "https://config28.flexeye.com/v1/iot_Default/hubEvents/30538"
    }
]
```

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://config28.flexeye.com/v1/iot_Default/timersAll
Query: state=3.0
Host: config28.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://config28.flexeye.com/v1/iot_Default/timers/T-394",
        "key" : "T-394"
      }, {
        "uri" : "https://config28.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:timerduration, on-expiry handlers (optional): 1. Log-custom-message (String) and/or 2. Do_Notify (Notification object)

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://config28.flexeye.com/v1/iot_Default/timers
Host: config28.flexeye.com
Authorization: Basic authentication
Accept: application/json
Content-Length: nnn
Content-Type: application/json

{

     "notifyType" : "0.0",
     "friends" : ["Jack", "Jill", "John"],
     "duration" : "3000",
     "logMessage" : "description about  the specified timer goes 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" : "NaN",
```

```
  "logMessage" : "description about  the specified timer goes here",
  "notifyType" : "0.0",
  "startTime" : "2013-11-07T17:02:04.261Z",

    "friends" : ["Jack", "Jill", "John" ],
  "friendsCursor" : {
    "count" : "0",
    "nextUri" : null,
    "prevUri" : null
  },
  "uri" : "https://config28.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://config28.flexeye.com/v1/iot_Default/dms
Host: config28.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://config28.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://config28.flexeye.com/v1/iot_Default/dms/DM2
Host: config28.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://config28.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://config28.flexeye.com/v1/iot_Default/applications
Host: config28.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://config28.flexeye.com/v1/iot_Default/applications/myGuardianTes
    t"
    }
]
```

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: 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: Deregister an existing application on the Hub

```
PUT https://config28.flexeye.com/v1/iot_Default/applications/MyGuardian
Host: config28.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://config28.flexeye.com/v1/iot_Default/applications/myGuardianTes
    t"
    }


]
```

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://config28.flexeye.com/v1/iot_Default/lms
Host: config28.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://config28.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: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: Deregister an existing Location Manager "LM1" on the Hub
```
PUT https://config28.flexeye.com/v1/iot_Default/lms/LM1
Host: config28.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://config28.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://config28.flexeye.com/v1/iot_Default/pms
```

```
Host: config28.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://config28.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: 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: Deregister an existing Person Manager "PM1" on the Hub.
```
PUT https://config28.flexeye.com/v1/iot_Default/pms/PM3
Host: config28.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://config28.flexeye.com/v1/iot_Default/pms/PM3"
      }

]
```

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://config28.flexeye.com/v1/iot_Default/pms/dpm/authenticateUser
Host: config28.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://config28.flexeye.com/v1/iot_Default/pms/dpm/loginUser
Host: config28.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"
}
```

3.2.3 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://config28.flexeye.com/v1/iot_Default/pms/dpm/logoutUser
Host: config28.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 "pm1"
```
GET https://config28.flexeye.com/v1/iot_Default/pms/pm1/pmUsers
Query: firstName=John
Host: config28.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://config28.flexeye.com/v1/iot_Default/pms/pm1/pmUsers/John_Perso
     n_1",
         "key" : "John_Person_1"
       }, {
         "uri" :
     "https://config28.flexeye.com/v1/iot_Default/pms/pm1/pmUsers/John_Perso
     n_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 "pm1"
```
POST https://config28.flexeye.com/v1/iot_Default/pms/pm1/pmUsers
Host: config28.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://config28.flexeye.com/v1/iot_Default/pms/pm1/pmUsers/user121",
      "firstName" : "user121",
      "lastName" : "user121",
      "emailId" : "user121@flexeye.com",
      "id" : "user121",
      "phoneNumber" : "1234567891"
}
```

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://config28.flexeye.com/v1/iot_Default/lms/CCSR_LM/locations
Query: addr_postcode=GU24HJ
Host: config28.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://config28.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://config28.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://config28.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 person manager "pm1".
```
GET https://config28.flexeye.com/v1/iot_Default/pms/pm1/pmUsers/user121
Host: config28.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://config28.flexeye.com/v1/iot_Default/pms/pm1/pmUsers/user121"
        }

]
```

3.4.2 Update Person

Description: Updates the state (properties) of the 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 an existing person " user121" in the hub.

```
PUT https://config28.flexeye.com/data/iot_Default/pms/pm1/pmUsers/user121
Host: config28.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://config28.flexeye.com/v1/iot_Default/pms/pm1/pmUsers/user121"
      }
]
```

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 "user121" under the Hub
```
DELETE https://config28.flexeye.com/v1/iot_Default/pms/pm1/pmUsers/user121
Host: config28.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 "user121" where their email is
"jsmith@mail.com"
```
GET
https://config28.flexeye.com/v1/iot_Default/pms/dpm/pmUsers/user121/trustees
Query: emailId=jsmith@mail.com
Host: config28.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" :
    "https://config28.flexeye.com/v1/iot_Default/pms/pm1/pmUsers/user121/tr
    ustees/testP1",
          "id" : "testP1"
        "emailId" : "jsmith@mail.com"
      } ],
      "trusteesCursor" : {
        "count" : "0",
        "nextUri" : null,
        "prevUri" : null
      }
    }

]
```

3.4.5 Add Trustee

Description: Sends an invitation to trustee, Adds the trustee when the trustee accepts the invitation
Synopsis:POST
Request Headers: Accept, Authorization, Content-Length, Content-Type
Request Parameters: trustees
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: "user121" sends an invitation to trustee named "TestUser01".
```
POST https://config28.flexeye.com/v1/iot_Default/allPeople/user121
Host: config28.flexeye.com/
Authorization: Basic authentication
Accept: application/json
Content-Length: nnn
Content-Type: application/json

{
      "trustees" : "TestUser01"
}
```

Example Response:

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


[
    {
      "id" : "user121",
      "name" : null,
      "phoneNumber" : "",
      "emailId" : "user121@flexeye.com",
      "trustees" : [ {
        "uri" :
"https://config28.flexeye.com/v1/iot_Default/pms/pm1/pmUsers/user121/trustees
/testP1",
        "key" : "testP1"
      }, {
        "uri" :
"https://config28.flexeye.com/v1/iot_Default/pms/pm1/pmUsers/user121/trustees
/TestUser01",
        "key" : "TestUser01"
      } ],
      "trusteesCursor" : {
        "count" : "0",
        "nextUri" : null,
        "prevUri" : null
      },
      "firstName" : "user121",
      "lastName" : "user121",
      "address" : null,
      "gender" : "NaN",
      "uri" :
"https://config28.flexeye.com/v1/iot_Default/pms/pm1/pmUsers/user121"
    }
]
```

3.4.6 Remove Trustee

Description: Removes a trustee from the list of trustees of 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: Remove a trustee named "TestUser01" from the "user121" trustee's list
```
DELETE
https://config28.flexeye.com/v1/iot_Default/pms/dpm/pmUsers/user121/trustees/T
estUser01
Host: config28.flexeye.com
Authorization: Basic authentication
```

Example Response:
```
HTTP/1.1 200 OK
Content-Type: application/vnd.com.flexeye.iot+json
Content-Length: nnn
```

3.4.7 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 "user121" to "TestUser01".
```
PUT https://config28.flexeye.com/v1/iot_Default/allPeople/user121
Host: config28.flexeye.com
Authorization: Basic authentication
Accept: application/json
Content-Length: nnn
Content-Type: application/json

{
     "defaultTrustee" : "TestUser01"
}
```

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://config28.flexeye.com/v1/iot_Default/pms/pm1/pmUsers/user121/defaultT
rustee"
       },
       "uri" :
"https://config28.flexeye.com/v1/iot_Default/pms/pm1/pmUsers/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 the person "user121" where the location's country is the UK.

```
GET https://config28.flexeye.com/v1/iot_Default/allPeople/user121/locations
Query: addr_country=UK
Host: config28.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://config28.flexeye.com/v1/iot_Default/pms/pm1/pmUsers/user121/lo
cations/location1",
      "key" : "location1"
   }, {
      "uri" :
"https://config28.flexeye.com/v1/iot_Default/pms/pm1/pmUsers/user121/lo
cations/TestLocation1",
      "key" : "TestLocation1"
   } ],
   "locationsCursor" : {
   "count" : "0",
   "nextUri" : null,
   "prevUri" : null
   }
  }
]
```

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 "user121".
```
PUT https://config28.flexeye.com/v1/iot_Default/allPeople/user121
Host: config28.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" : "user121",
       "name" : null,
      "pmRef" : {
          "uri" :
     "https://config28.flexeye.com/v1/iot_Default/pms/pm1/pmUsers/user121/pm
     Ref"
       },
       "phoneNumber" : "1234567891",
       "emailId" : "eN@f.com",
       "firstName" : "updatedFN",
       "lastName" : "updatedLN",
       "locations" : [ {
         "uri" :
     "https://config28.flexeye.com/v1/iot_Default/pms/pm1/pmUsers/user121/lo
     cations/TestLocation1",
         "key" : "TestLocation1"
       }, {
         "uri" :
     "https://config28.flexeye.com/v1/iot_Default/pms/pm1/pmUsers/user121/lo
     cations/CCSR",
         "key" : "CCSR"
       } ],
       "locationsCursor" : {
         "count" : "0",
         "nextUri" : null,
         "prevUri" : null
       },
         "uri" :
     "https://config28.flexeye.com/v1/iot_Default/pms/pm1/pmUsers/user121"
     }
]
```

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://config28.flexeye.com/v1/iot_Default/pms/dpm/pmUsers/user12/locations/
CCSR
Host: config28.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://config28.flexeye.com/v1/iot_Default/allPeople/user121
Host: config28.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://config28.flexeye.com/v1/iot_Default/pms/pm1/pmUsers/user121/de
     vices/135",
          "key" : "135"
       } ],
       "devicesCursor" : {
         "count" : "0",
         "nextUri" : null,
         "prevUri" : null
       },
        "uri" :
     "https://config28.flexeye.com/v1/iot_Default/pms/pm1/pmUsers/user121"
     }
]
```

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://config28.flexeye.com/v1/iot_Default/allPeople/user121/devices/135
Host: config28.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://config28.flexeye.com/v1/iot_Default/allPeople/user121/events
Query: type=1.0
Host: config28.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://config28.flexeye.com/v1/iot_Default/pms/pm1/pmUsers/user121/ev
    ents/30538",
        "key" : "30538"
      } ],
      "eventsCursor" : {
        "count" : "0",
        "nextUri" : null,
        "prevUri" : null
      }
    }
]
```

3.4.14 Log Event

Description: Logs an 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 output event with timestamp of **"2013-09-25T14:30:34.000Z"** for the person "user121".

```
POST https://config28.flexeye.com/v1/iot_Default/allPeople/user121/events
Host: config28.flexeye.com
Authorization: Basic authentication
Accept: application/json
Content-Length: nnn
Content-Type: application/json

{
      "timeStamp" : "2013-09-25T14:30:34.000Z"
      "type" : "0.0",
      "payload" : "10"
}
```

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_38",
      "isUpdated" : "NaN",
      "uri" :
    "https://config28.flexeye.com/v1/iot_Default/pms/pm1/pmUsers/user121/ev
    ents/Event_user121_38"
    }
]
```

3.4.15 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://config28.flexeye.com/v1/iot_Default/allPeople/user121/timers
Query: state=3.0
Host: config28.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://config28.flexeye.com/v1/iot_Default/allPeople/user121/timers/T
      imer_user121_1",
          "key" : "Timer_user121_1"
        }, {
          "uri" :
      "https://config28.flexeye.com/v1/iot_Default/allPeople/user121/timers/T
      imer_user121_2",
          "key" : "Timer_user121_2"
        } ],
        "timersCursor" : {
          "count" : "0",
          "nextUri" : null,
          "prevUri" : null
        }
      }

]
```

3.4.16 Start Timer

Description: Creates and starts a Timer with the given duration for the Person. Option to specify an on-Expiry handler which will perform an action if the Timer is expires.
Synopsis: POST
Request Headers: Accept, Authorization, Content-Length, Content-Type
Request Parameters: duration
Optional Query parameters: notifyType, notifyCoverage
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 and starts a Timer with the given duration(3000 seconds) for the Person "user121".
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://config28.flexeye.com/v1/iot_Default/allPeople/user121/timers
Host: config28.flexeye.com
Authorization: Basic authentication
Accept: application/json
Content-Length: nnn
Content-Type: application/json

{
      "notifyType" : "0.0",
      "notifyCoverage" : "1.0",
      "duration" : "3000",
      "logMessage" : "an example timer expiry message"
}
```

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_user121_1",
      "notifyCoverage" : "1.0",
      "logMessage" : "an example timer expiry message",
```

```
        "notifyType" : 0.0,
        "startTime" : "2013-11-08T11:30:11.400Z",

        "uri" :
    "https://config28.flexeye.com/v1/iot_Default/allPeople/user121/timers/T
    imer_user121_1"
        }
]
```

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://config28.flexeye.com/v1/iot_Default/allPeople/user121/notifications
Query: type=0.0 (Email)
Host: config28.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://config28.flexeye.com/v1/iot_Default/allPeople/user121/notifica
    tions/N-1077",
```

```
      "key" : "N-1077"
    }, {
      "uri" :
"https://config28.flexeye.com/v1/iot_Default/allPeople/user121/notifica
tions/N-1078",
      "key" : "N-1078"
    } ],
    "notificationsCursor" : {
      "count" : "0",
      "nextUri" : null,
      "prevUri" : null
    }
  }
]
```

3.4.18 Send Notification

Description: Used to send a notification to a Person.
Synopsis: POST (Operation)
Request Headers: Accept, Authorization, Content-Length, Content-Type
Request Parameters: method, payload, senderVal, subjectVal
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 named "N112" to a Person.
```
POST
https://config28.flexeye.com/v1/iot_Default/allPeople/user1/sendNotification
Host: config28.flexeye.com
Authorization: Basic authentication
Accept: application/json
Content-Length: nnn
Content-Type: application/json

{
      "method" : "Email",
      "payload" : "xxxxx",
      "senderVal" : "xxx",
      "subjectVal" : "Notification subject is optional"
}
```

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

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

3.4.19 Accept Trustee Invitation

Description: Used to add trustee after accepting trustee invitation
Synopsis: POST (Operation)
Request Headers: Accept, Authorization, Content-Length, Content-Type
Request 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: Accept the trustee invitation of "user2" for the person "user1".

```
POST https://config28.flexeye.com/v1/iot_Default/allPeople/user1/accept
Host: config28.flexeye.com
Authorization: Basic authentication
Accept: application/json
Content-Length: nnn
Content-Type: application/json

{
      "personID" : "user2"
}
```

Example Response:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.com.flexeye.iot+json
Content-Length: nnn

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

3.4.20 Reject Trustee Invitation

Description: Used to reject trustee invitation(Note: Trustee will not be added in this case)
Synopsis: POST (Operation)
Request Headers: Accept, Authorization, Content-Length, Content-Type
Request 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: User1 rejects user2 as a trustee.
```
POST https://config28.flexeye.com/v1/iot_Default/allPeople/user1/reject
Host: config28.flexeye.com
Authorization: Basic authentication
Accept: application/json
Content-Length: nnn
Content-Type: application/json

{
     "personID" : "user2"
}
```

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

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

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://config28.flexeye.com/v1/iot_Default/allLocations/FlexeyeLoc
Host: config28.flexeye.com
Authorization: Basic authentication
Accept: N/A
Content-Length: N/A
Content-Ty/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://config28.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://config28.flexeye.com/data/iot_Defaul/allLocations/location1
Host: config28.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/11 200 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://config28.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://config28.flexeye.com/v1/iot_Default/lms/CCSR_LM/locations/location1
Host: config28.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://config28.flexeye.com/v1/iot_Default/allLocations/CCSR/events
Query: type=1.0 (Input)
Host: config28.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://config28.flexeye.com/v1/iot_Default/allLocations/CCSR/events/3
     5",
         "key" : "35"
       }, {
         "uri" :
     "https://config28.flexeye.com/v1/iot_Default/allLocations/CCSR/events/4
     1",
         "key" : "41"
       }, {
         "uri" :
     "https://config28.flexeye.com/v1/iot_Default/allLocations/CCSR/events/4
     2",
         "key" : "42"
       } ],
       "eventsCursor" : {
         "count" : "0",
         "nextUri" : null,
         "prevUri" : null
       }
     }
]
```


3.5.5 Log Event

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

```
POST https://config28.flexeye.com/v1/iot_Default/allLocations/CCSR/events
Host: config28.flexeye.com
Authorization: Basic authentication
Accept: application/json
Content-Length: nnn
Content-Type: application/json


{
     "timeStamp" : "2013-08-21T18:42:12.000Z"
     "type" : "1.0",
     "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" : "80",
       "uri" :
     "https://config28.flexeye.com/v1/iot_Default/allLocations/CCSR/events/8
     0"
     }
]
```

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://config28.flexeye.com/v1/iot_Default/dms/CCSR_DM
Host: config28.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://config28.flexeye.com/v1/iot_Default/dms/CCSR_DM/owner"
      },
      "uri" : "https://config28.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://config28.flexeye.com/v1/iot_Default/dms/DM1
Host: config28.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://config28.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://config28.flexeye.com/v1/iot_Default/dms/CCSR_DM/devices
Query: isActive=1.0 (true)
Host: config28.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" and device type to "MultiSensor.

```
POST https://config28.flexeye.com/v1/iot_Default/dms/CCSR_DM/devices
Host: config28.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"
}
```

Example Response:

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


{
      "uri" :
      "https://config28.flexeye.com/v1/iot_Default/dms/CCSR_DM/devices/device
      02",
      "id " : "device02",
      "name" : "device02",
      "deviceType " : "MultiSensor",
      "description " : "This is our First Device"
}
```

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://config28.flexeye.com/v1/iot_Default/dms/CCSR_DM/events
Query: type=0.0 (Output)
Host: config28.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" : {
    "events" : [ {
      "uri" :
"https://config28.flexeye.com/v1/iot_Default/dms/CCSR_DM/30492",
      "key" : "30492"
    } ],
    "eventsCursor" : {
      "count" : "0",
      "nextUri" : null,
      "prevUri" : null
    }
  }
]
```

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://config28.flexeye.com/v1/iot_Default/allDevices/135
Host: config28.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://config28.flexeye.com/v1/iot_Default/allDevices/135/reading_light",
      "key" : "reading_light"
    }, {
```

```
      "uri" :
"https://config28.flexeye.com/v1/iot_Default/allDevices/135/reading_mic",
      "key" : "reading_mic"
    }, {
      "uri" :
"https://config28.flexeye.com/v1/iot_Default/allDevices/135/reading_PIR",
      "key" : "reading_PIR"
    }, {
      "uri" :
"https://config28.flexeye.com/v1/iot_Default/allDevices/135/reading_temp",
      "key" : "reading_temp"
    }, {
      "uri" :
"https://config28.flexeye.com/v1/iot_Default/allDevices/135/reading_watts",
      "key" : "reading_watts"
    } ],
    "subDevicesCursor" : {
      "count" : "0",
      "nextUri" : null,
      "prevUri" : null
    }
  },
  "uri" : "https://config28.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://config28.flexeye.com/v1/iot_Default/allDevices/135
 Host: config28.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://config28.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://config28.flexeye.com/v1/iot_Default/dms/Flexeye_DM/devices/F_D010
Host: config28.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://config28.flexeye.com/v1/iot_Default/allDevices/135/subDevices
Query: location=CCSR
Host: config28.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://config28.flexeye.com/v1/iot_Default/allDevices/135/reading_lig
     ht",
           "key" : "reading_light"
         }, {
           "uri" :
     "https://config28.flexeye.com/v1/iot_Default/allDevices/135/reading_mic
     ",
           "key" : "reading_mic"
         }, {
           "uri" :
     "https://config28.flexeye.com/v1/iot_Default/allDevices/135/reading_PIR
     ",
```

```
              "key" : "reading_PIR"
          }, {
              "uri" :
      "https://config28.flexeye.com/v1/iot_Default/allDevices/135/reading_tem
      p",
              "key" : "reading_temp"
          }, {
              "uri" :
      "https://config28.flexeye.com/v1/iot_Default/allDevices/135/reading_wat
      ts",
              "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://config28.flexeye.com/v1/iot_Default/allDevices/135/events
Query: type=1.0 (Input)
Host: config28.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" : {
        "events" : [ {
          "uri" :
     "https://config28.flexeye.com/v1/iot_Default/allDevices/135/30493",
          "key" : "30493"
        } ],
        "eventsCursor" : {
          "count" : "0",
          "nextUri" : null,
          "prevUri" : null
        }
      }
]
```

3.7.6 Device IO Event (Out)

Description: Send an output event from an existing Device for a registered Device Manager on the Hub.
Synopsis: POST
Request Headers: Accept, Authorization, Content-Length, Content-Type
Request Parameters: Type, 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://config28.flexeye.com/v1/iot_Default/dms/CCSR_DM/devices/135/outputs
Host: config28.flexeye.com
Authorization: Basic authentication
Accept: application/json
Content-Length: nnn
Content-Type: application/json

{
     "timeStamp" : "2013-09-25T14:30:34.000Z",
     "type" : "0.0",
     "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/events/30549/d
evice"
  },
  "id" : "30549",
  "uri" :
"https://localhost:8181/v1/iot_Default/dms/CCSR_DM/devices/135/events/30549"
}
```

3.7.7 Device IO Event (In)

Description: Receive an input event to an existing Device from the Hub
Synopsis: POST
Request Headers: Accept, Authorization, Content-Length, Content-Type
Request Parameters: Type, 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://config28.flexeye.com/v1/iot_Default/allDevices/125/inEvents
Host: config28.flexeye.com
Authorization: Basic authentication
Accept: application/json
Content-Length: nnn
Content-Type: application/json

{
     "timeStamp" : "2013-09-25T14:30:34.000Z",
     "type" : "1.0",
     "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/events/30546/device"
  },
  "id" : "30546",
  "uri" : "https://localhost:8181/v1/iot_Default/allDevices/125/events/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://config28.flexeye.com/v1/iot_Default/applications/myGuardian
Host: config28.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://config28.flexeye.com/v1/iot_Default/applications/myGuardian/devices
Host: config28.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://config28.flexeye.com/v1/iot_Default/applications/myGuardian/device
s/134",
    "key" : "134"
  }, {
    "uri" :
"https://config28.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://config28.flexeye.com/v1/iot_Default/applications/myGuardian/devices/1
35
Host: config28.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://config28.flexeye.com/v1/iot_Default/applications/myGuardian/devices/
135"


}
```

3.8.4 Get Events

Description: Returns a list of output 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://config28.flexeye.com/v1/iot_Default/applications/myGuardian/devices/1
35/events
Query: type=0.0 (Output)
Host: config28.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://config28.flexeye.com/v1/iot_Default/applications/myGuardian/device
s/135/events/30493",
    "key" : "30493"
  }, {
    "uri" :
"https://config28.flexeye.com/v1/iot_Default/applications/myGuardian/device
s/135/events/30539",
    "key" : "30539"
  } ],
  "eventsCursor" : {
    "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://config28.flexeye.com/v1/iot_Default/applications/myGuardian/users
Query: firstName=John
Host: config28.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://config28.flexeye.com/v1/iot_Default/applications/myGuardian/us
      ers/
      JohnSmith",
        "key" : "JohnSmith"
        }, {
        "uri" :
      "https://config28.flexeye.com/v1/iot_Default/applications/myGuardian/us
      ers/
      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: PUT
Request Headers: Accept, Authorization, Content-Length, Content-Type
Request Parameters: users
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".

```
PUT https://config28.flexeye.com/v1/iot_Default/applications/myGuardian
Host: config28.flexeye.com
Authorization: Basic authentication
Accept: application/json
Content-Length: nnn
Content-Type: application/json


{
      "users" : "user121"
}
```

Example Response:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.com.flexeye.iot+json
Content-Length: nnn


{
  "name" : "myGuardian",
  "description" : "Security Application",
  "dateRegistered" : "2013-09-24T06:19:49.000Z",
  "isRegistered" : 1.0,
  "users" : [ {
  "uri" :
"https://config28.flexeye.com/v1/iot_Default/applications/myGuardian/
JoeBloggs",
    "key" : "JoeBloggs"
  }, {
    "uri" :
"https://config28.flexeye.com/v1/iot_Default/applications/myGuardian/
user121",
    "key" : "user121"
  } ],
```

```
  "usersCursor" : {
    "count" : "0",
    "nextUri" : null,
    "prevUri" : null
  },
  },
  "uri" :
"https://config28.flexeye.com/v1/iot_Default/applications/myGuardian"
}
```

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://config28.flexeye.com:8181/v1/iot_Default/applications/myGuardian/user
s/user1/authenticateUser
Host: config28.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


{
```

111

```
    "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://config28.flexeye.com/v1/iot_Default/applications/myGuardian/users/use
r1/loginUser
Host: config28.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".

```
POST
https://config28.flexeye.com/v1/iot_Default/applications/myGuardian/users/use
r1/logoutUser
Host: config28.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-105".
```
GET https://config28.flexeye.com/v1/iot_Default/timers/T-105
Host: config28.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://config28.flexeye.com/v1/iot_Default/timers/T-105"
}
```

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 "T105".
```
POST https://config28.flexeye.com/v1/iot_Default/timers/T105/pause
Host: config28.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 "T105"
```
POST https://config28.flexeye.com/v1/iot_Default/timers/T105/resume
Host: config28.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
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 "T105" before it expires.
```
POST https://config28.flexeye.com/v1/iot_Default/timers/T105/stop
Host: config28.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 "T105"
```
DELETE https://config28.flexeye.com/v1/iot_Default/timers/T105
Host: config28.flexeye.com
Authorization: Basic authentication
```

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

3.10 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://config28.flexeye.com/v1/iot_Default/allPeople/gs/notifications/N-
1070
Host: config28.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://config28.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://config28.flexeye.com/v1/iot_Default/allPeople/gs/notifications/N-1070
Host: config28.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 entities 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.

4.1.4 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

4.1.5 Version Control

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 1.2) with properties: server, scheme, host, port, path, apiCallBkFunc, represents and representedBy. Updated REST request path for Device IO Input (section 3.7.7) to "/inEvent" instead of "/inputs".