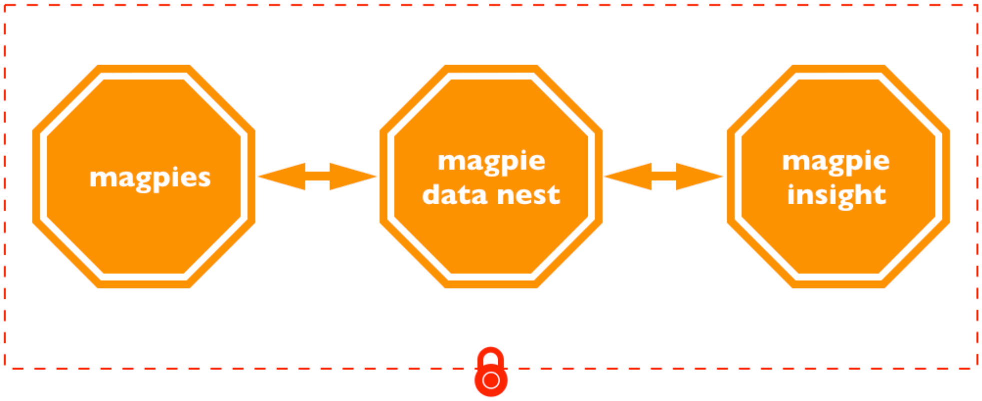
magpiecare FHIR API Developer Document

The Platform

There are 3 core components that make up the magpiecare platform as a service namely - magpies, magpie data nest and magpie insight.



The magpiecare platform as a service makes it easy to acquire the patient data over bluetooth low energy devices and sensors using magpie as a real time event collector. The magpie device then delivers the sensor generated data in real-time to magpie data nest in the cloud for realtime monitoring, historical plumbing, contextual awareness, visualization, analytics and notifications. The above pictorial representation render the inter component communication and event flow across the platform.

“magpie device” is a credit-card sized **secure**, **zero touch device** that plugs to USB port but monitored and managed remotely by magpie operations center. It communicates with biosensors over radio protocols and enables realtime acquisition of patient vitals. It also does realtime acquisition of sensor-instrumentation data including battery voltage for proactively addressing issues with the devices. It transports the data over secure communication to magpie secure cloud called “magpie data nest”. With a seamless plug-and-play deployment model, magpie is completely independent of end-user behaviors, technology roadblocks, connectivity issues, geographic distance and transforms the way remote environment specific ecosystem is managed and delivered.

The **“magpie data nest”** is a secure multi-tenant but isolated storage system supporting normalization and correlation of data received from magpies. It supports continuous delivery model and provides tools in the form of alert logics, event handlers, incident invocations to the supported operations from any geographical distance.

The **magpie “insight** portal provides the visualization, FHIR API and analytics for clients to connect and consume the contextual, processed, historical and time-series patient vitals, machine data and notifications. That processed data can then be used to support patient surveillance, post-acute care management, and in support of better patient outcomes.

The magpie insight FHIR API

Per the FHIR specification, Observations are a central element in healthcare, used to support diagnosis, monitor progress, determine baselines and patterns and even capture demographic characteristics. Most observations are simple name/value pair assertions with some metadata, but some observations group other observations together logically, or even are multi-component observations. Magpie specific expected uses for the Observation resource include Vital signs and measurements emitted by devices.

magpie observation resource collection

POST, GET **/Observation**

APIKEY Request Header (Required for apikey:key-value)

API GET Query Parameters *OR* API POST JSON Body (Content-Type:application/json)

API Request Header

Example Resource:

OBSERVATION HTTP: GET Request https://xxxxxx/fihr/Observation?puid=060347IZ.

OBSERVATION HTTP POST request (URL): https://xxxxxx/fihr/Observation
POST request (Header): Content-Type:application/JSON JSON Body: {"puid": "060347IZ"}

Crafting CURL Request

```
curl -v -X GET -H 'apikey:xxxxxxxxxxxxxxx' "https://xxxxxx/fhir/Observation?
puid=1234567891&vitals=oxi"
```

puid <val> Request Parameter(Required): The required puid (query | JSON body) supports the specification of the distinct PUID to look for and retrieve distinct patient objects. If not specified returns object not found error.

vitals <bp | oxi | hrm |act> Request Parameter (Required) : The "vitals" query parameter is a choice query associated with the "Observation" resource's API request for choosing the device data of interest.

earliest <time-val> Request Parameter (optional): This is a time- modifier query parameter to customize the time range for API search and retrieval. It tells the Magpie GRID to retrieve the associated timestamped event.

The "earliest" query param can work standalone or in tandem with "latest" time modifier. For pagination process it provides timehash functionality. The parameter takes time-values of style -2y or -2m or -60d or -2h or -8m or -8s to reflect upon years, months, days, hours, mins or seconds or its combination.

Another parameter timehash an integer-hash that is returned for each event can be used by the clients for "pagination" as well as to maintain the state of "last record read" to avoid duplicate queries. The timehash can be provided as value to "earliest" query to retrieve events from the last read.

latest <time-val> Request Parameter (optional): This is a time-modifier query parameter to customize the time range for API requests. It tells the Magpie GRID to retrieve the associated timestamped event until the latest specified.

The default is "now (most recent)". The "latest" query param can work standalone or in tandem with "earliest" time modifier.

recent <int> Request Parameter (optional): The recent query returns the recent n numbers of specified results to the request.This implies most recent n vitals for a historical search or the top n event for the real time request.

oldest <int> Request Parmeter (optional): The oldest query returns the bottom n numbers of specified results to the request. The results are returned in the reversed order starting at the oldest of the result set.

Resource Structure

Access token / api-key is required for accessing "Magpie Resource". Once you've registered your client it's easy to start requesting data from the API. The REST endpoint **Observation** resounce can only accessible via http**s** and are located at **api.magpie.xxxxxx** or partner bound API proxy. Please contact support for your distinct contracted proxy URL.

The Envelope

Every response is contained by an envelope. Each response has a predictable set of keys with which you can expect to interact:

```
{
"resourceType": <resource>,
"id": <id-type>,
"status": "generated",
"entity": "magpie insight edge FHIR DSTU2", "data": [
{},
{},
.
.
.
{}
]
}
```

"HTTP Response status"

The status of the response is communicated in the HTTP response itself by way of HTTP compliant "status codes". If all goes well HTTP response header will have a status code of "200 OK". HTTP Error Codes returned include "401 - Unauthorized" when there is API Key issue or "404 - Object not found" when HTTP requests are not correctly set or when queried for a non-existent resource.

"status"

This STATUS key depicts the "generated" value to outline the response as being generated by the target.

"id"

This key pertains to the 'id' of the queried response type and includes one of "weight-scale", "SatO2", "blood- pressure", "vitals-summary-count", "patient-unit-data", "patient-assigned-sensors", "magpies" or "realm".

"entity"

This key pertains to the type of care supported by the provider and preset at on-boarding time. It is typically a static value depicting "magpie insight FHIR DSTU2 platform".

"resourceType"

This key pertains to the resource returned in the HTTP response against whom the request was made.

"data"

The data key is the meat of the response. It is typically an array of dictionary. It is here you'll find the data you requested hence the name.

Swagger

The magpie insight FHIR framework supports Swagger, a set of open-source tools built around openAPI specification that can help with build, documentation and consumption of FHIR API. The swagger documentation allows to describe the structure of magpie FHIR API so compliant applications and utilities can generate client libraries, consume and read in the data.

"magpie swagger JSON"

```
{
  "swagger": "2.0",
  "info": {
    "description": "magpie API for Patient Vitals Request",
    "version": "2020-04-20T13:33:24Z",
    "title": "demoFHIR"
  },
  "host": “XXXXXX.XXXXXX.XXX",
  "basePath": "/prod",
  "schemes": [
    "https"
  ],
  "paths": {
    "/Observation": {
      "get": {
        "consumes": [
          "application/json"
        ],
        "produces": [
          "application/json"
        ],
        "parameters": [
          {
            "name": "puid",
            "in": "query",
            "required": true,
            "type": "string"
          },
          {
            "name": "latest",
            "in": "query",
            "required": false,
            "type": "string"
          },
          {
            "name": "vitals",
            "in": "query",
            "required": true,
            "type": "string"
          },
          {
            "name": "recent",
            "in": "query",
            "required": false,
            "type": "string"
          },
          {
            "name": "oldest",
            "in": "query",
            "required": false,
            "type": "string"
          },
          {
            "name": "earliest",
            "in": "query",
            "required": false,
            "type": "string"
          }
        ],
        "responses": {
          "200": {
            "description": "200 response",
            "schema": {
              "$ref": "#/definitions/Empty"
            }
          }
        },
        "security": [
          {
            "api_key": []
          }
        ]
      }
    }
  },
  "securityDefinitions": {
    "api_key": {
      "type": "apiKey",
      "name": "x-api-key",
      "in": "header"
    }
  },
  "definitions": {
    "Empty": {
      "type": "object",
      "title": "Empty Schema"
    }
  }
}
```