Purpose

• Widely adopted API spec format
  • Migrate from RAML 0.8 to OpenAPI 3.0
• More readable and useful API document
  • Swagger
About Swagger

Swagger is a powerful yet easy-to-use suite of API developer tools for teams and individuals, enabling development across the entire API lifecycle, from design and documentation, to test and deployment.

• Support OpenAPI Specification
  • https://github.com/OAI/OpenAPI-Specification
  • The OpenAPI Specification is a community-driven open specification within the OpenAPI Initiative, a Linux Foundation Collaborative Project.

• Open Source Tools
  • Swagger Editor (Apache 2.0): Design APIs in a powerful editor which visually renders your OpenAPI definition and provides real-time error feedback.
  • Swagger UI (Apache 2.0): Automatically generate documentation from your OpenAPI definition for visual interaction, and easier consumption.
  • Swagger Codegen (Apache 2.0): Build and enable consumption of your API by generating server stubs and client SDKs with minimal plumbing.

• Cloud platform - SwaggerHub ($30/Month)
  • SwaggerHub brings together all of the open source Swagger tooling into one easy-to-use platform in the cloud.
  • SwaggerHub can synchronize with github.
  • Provide the API Auto Mocking.
RAML 0.8

title: core-data
version: "1.0.0"
baseUri: "http://localhost:48080/api/v1"

Schemas:

- event: {'type":":object","$schema":"http://json-schema.org/draft-03/schema#","description":"Core device'
- reading: {'type":":object","$schema":"http://json-schema.org/draft-03/schema#","description":"Core devi'
- valueDescriptor: {'type":":object","$schema":"http://json-schema.org/draft-03/schema#","description":f'

/event:
displayName: Event Resource
description: example - http://localhost:48080/api/v1/event
post:
description: Add a new event (with its associated readings). Prefers the event device is a device name
displayName: add an event (and associated readings)
body:
  application/json:
    schema: event
    example: '{"origin":1471806386919,"device":"livingroomthermostat","readings":{"origin":1471806386919,"device":"livingroomthermostat","readings":[]} responses:

  "200":
    description: new event database generated id
  "400":
    description: creation request is invalid
  "404":
    description: if the a reading is associated to a non-existent value descriptor, or if device ve
  "500":
    description: for unknown or unanticipated issues.

put:
description: Update the event data (not including updating the readings). NotFoundException (HTTP 404)
displayName: update an event
body:
  application/json:
    schema: event
    example: '{"id":57ba04a1189b95b8afcdaf2d7","pushed":1471806399999}'
responses:

  "200":
    description: boolean on success of update request
OpenAPI 3.0

```yaml
openapi: 3.0.0

servers:
  - description: CoreData localhost
    url: http://localhost:8080/api/v1

info:
  description: EdgeX Foundry device
  version: "1.0.0"
  title: CoreData service
  termsOfService: 'http://swagger.io/terms/
  contact: <1 key>
  license: <2 keys>

tags:
  - name: reading
    description: device readind data
    externalDocs: <3 keys>

paths:
  /reading:
    get:
      tags:
        - reading
      summary: get reading
      operationId: getReading
      responses:
        '200':
          description: successful operation
          content:
            application/json:
              schema:
                type: array
        '400':
          description: Invalid status value

components:
  schemas:
    Reading:
      type: object
      properties:
```
OpenAPI 3.0 - Security

```yaml
openapi: 3.0.0
# Added by API Auto Mocking Plugin
servers:
  - description: SwaggerHub API Auto Mocking
    url: https://virtserver.swaggerhub.com/weichou/oauth2/1.0.0
  info:
    version: "1.0.0"
    title: Sample Access Code Flow OAuth2 Project
    description: >-
      This is an example of using OAuth2 Access Code Flow in a
      specification to
      describe security to your API.

security:
  - accessToken:
    - read
    - write
paths:
  /example:
    get:

securitySchemes:
  accessToken:
    type: oauth2
    flows:
      authorizationCode:
        authorizationUrl: 'https://dev-396343.oktapreview.com/oauth2/default/v1/authorize'
        tokenUrl: 'http://example.com/oauth/token'
        scopes:
          write: allows modifying resources
          read: allows reading resources
```

![Image of code snippet](image-url)
SwaggerHub - Online Document and API Auto Mucking for trying out the EdgeX feature
The End
core-data API documentation version 1.0.0

http://localhost:48080/api/v1

Event Resource

example - http://localhost:48080/api/v1/event

/event

- POST: Add a new event (with its associated readings). Prefers the event device is a device name but can also be a device id (database generated). DataValidationException (HTTP 409) if the a reading is associated to a non-existent value descriptor. ServiceException (HTTP 500) for unknown or unanticipated issues.

- PUT: Update the event data (not including updating the readings). NotFoundException (HTTP 404) if the event cannot be found by id. ServiceException (HTTP 500) for unknown or unanticipated issues.

- GET: Fetch all events with their associated readings. LimitExceededException (HTTP 413) if the number of events exceeds the current max limit. ServiceException (HTTP 500) for unknown or unanticipated issues.
EdgeX Foundry Core Data Service includes the device and sensor collected data database and APIs to expose the database to other services as well as north-bound integration. The database is secure. Direct access to the database is restricted to the Core Data service APIs. Core Data also provides the REST API to create and register a new device.

Terms of service
Contact the developer
Apache 2.0

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Links</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>successful operation</td>
<td>No links</td>
</tr>
</tbody>
</table>

**GET /reading**

**Parameters**

No parameters

**Responses**

Example Value

```
{}
```

**Try it out**
Test API with Swagger - approach 1

It need to add **Access-Control-Allow-Origin** header at EdgeX services

```go
w.Header().Set("Access-Control-Allow-Origin", "*")
w.Header().Set("Access-Control-Allow-Headers", "Content-Type, access-control-allow-origin, access-control-allow-headers")
```

Or add a proxy
Test API with Swagger - approach 2

EdgeX

CoreData
  Swagger-ui

Core MetaData
  Swagger-ui

Core Command
  Swagger-ui