Tizen/Artik IoT Lecture Chapter 6. OCF Resource Type & IoTivity Simulator

Sungkyunkwan University

RESTful API Modeling Language (RAML)

- **RAML makes it easy to manage the whole API lifecycle from design to sharing**
  - http://raml.org
- **Machine readable API design that is *human friendly***
- **Support several languages**
  - node.js, Java, .NET, Python
- **The RAML definition are used to describe the payloads of the CRUDN operations**
  - Used in simulation
  - Validating of responses

```
#%RAML 1.0
title: World Music API
baseUri: http://example.api.com/{version}
version: v1

uses:
  Songs: !include libraries/songs.raml

annotationTypes:
  monitoringInterval:
    parameters:
      value: integer

traits:
  secured: !include secured/accessToken.raml

/songs:
  is: secured
  get:
    (monitoringInterval): 30
    queryParameters:
      genre:
        description: filter the songs by genr
  post:
  /{songId}:
    get:
      responses:
        200:
          body:
            application/json:
              type: Songs.Song
            application/xml:
              schema: !include schemas/songs.
              example: !include examples/song
```

CRUDN Operation Response Codes

- **A resource can be created or updated depending on the resource type's definition and allowed CRUDN operations**
- **The operation may have different response code with different meaning**

Response Code	Meaning
200	Payload of the response will confirm the change
201	Payload is the URL of the Resource that was created by the server as a result of a CREATE operation
204	OK, everything went well, no payload provided
403	*CASE1:* On RETRIEVE, the server does not support the values provided *CASE2:* the server could not CREATE or UPDATE the Resource due to a problem with the provided payload

OIC defined Resource Types

- **There are many resource types defined by OIC**
  - 62 resource types are defined
- **RAML and JSON set**
  - Acceleration.raml
  - oic.r.sensor.acceleration.json

Table 6-1 Alphabetical list of Resource Types

Friendly Name (informative)	Resource Type (rt)	Section
Acceleration Sensor	oic.r.sensor.acceleration	6.52
Activity Count	oic.r.sensor.activity.count	6.23
Altimeter	oic.r.altimeter	6.57
Atmospheric Pressure	oic.r.sensor.atmosphericpressure	6.24
Air Flow	oic.r.airflow	6.1
Air Flow Control	oic.r.airflowcontrol	6.2
Audio Controls	oic.r.audio	6.25
Auto Focus	oic.r.autofocus	6.26
Automatic Document Feeder	oic.r.automaticdocumentfeeder	6.27
Auto White Balance	oic.r.colour.autowhitebalance	6.31
Basic Resource Schema	Not Applicable	Annex A.1

Example: Acceleration

- **URI**: /AccelerationResURI
- **Rt**: oic.r.sensor.acceleration

JSON definition: oic.r.sensor.acceleration.json

```
{
  "id": "http://openinterconnect.org/iotdatamodels/schemas/oic.r.sensor.acceleration.json#",
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description" : "Copyright (c) 2016 Open Connectivity Foundation, Inc. All rights reserved.",
  "title": "Acceleration Sensor",
  "definitions": {
    "oic.r.sensor.acceleration": {
      "properties": {
        "acceleration": {
            "type": "number",
            "description": "ReadOnly, sensed acceleration experienced in 'g'."
        }
      }
    }
  },
  "type": "object",
  "allOf": [
    {"$ref": "oic.core.json#/definitions/oic.core"},
    {"$ref": "oic.baseResource.json#/definitions/oic.r.baseresource"},
    {"$ref": "#/definitions/oic.r.sensor.acceleration"}
  ],
  "required": ["acceleration"]
}
```

RAML definition: Acceleration.raml

```
#%RAML 0.8
title: OICAcceleration
version: v1.1.0-20160519
documentation:
  - title: Copyright (c) 2016 Open Connectivity Foundation, Inc.
    content: |
        Redistribution and use in source and binary forms, with o
        1.  Redistributions of source code must retain the above
        2.  Redistributions in binary form must reproduce the abo

        THIS SOFTWARE IS PROVIDED BY THE Open Connectivity Founda
schemas:
  - acceleration: !include oic.r.sensor.acceleration.json
traits:
  - interface:
      queryParameters:
        if:
          enum: ["oic.if.s","oic.if.baseline"]

/AccelerationResURI:
  description: |
    This resource provides a measure of proper acceleration (g fo
    (which is dependent on the co-ordinate system and the observe
    The value is a float which describes the acceleration experie

  displayName: Acceleration Sensor
  is: [ interface ] # valid for all methods

  get:
    responses:
      200:
        body:
          application/json:
            schema: acceleration
            example: |
              {
                "rt":           ["oic.r.sensor.acceleration"],
                "id":           "unique_example_id",
                "acceleration":  0.5
              }
```

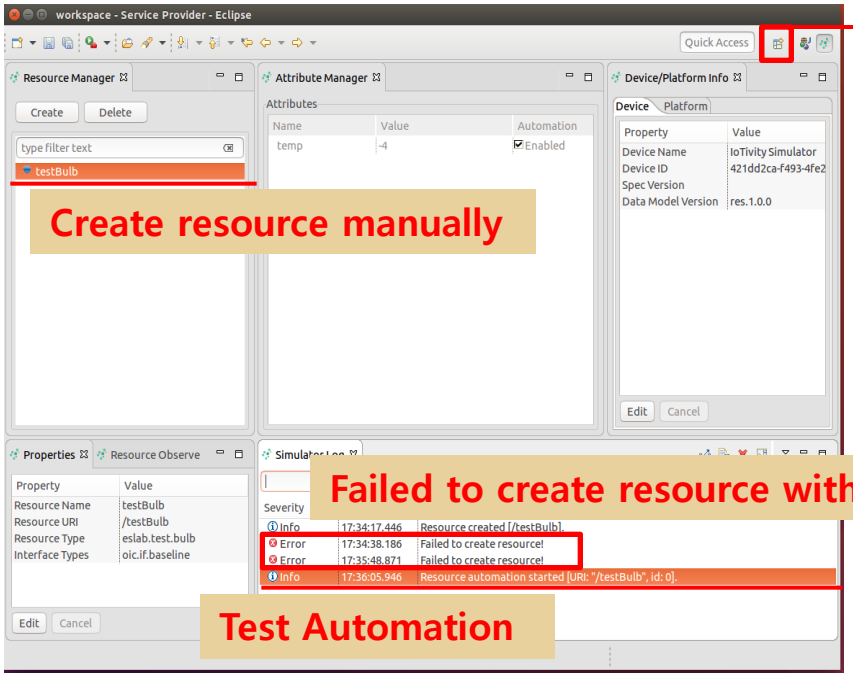
Read	oic.if.r	It allows only permission to read.
Read Write	oic.if.rw	It allows to both read and write.
Actuator	oic.if.a	It includes creating, updating and retrieving actuator values.
Sensor	oic.if.s	It allows only reading the sensor values.

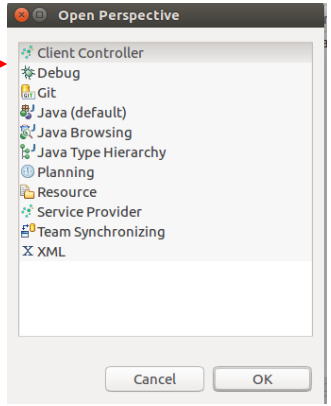
Name	Interface	
Baseline	oic.if.baseline	It includes all the information about the resource including meta-data and collection information about the resources. This is the default interface type.
Linked List	oic.if.ll	It includes only the collection information about the resources. This is the default interface type for /oic/res.
Batch	oic.if.b	It allows aggregating interaction with all resources. Each resource will be interacted separately, and response from them will be aggregated.

IoTivity Simulator: Service Provider

- **IoTivity Simulator provides two functions**
  - Service provider
  - Client controller

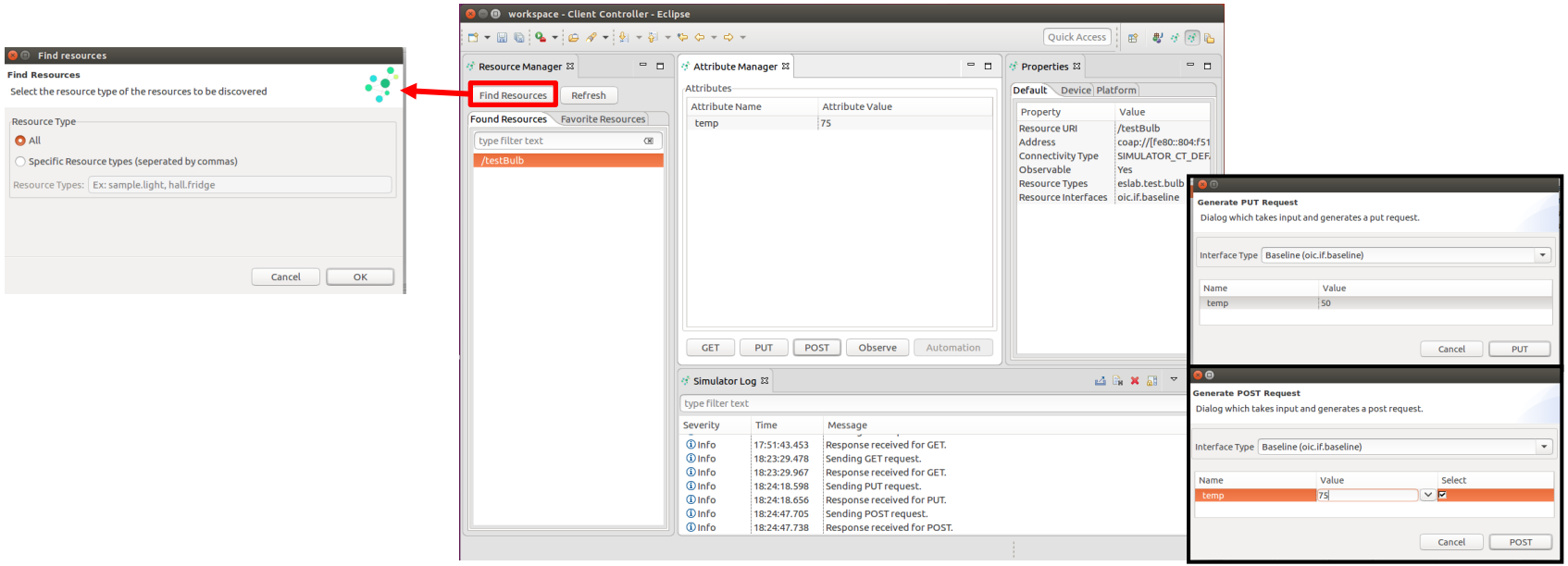
Two ways to create resource





IoTivity Simulator: Client Controller

- **Find resource at the client side**
- **You can use CRUDN operations**
  - GET, PUT, POST, OBSERVE, AUTOMATION



IoTivity Simulator: Observe

- **During OBSERVE, provider can notify to client**