OCF 2.3 – Modified Resources for ZWave Equivalency

1.1 Carbon Dioxide Sensor

1.1.1 Introduction

This resource describes whether carbon dioxide has been sensed or not.
The value is a boolean.
A value of 'true' means that carbon dioxide has been detected.
A value of 'false' means that carbon dioxide has not been detected.

1.1.2 Example URI

/CarbonDioxideResURI

1.1.3 Resource type

The resource type (rt) is defined as: ['oic.r.sensor.carbondioxide'].

1.1.4 Swagger 2.0 definition


```
{
  "swagger": "2.0",
  "info": {
    "title": "Carbon Dioxide Sensor",
    "version": "v1.1.0-20160519",
    "license": {
      "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
      "x-description": "Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:\n        1.
Redistributions of source code must retain the above copyright notice, this list of conditions and
the following disclaimer.\n       2.  Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.\n\n       THIS SOFTWARE IS PROVIDED BY THE Open
Connectivity Foundation, INC. \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n       IN NO EVENT SHALL THE Open Connectivity
Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n        HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
OF SUCH DAMAGE.\n"
    }
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/CarbonDioxideResURI" : {
      "get": {
        "description": "This resource describes whether carbon dioxide has been sensed or not.\nThe
value is a boolean.\nA value of 'true' means that carbon dioxide has been detected.\nA value of
'false' means that carbon dioxide has not been detected.\n",
        "parameters": [
          {"$ref": "#/parameters/interface"}
        ],
        "responses": {
            "200": {
              "description" : "",
              "x-example":
                {
```

```
                "rt":    ["oic.r.sensor.carbondioxide"],
                "id":    "unique_example_id",
                "value": true
              },
            "schema": { "$ref": "#/definitions/CO2" }
          }
        }
      }
    }
  },
  "parameters": {
    "interface" : {
      "in" : "query",
      "name" : "if",
      "type" : "string",
      "enum" : ["oic.if.s", "oic.if.baseline"]
    }
  },
  "definitions": {
    "CO2" : {
      "properties": {
        "rt" :  {
          "description": "Resource Type",
          "items": {
            "maxLength": 64,
            "type": "string",
            "enum": ["oic.r.sensor.carbondioxide"]
          },
          "minItems": 1,
          "readOnly": true,
          "type": "array"
        },
        "precision" :  {
          "description": "Accuracy granularity of the exposed measurement Property",
          "readOnly": true,
          "type": "number"
        },
        "value" :  {
          "description": "true = sensed, false = not sensed.",
          "readOnly": true,
          "type": "boolean"
        },
        "measurement":  {
          "type": "number",
          "description": "Measured value for this sensor, units are in ppm",
          "readOnly": true
        },
        "n" :  {
          "description": "Friendly name of the resource",
          "maxLength": 64,
          "readOnly": true,
          "type": "string"
        },
        "range" :  {
          "description": "The valid range for the measurement Property",
          "items": {
            "type": "number"
          },
          "maxItems": 2,
          "minItems": 2,
          "readOnly": true,
          "type": "array"
        },
        "step" :  {
          "type": "number",
```

```
          "description": "Step value across the defined range",
          "readOnly": true
        },
        "id" :  {
          "description": "Instance ID of this specific resource",
          "maxLength": 64,
          "readOnly": true,
          "type": "string"
        },
        "if" :  {
          "description": "The interface set supported by this resource",
          "items": {
            "enum": [
              "oic.if.baseline",
              "oic.if.s"
            ],
            "type": "string"
          },
          "minItems": 1,
          "readOnly": true,
          "type": "array"
        }
      },
      "type" : "object",
      "required": ["value"]
    }
  }
}
```

1.1.5 Property definition

Table 1The properties definitions of the resource with type 'rt' = CarbonDioxideResURI

Property name	Value type	Mandatory	Access mode	Description
if	array: see schema	No	Read Only	The interface set supported by this resource
n	string	No	Read Only	Friendly name of the resource
id	string	No	Read Only	Instance ID of this specific resource
measurement	number	No	Read Only	Measured value for this sensor, units are in ppm
value	boolean	Yes	Read Only	true = sensed, false = not sensed.
precision	number	No	Read Only	Accuracy granularity of the exposed

				measurement Property
step	number	No	Read Only	Step value across the defined range
range	array: see schema	No	Read Only	The valid range for the measurement Property
rt	array: see schema	No	Read Only	Resource Type

1.1.6 CRUDN behaviour

Table 2The CRUDN operations of the resource with type 'rt' = CarbonDioxideResURI

Resource	Create	Read	Update	Delete	Notify
/CarbonDioxideResURI		get			observe

1.2 Carbon Monoxide Sensor

1.2.1 Introduction

This resource describes whether carbon monoxide has been sensed or not.
The value is a boolean.
A value of 'true' means that carbon monoxide has been detected.
A value of 'false' means that carbon monoxide has not been detected.

1.2.2 Example URI

/CarbonMonoxideResURI

1.2.3 Resource type

The resource type (rt) is defined as: ['oic.r.sensor.carbonmonoxide'].

1.2.4 Swagger 2.0 definition


```
{
  "swagger": "2.0",
  "info": {
    "title": "Carbon Monoxide Sensor",
    "version": "v1.1.0-20160519",
    "license": {
      "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
      "x-description": "Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:\n        1.
Redistributions of source code must retain the above copyright notice, this list of conditions and
the following disclaimer.\n        2.  Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.\n\n        THIS SOFTWARE IS PROVIDED BY THE Open
Connectivity Foundation, INC. \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n        IN NO EVENT SHALL THE Open Connectivity
Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n        HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
```

```
OF SUCH DAMAGE.\n"
    }
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/CarbonMonoxideResURI" : {
      "get": {
        "description": "This resource describes whether carbon monoxide has been sensed or
not.\nThe value is a boolean.\nA value of 'true' means that carbon monoxide has been detected.\nA
value of 'false' means that carbon monoxide has not been detected.\n",
        "parameters": [
          {"$ref": "#/parameters/interface"}
        ],
        "responses": {
            "200": {
              "description" : "",
              "x-example":
                {
                  "rt":     ["oic.r.sensor.carbonmonoxide"],
                  "id":     "unique_example_id",
                  "value":  true
                },
              "schema": { "$ref": "#/definitions/CO" }
            }
        }
      }
    }
  },
  "parameters": {
    "interface" : {
      "in" : "query",
      "name" : "if",
      "type" : "string",
      "enum" : ["oic.if.s", "oic.if.baseline"]
    }
  },
  "definitions": {
    "CO" : {
      "properties": {
        "rt" :  {
          "description": "Resource Type",
          "items": {
            "maxLength": 64,
            "type": "string",
            "enum": ["oic.r.sensor.carbonmonoxide"]
          },
          "minItems": 1,
          "readOnly": true,
          "type": "array"
        },
        "precision" :  {
          "description": "Accuracy granularity of the exposed measurement Property",
          "readOnly": true,
          "type": "number"
        },
        "value" :  {
          "description": "true = sensed, false = not sensed.",
          "readOnly": true,
          "type": "boolean"
        },
        "measurement":  {
          "type": "number",
          "description": "Measured value for this sensor, units are in ppm",
```

```
        "readOnly": true
      },
      "n" :  {
        "description": "Friendly name of the resource",
        "maxLength": 64,
        "readOnly": true,
        "type": "string"
      },
      "range" :  {
        "description": "The valid range for the measurement Property",
        "items": {
          "type": "number"
        },
        "maxItems": 2,
        "minItems": 2,
        "readOnly": true,
        "type": "array"
      },
      "step" : {
        "type": "number",
        "description": "Step value across the defined range",
        "readOnly": true
      },
      "id" :  {
        "description": "Instance ID of this specific resource",
        "maxLength": 64,
        "readOnly": true,
        "type": "string"
      },
      "if" :  {
        "description": "The interface set supported by this resource",
        "items": {
          "enum": [
            "oic.if.baseline",
            "oic.if.s"
          ],
          "type": "string"
        },
        "minItems": 1,
        "readOnly": true,
        "type": "array"
      }
    },
    "type" : "object",
    "required": ["value"]
  }
 }
}
```


1.2.5 Property definition

Table 3The properties definitions of the resource with type 'rt' = CarbonMonoxideResURI

Property name	Value type	Mandatory	Access mode	Description
n	string	No	Read Only	Friendly name of the resource
id	string	No	Read Only	Instance ID of this specific resource
measurement	number	No	Read Only	

				Measured value for this sensor, units are in ppm
range	array: see schema	No	Read Only	The valid range for the measurement Property
if	array: see schema	No	Read Only	The interface set supported by this resource
value	boolean	Yes	Read Only	true = sensed, false = not sensed.
rt	array: see schema	No	Read Only	Resource Type
precision	number	No	Read Only	Accuracy granularity of the exposed measurement Property
step	number	No	Read Only	Step value across the defined range

1.2.6 CRUDN behaviour

Table 4The CRUDN operations of the resource with type 'rt' = CarbonMonoxideResURI

Resource	Create	Read	Update	Delete	Notify
/CarbonMonoxideResURI		get			observe

1.3 Generic Sensor

1.3.1 Introduction

This resource describes whether some value or property or entity has been sensed or not.
The value is a boolean.
A value of 'true' means that the target has been sensed.
A value of 'false' means that the target has not been sensed.

1.3.2 Example URI

/GenericSensorResURI

1.3.3 Resource type

The resource type (rt) is defined as: ['oic.r.sensor'].

1.3.4 Swagger 2.0 definition

```
{
  "swagger": "2.0",
```

```
  "info": {
    "title": "Generic Sensor",
    "version": "v1.1.0-20160519",
    "license": {
      "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
      "x-description": "Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:\n        1.
Redistributions of source code must retain the above copyright notice, this list of conditions and
the following disclaimer.\n        2.  Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.\n\n        THIS SOFTWARE IS PROVIDED BY THE Open
Connectivity Foundation, INC. \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n        IN NO EVENT SHALL THE Open Connectivity
Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n        HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
OF SUCH DAMAGE.\n"
    }
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/GenericSensorResURI" : {
      "get": {
        "description": "This resource describes whether some value or property or entity has been
sensed or not.\nThe value is a boolean.\nA value of 'true' means that the target has been
sensed.\nA value of 'false' means that the target has not been sensed.\n",
        "parameters": [
          {"$ref": "#/parameters/interface"}
        ],
        "responses": {
            "200": {
              "description" : "",
              "x-example":
                {
                  "rt":    ["oic.r.sensor"],
                  "id":    "unique_example_id",
                  "value": true
                }
                ,
              "schema": { "$ref": "#/definitions/Sensor" }
            }
        }
      }
    }
  },
  "parameters": {
    "interface" : {
      "in" : "query",
      "name" : "if",
      "type" : "string",
      "enum" : ["oic.if.s", "oic.if.baseline"]
    }
  },
  "definitions": {
    "Sensor" : {
      "properties": {
        "rt" :  {
          "description": "Resource Type",
          "items": {
            "maxLength": 64,
```

```
            "type": "string",
            "enum": ["oic.r.sensor"]
          },
          "minItems": 1,
          "readOnly": true,
          "type": "array"
        },
        "precision" :   {
          "description": "Accuracy granularity of the exposed mesurement",
          "readOnly": true,
          "type": "number"
        },
        "value" :   {
          "description": "true = sensed, false = not sensed.",
          "readOnly": true,
          "type": "boolean"
        },
        "measurement":  {
          "type": "number",
          "description": "Measured value for this sensor, units depend on the specific type of
sensor",
          "readOnly": true
        },
        "n" :   {
          "description": "Friendly name of the resource",
          "maxLength": 64,
          "readOnly": true,
          "type": "string"
        },
        "range" :   {
          "description": "The valid range for the measurement Property",
          "items": {
            "type": "number"
          },
          "maxItems": 2,
          "minItems": 2,
          "readOnly": true,
          "type": "array"
        },
        "step" :  {
          "type": "number",
          "description": "Step value across the defined range",
          "readOnly": true
        },
        "id" :   {
          "description": "Instance ID of this specific resource",
          "maxLength": 64,
          "readOnly": true,
          "type": "string"
        },
        "if" :  {
          "description": "The interface set supported by this resource",
          "items": {
            "enum": [
              "oic.if.baseline",
              "oic.if.s"
            ],
            "type": "string"
          },
          "minItems": 1,
          "readOnly": true,
          "type": "array"
        }
      },
      "type" : "object"
```

```
    }
  }
}
```

1.3.5 Property definition

Table 5The properties definitions of the resource with type 'rt' = GenericSensorResURI

Property name	Value type	Mandatory	Access mode	Description
rt	array: see schema		Read Only	Resource Type
id	string		Read Only	Instance ID of this specific resource
value	boolean		Read Only	true = sensed, false = not sensed.
precision	number		Read Only	Accuracy granularity of the exposed mesurement
if	array: see schema		Read Only	The interface set supported by this resource
step	number		Read Only	Step value across the defined range
n	string		Read Only	Friendly name of the resource
range	array: see schema		Read Only	The valid range for the measurement Property
measurement	number		Read Only	Measured value for this sensor, units depend on the specific type of sensor

1.3.6 CRUDN behaviour

Table 6The CRUDN operations of the resource with type 'rt' = GenericSensorResURI

Resource	Create	Read	Update	Delete	Notify
/GenericSensorResURI		get			observe

1.4 Smoke Sensor

1.4.1 Introduction

This resource describes whether smoke has been sensed or not.
The value is a boolean.
A value of 'true' means that smoke has been sensed.
A value of 'false' means that smoke not been sensed.

1.4.2 Example URI

/SmokeSensorResURI

1.4.3 Resource type

The resource type (rt) is defined as: ['oic.r.sensor.smoke'].

1.4.4 Swagger 2.0 definition


```
{
  "swagger": "2.0",
  "info": {
    "title": "Smoke Sensor",
    "version": "v1.1.0-20160519",
    "license": {
      "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
      "x-description": "Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:\n        1.
Redistributions of source code must retain the above copyright notice, this list of conditions and
the following disclaimer.\n        2.  Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.\n\n        THIS SOFTWARE IS PROVIDED BY THE Open
Connectivity Foundation, INC. \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n        IN NO EVENT SHALL THE Open Connectivity
Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n        HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
OF SUCH DAMAGE.\n"
    }
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/SmokeSensorResURI" : {
      "get": {
        "description": "This resource describes whether smoke has been sensed or not.\nThe value is
a boolean.\nA value of 'true' means that smoke has been sensed.\nA value of 'false' means that
smoke not been sensed.\n",
        "parameters": [
          {"$ref": "#/parameters/interface"}
        ],
        "responses": {
            "200": {
              "description" : "",
              "x-example":
                {
                  "rt":     ["oic.r.sensor.smoke"],
                  "id":     "unique_example_id",
                  "value":  true
                },
              "schema": { "$ref": "#/definitions/smoke" }
```

```
            }
          }
        }
      }
    },
    "parameters": {
      "interface" : {
        "in" : "query",
        "name" : "if",
        "type" : "string",
        "enum" : ["oic.if.s", "oic.if.baseline"]
      }
    },
    "definitions": {
      "smoke" : {
        "properties": {
          "rt" :  {
            "description": "Resource Type",
            "items": {
              "maxLength": 64,
              "type": "string",
              "enum": ["oic.r.sensor.smoke"]
            },
            "minItems": 1,
            "readOnly": true,
            "type": "array"
          },
          "precision" :  {
            "description": "Accuracy granularity of the exposed measurement Property",
            "readOnly": true,
            "type": "number"
          },
          "value" :  {
            "description": "true = sensed, false = not sensed.",
            "readOnly": true,
            "type": "boolean"
          },
          "measurement":  {
            "type": "number",
            "description": "Measured value for this sensor, this is a percentage",
            "readOnly": true
          },
          "n" :  {
            "description": "Friendly name of the resource",
            "maxLength": 64,
            "readOnly": true,
            "type": "string"
          },
          "range" :  {
            "description": "The valid range for the measurement Property",
            "items": {
              "type": "number"
            },
            "maxItems": 2,
            "minItems": 2,
            "readOnly": true,
            "type": "array"
          },
          "step" : {
            "type": "number",
            "description": "Step value across the defined range",
            "readOnly": true
          },
          "id" :  {
            "description": "Instance ID of this specific resource",
```

```
          "maxLength": 64,
          "readOnly": true,
          "type": "string"
        },
        "if" :  {
          "description": "The interface set supported by this resource",
          "items": {
            "enum": [
              "oic.if.baseline",
              "oic.if.s"
            ],
            "type": "string"
          },
          "minItems": 1,
          "readOnly": true,
          "type": "array"
        }
      },
      "type" : "object",
      "required": ["value"]
    }
  }
}
```

1.4.5 Property definition

Table 7The properties definitions of the resource with type 'rt' = SmokeSensorResURI

Property name	Value type	Mandatory	Access mode	Description
n	string	No	Read Only	Friendly name of the resource
if	array: see schema	No	Read Only	The interface set supported by this resource
precision	number	No	Read Only	Accuracy granularity of the exposed measurement Property
value	boolean	Yes	Read Only	true = sensed, false = not sensed.
measurement	number	No	Read Only	Measured value for this sensor, this is a percentage
rt	array: see schema	No	Read Only	Resource Type
id	string	No	Read Only	

				Instance ID of this specific resource
step	number	No	Read Only	Step value across the defined range
range	array: see schema	No	Read Only	The valid range for the measurement Property

1.4.6 CRUDN behaviour

Table 8The CRUDN operations of the resource with type 'rt' = SmokeSensorResURI

Resource	Create	Read	Update	Delete	Notify
/SmokeSensorResURI		get			observe

1.5 Water Sensor

1.5.1 Introduction

This resource describes whether water has been sensed or not.
The value is a boolean.
A value of 'true' means that water has been sensed.
A value of 'false' means that water not been sensed.

1.5.2 Example URI

/WaterResURI

1.5.3 Resource type

The resource type (rt) is defined as: ['oic.r.sensor.water'].

1.5.4 Swagger 2.0 definition


```
{
  "swagger": "2.0",
  "info": {
    "title": "Water Sensor",
    "version": "v1.1.0-20160519",
    "license": {
      "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
      "x-description": "Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:\n        1.
Redistributions of source code must retain the above copyright notice, this list of conditions and
the following disclaimer.\n        2.  Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.\n\n        THIS SOFTWARE IS PROVIDED BY THE Open
Connectivity Foundation, INC. \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n        IN NO EVENT SHALL THE Open Connectivity
Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n        HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
OF SUCH DAMAGE.\n"
```

```
    }
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/WaterResURI" : {
      "get": {
        "description": "This resource describes whether water has been sensed or not.\nThe value is
a boolean.\nA value of 'true' means that water has been sensed.\nA value of 'false' means that
water not been sensed.\n",
        "parameters": [
          {"$ref": "#/parameters/interface"}
        ],
        "responses": {
            "200": {
              "description" : "",
              "x-example":
                {
                  "rt":    ["oic.r.sensor.water"],
                  "id":    "unique_example_id",
                  "value": true
                },
              "schema": { "$ref": "#/definitions/Water" }
            }
        }
      }
    }
  },
  "parameters": {
    "interface" : {
      "in" : "query",
      "name" : "if",
      "type" : "string",
      "enum" : ["oic.if.s", "oic.if.baseline"]
    }
  },
  "definitions": {
    "Water" : {
      "properties": {
        "rt" : {
          "description": "Resource Type",
          "items": {
            "maxLength": 64,
            "type": "string",
            "enum": ["oic.r.sensor.water"]
          },
          "minItems": 1,
          "readOnly": true,
          "type": "array"
        },
        "precision": {
          "description": "Accuracy granularity of the exposed measurement Property",
          "readOnly": true,
          "type": "number"
        },
        "value" :  {
          "description": "true = sensed, false = not sensed.",
          "readOnly": true,
          "type": "boolean"
        },
        "measurement":  {
          "type": "number",
          "description": "Measured value for this sensor in units of litres/hr",
          "readOnly": true
```

```
        },
        "n" : {
          "description": "Friendly name of the resource",
          "maxLength": 64,
          "readOnly": true,
          "type": "string"
        },
        "range" :  {
          "description": "The valid range for the measurement Property",
          "items": {
            "type": "number"
          },
          "maxItems": 2,
          "minItems": 2,
          "readOnly": true,
          "type": "array"
        },
        "step" :  {
          "type": "number",
          "description": "Step value across the defined range",
          "readOnly": true
        },
        "id" :   {
          "description": "Instance ID of this specific resource",
          "maxLength": 64,
          "readOnly": true,
          "type": "string"
        },
        "if" :  {
          "description": "The interface set supported by this resource",
          "items": {
            "enum": [
              "oic.if.baseline",
              "oic.if.s"
            ],
            "type": "string"
          },
          "minItems": 1,
          "readOnly": true,
          "type": "array"
        }
      },
      "type" : "object",
      "required": ["value"]
    }
  }
}
```

1.5.5 Property definition

Table 9The properties definitions of the resource with type 'rt' = WaterResURI

Property name	Value type	Mandatory	Access mode	Description
id	string	No	Read Only	Instance ID of this specific resource
rt	array: see schema	No	Read Only	Resource Type
precision	number	No	Read Only	Accuracy granularity of the

				exposed measurement Property
if	array: see schema	No	Read Only	The interface set supported by this resource
n	string	No	Read Only	Friendly name of the resource
range	array: see schema	No	Read Only	The valid range for the measurement Property
step	number	No	Read Only	Step value across the defined range
value	boolean	Yes	Read Only	true = sensed, false = not sensed.
measurement	number	No	Read Only	Measured value for this sensor in units of litres/hr

1.5.6 CRUDN behaviour

Table 10The CRUDN operations of the resource with type 'rt' = WaterResURI

Resource	Create	Read	Update	Delete	Notify
/WaterResURI		get			observe