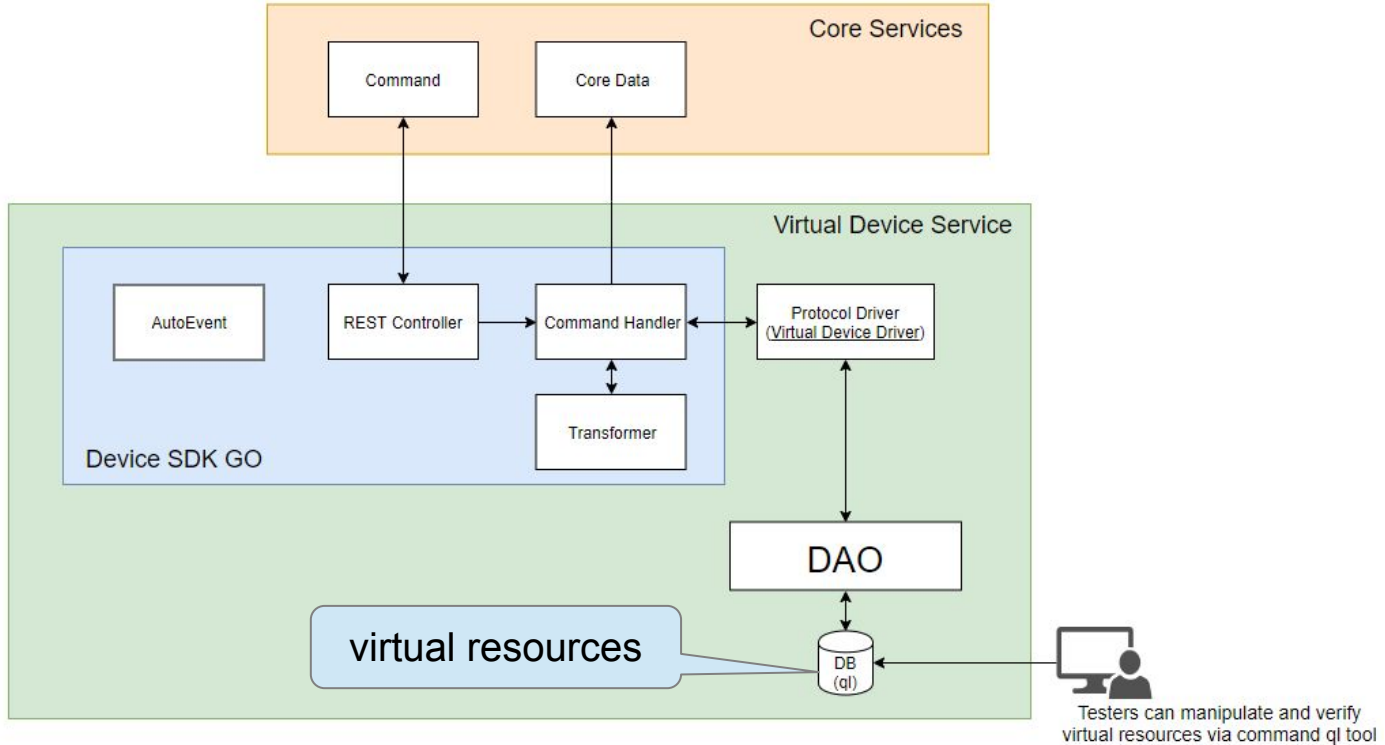


EdgeX Virtual Device Service GO Demonstration

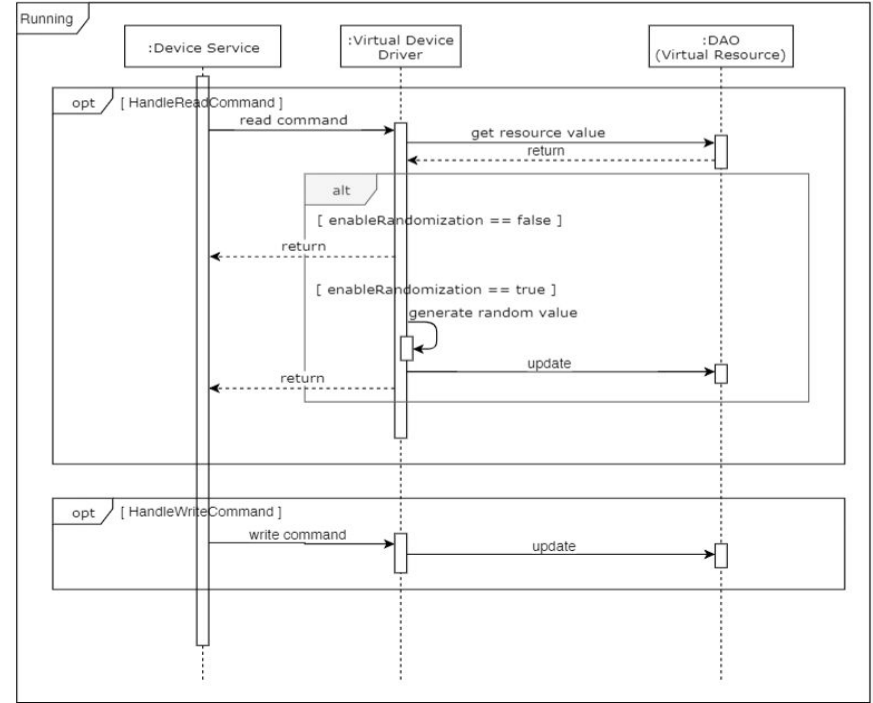
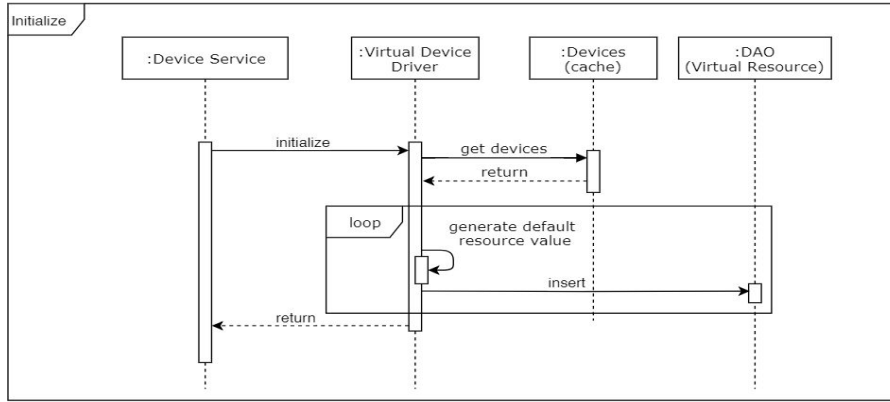
Table of Contents

- Overview
- Sequence Diagram
- Environment
- Introduction
 - Data Type & Device Profile
 - Find Executable Commands
 - Virtual Resource Table Schema
 - Access q1 DB
 - Common sql with q1 command
- Example (Take resource RandomValue_Int8 as an example)
 - Get resource's value
 - Set resource's Enable_Randomization to false
 - Set resource's value to 99
 - Set resource's Enable_Randomization and value simultaneously

Overview



Sequence Diagram



Environment

- GO: <https://golang.org/doc/install>
- Docker: <https://docs.docker.com/install/>
- Docker Compose: <https://docs.docker.com/compose/install/>
- EdgeX docker-compose file:
 - Base on [docker-compose-nexus.yml](#), switch image version to 1.0.0
- Command ql: <https://godoc.org/modernc.org/ql/ql>

Virtual Device Service GO Introduction: Data Type & Device Profile

Resource data types:

- Bool
- Int8, Int16, Int32, Int64
- Uint8, Uint16, Uint32, Uint64
- Float32, Float64

Device Profiles:

- Random-Boolean-Generator
 - [device.virtual.bool.yaml](#)
- Random-Integer-Generator
 - [device.virtual.int.yaml](#)
- Random-UnsignedInteger-Generator
 - [device.virtual.uint.yaml](#)
- Random-Float-Generator
 - [device.virtual.float.yaml](#)

Virtual Device Service GO Introduction:

Adding device-virtual service to docker-compose file

```
#####  
# Device Services  
#####  
  
device-virtual:  
  image: edgexfoundry/docker-device-virtual-go:alpha  
  ports:  
    - "49988:49988"  
  container_name: device-virtual  
  hostname: device-virtual  
  networks:  
    - edgex-network  
  volumes:  
    - db-data:/data/db  
    - log-data:/edgex/logs  
    - consul-config:/consul/config  
    - consul-data:/consul/data  
    - db-devicevirtual:/db  
  depends_on:  
    - data  
    - command
```

device virtual image

Mount ql db directory

deploy and start services...

```
$ docker-compose pull  
$ docker-compose up -d
```

Device Virtual Service GO Introduction:

Find executable commands

- Random-Boolean-Generator
 - [http://\[host\]:48082/api/v1/device/name/Random-Boolean-Generator01](http://[host]:48082/api/v1/device/name/Random-Boolean-Generator01)
- Random-Integer-Generator
 - [http://\[host\]:48082/api/v1/device/name/Random-Integer-Generator01](http://[host]:48082/api/v1/device/name/Random-Integer-Generator01)
- Random-UnsignedInteger-Generator
 - [http://\[host\]:48082/api/v1/device/name/Random-UnsignedInteger-Generator01](http://[host]:48082/api/v1/device/name/Random-UnsignedInteger-Generator01)
- Random-Float-Generator
 - [http://\[host\]:48082/api/v1/device/name/Random-Float-Generator01](http://[host]:48082/api/v1/device/name/Random-Float-Generator01)

Virtual Device Service GO Introduction:

Find executable commands-2

```
device.virtual.int.yaml
```

```
commands:
-
  name: "RandomValue_Int8"
  get:
    path: "/api/v1/device/{deviceId}/RandomValue_Int8"
    responses:
    -
      code: "200"
      description: ""
      expectedValues: ["RandomValue_Int8"]
    -
      code: "503"
      description: "service unavailable"
      expectedValues: []
  put:
    path: "/api/v1/device/{deviceId}/RandomValue_Int8"
    parameterNames: ["RandomValue_Int8", "Enable_Randomization"]
    responses:
    -
      code: "200"
      description: ""
    -
      code: "503"
      description: "service unavailable"
```

Device Profile-commands

```
configuration.toml
```

```
# Pre-define Schedule Configuration
[[Schedules]]
Name = "5sec-schedule"
Frequency = "PT5S"

[[ScheduleEvents]]
Name = "readValue_int8"
Schedule = "5sec-schedule"
[ScheduleEvents.Addressable]
HTTPMethod = "GET"
Path = "/api/v1/device/name/Random-Integer-Generator01/RandomValue_Int8"
```

Service Configuration-Schedule/ScheduleEvent

Will be replaced by AutoEvent in 1.0.0

Reminder:

1. The "Enable_Randomization" parameter is used to control whether the device returns a random value of the resource.
2. If you want to set a specified value to the resource, need to set Enable_Randomization to false simultaneously. Otherwise, the value will be updated by a random value when executing get value command next time. See [example](#)

Virtual Device Service GO Introduction: Virtual Resource Table Schema

Table name: **VIRTUAL_RESOURCE**

COLUMN	TYPE
DEVICE_NAME	string
COMMAND_NAME	string
DEVICE_RESOURCE_NAME	string
ENABLE_RANDOMIZATION	bool
DATA_TYPE	string
VALUE	string

On virtual device service initializing:

1. Clean data from DB
2. Insert default values into DB according to the managed devices from cache

```
"DEVICE_NAME", "COMMAND_NAME", "DEVICE_RESOURCE_NAME", "ENABLE_RANDOMIZATION", "DATA_TYPE", "VALUE"  
"Random-Float-Generator01", "RandomValue_Float64", "RandomValue_Float64", true, "Float64", "3.14159265359"  
"Random-Float-Generator01", "RandomValue_Float32", "RandomValue_Float32", true, "Float32", "3.14159"  
"Random-UnsignedInteger-Generator01", "RandomValue_Uint64", "RandomValue_Uint64", true, "Uint64", "0"  
"Random-UnsignedInteger-Generator01", "RandomValue_Uint32", "RandomValue_Uint32", true, "Uint32", "0"  
"Random-UnsignedInteger-Generator01", "RandomValue_Uint16", "RandomValue_Uint16", true, "Uint16", "0"  
"Random-UnsignedInteger-Generator01", "RandomValue_Uint8", "RandomValue_Uint8", true, "Uint8", "0"  
"Random-Integer-Generator01", "RandomValue_Int64", "RandomValue_Int64", true, "Int64", "0"  
"Random-Integer-Generator01", "RandomValue_Int32", "RandomValue_Int32", true, "Int32", "0"  
"Random-Integer-Generator01", "RandomValue_Int16", "RandomValue_Int16", true, "Int16", "0"  
"Random-Integer-Generator01", "RandomValue_Int8", "RandomValue_Int8", true, "Int8", "0"  
"Random-Boolean-Generator01", "RandomValue_Bool", "RandomValue_Bool", true, "Bool", "true"
```

Virtual Device Service GO Introduction:

Access ql DB

Find mountpoint:

```
$ docker volume inspect edgex_temp_db-devicevirtual
```

```
[
  {
    "CreatedAt": "2019-02-25T17:08:51+08:00",
    "Driver": "local",
    "Labels": {
      "com.docker.compose.project": "edgex_temp",
      "com.docker.compose.version": "1.22.0",
      "com.docker.compose.volume": "db-devicevirtual"
    },
    "Mountpoint": "/var/lib/docker/volumes/edgex_temp_db-devicevirtual/_data"
  },
  {
    "Name": "edgex_temp_db-devicevirtual",
    "Options": null,
    "Scope": "local"
  }
]
```

```
$ cd /var/lib/docker/volumes/edgex_temp_db-devicevirtual/_data
```

(see ql commands on the next page)

Virtual Device Service GO Introduction:

Common sql with ql command

Query all data:

```
$ ql -db deviceVirtual.db -fld "select * from VIRTUAL_RESOURCE"
```

Update enable_randomization:

```
$ ql -db deviceVirtual.db "update VIRTUAL_RESOURCE set ENABLE_RANDOMIZATION=false  
where DEVICE_NAME=\"Random-Integer-Generator01\" and DEVICE_RESOURCE_NAME  
=\"RandomValue_Int8\""
```

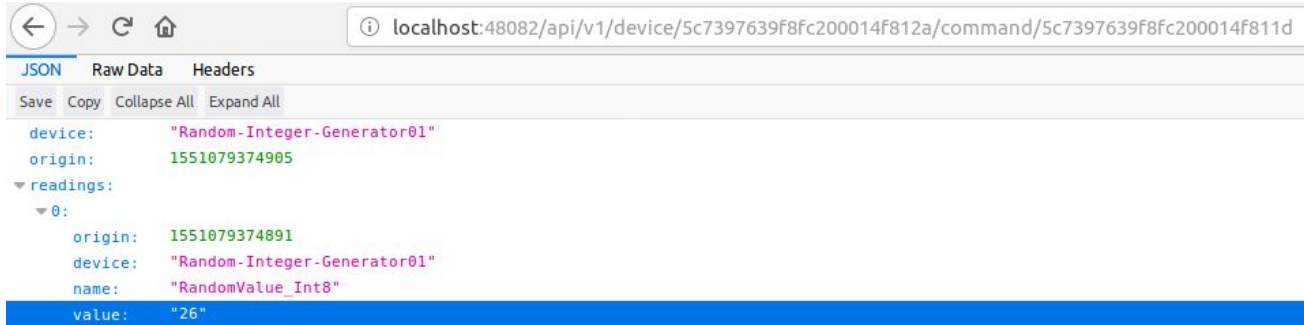
Update value:

```
$ ql -db deviceVirtual.db "update VIRTUAL_RESOURCE set VALUE=\"26\" where DEVICE_NAME  
=\"Random-Integer-Generator01\" and DEVICE_RESOURCE_NAME=\"RandomValue_Int8\""
```

Virtual Device Service GO Example-1: (Take resource RandomValue_Int8 as an example)

Get resource's value

<http://localhost:48082/api/v1/device/5c7397639f8fc200014f812a/command/5c7397639f8fc200014f811d>



```
localhost:48082/api/v1/device/5c7397639f8fc200014f812a/command/5c7397639f8fc200014f811d
JSON Raw Data Headers
Save Copy Collapse All Expand All
device: "Random-Integer-Generator01"
origin: 1551079374905
readings:
  0:
    origin: 1551079374891
    device: "Random-Integer-Generator01"
    name: "RandomValue_Int8"
    value: "26"
```

Virtual Device Service GO Example-2:

Set resource's Enable_Randomization to false

Send PUT request:

PUT `http://192.168.56.101:48082/api/v1/device/5c7397639f8fc200014f812a/command/5c7397639f8fc200014f811d` Send Save

Params Authorization Headers (1) **Body** Pre-request Script Tests Cookies Code Comments

none form-data x-www-form-urlencoded raw binary JSON (application/json) Beautify

```
1 [{"Enable_Randomization":"false"}]
```

Body Cookies Headers (3) Test Results Status: 200 OK Time: 41 ms Size: 107 B

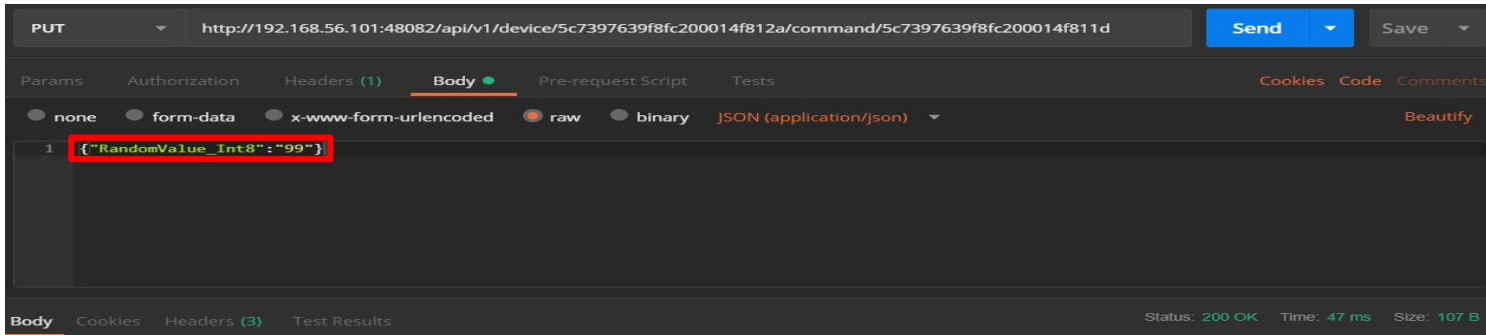
See result using command ql:

```
root@felix-VirtualBox: /var/lib/docker/volumes/edgex_temp_db-devicevirtual/_data# ql -db deviceVirtual.db -fld "select * from VIRTUAL_RESOURCE"
"DEVICE_NAME", "COMMAND_NAME", "DEVICE_RESOURCE_NAME", "ENABLE_RANDOMIZATION", "DATA_TYPE", "VALUE"
"Random-Float-Generator01", "RandomValue_Float64", "RandomValue_Float64", true, "Float64", "3.14159265359"
"Random-Float-Generator01", "RandomValue_Float32", "RandomValue_Float32", true, "Float32", "3.14159"
"Random-UnsignedInteger-Generator01", "RandomValue_Uint64", "RandomValue_Uint64", true, "Uint64", "0"
"Random-UnsignedInteger-Generator01", "RandomValue_Uint32", "RandomValue_Uint32", true, "Uint32", "0"
"Random-UnsignedInteger-Generator01", "RandomValue_Uint16", "RandomValue_Uint16", true, "Uint16", "0"
"Random-UnsignedInteger-Generator01", "RandomValue_Uint8", "RandomValue_Uint8", true, "Uint8", "0"
"Random-Integer-Generator01", "RandomValue_Int64", "RandomValue_Int64", true, "Int64", "0"
"Random-Integer-Generator01", "RandomValue_Int32", "RandomValue_Int32", true, "Int32", "0"
"Random-Integer-Generator01", "RandomValue_Int16", "RandomValue_Int16", true, "Int16", "0"
"Random-Integer-Generator01", "RandomValue_Int8", "RandomValue_Int8", false, "Int8", "26"
"Random-Boolean-Generator01", "RandomValue_Bool", "RandomValue_Bool", true, "Bool", "true"
```

Virtual Device Service GO Example-3:

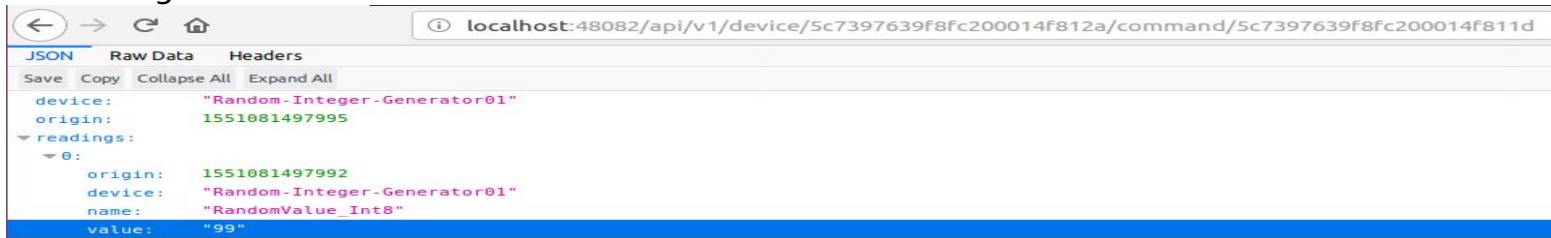
Set resource's value to 99

Send PUT request:



A screenshot of a REST client interface. The URL bar shows a PUT request to `http://192.168.56.101:48082/api/v1/device/5c7397639f8fc200014f812a/command/5c7397639f8fc200014f811d`. The 'Body' tab is selected, and the request body is `{"RandomValue_Int8": "99"}`, which is highlighted with a red box. The status bar at the bottom indicates a successful response: `Status: 200 OK`, `Time: 47 ms`, and `Size: 107 B`.

Get Int8 value again:



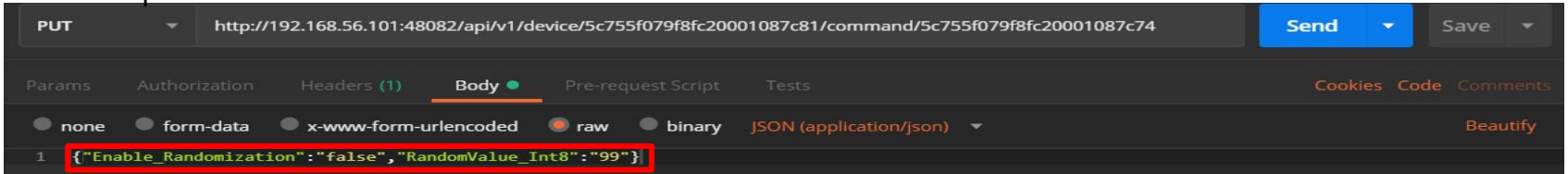
A screenshot of a browser's developer console showing the response of a GET request. The URL is `localhost:48082/api/v1/device/5c7397639f8fc200014f812a/command/5c7397639f8fc200014f811d`. The response is displayed in JSON format:

```
device: "Random-Integer-Generator01"
origin: 1551081497995
readings:
  0:
    origin: 1551081497992
    device: "Random-Integer-Generator01"
    name: "RandomValue_Int8"
    value: "99"
```

Virtual Device Service GO Example-4:

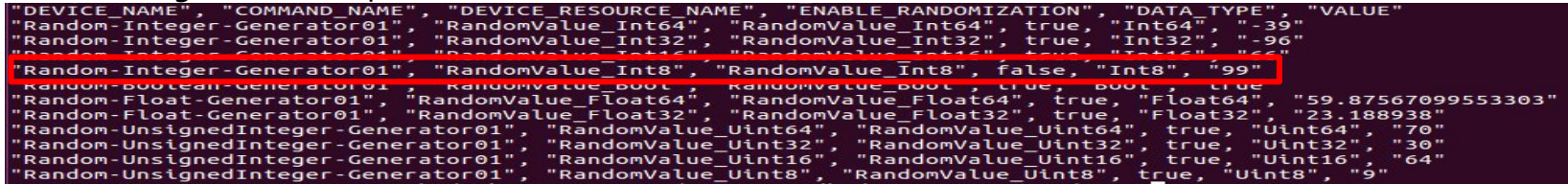
Set resource's Enable_Randomization and Value simultaneously

Send PUT request:



A screenshot of a REST client interface. The URL is `http://192.168.56.101:48082/api/v1/device/5c755f079f8fc20001087c81/command/5c755f079f8fc20001087c74`. The request body is `{"Enable_Randomization": "false", "RandomValue_Int8": "99"}`, which is highlighted with a red box.

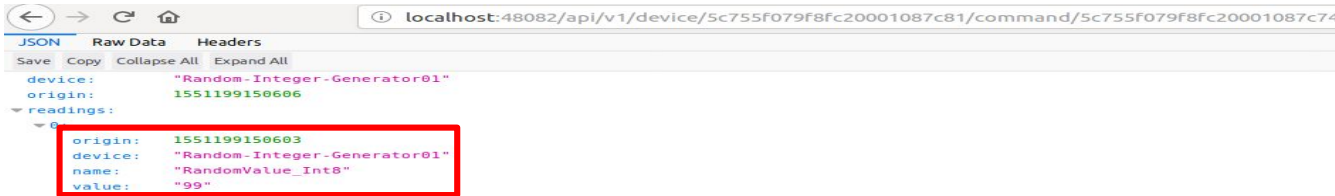
See result using command ql:



A screenshot of a terminal window showing the output of the 'ql' command. The output is a table with columns: DEVICE_NAME, COMMAND_NAME, DEVICE_RESOURCE_NAME, ENABLE_RANDOMIZATION, DATA_TYPE, and VALUE. The row for 'Random-Integer-Generator01' with 'RandomValue_Int8' is highlighted with a red box, showing 'ENABLE_RANDOMIZATION' as 'false' and 'VALUE' as '99'.

DEVICE_NAME	COMMAND_NAME	DEVICE_RESOURCE_NAME	ENABLE_RANDOMIZATION	DATA_TYPE	VALUE
Random-Integer-Generator01	RandomValue_Int64	RandomValue_Int64	true	Int64	-39
Random-Integer-Generator01	RandomValue_Int32	RandomValue_Int32	true	Int32	-96
Random-Integer-Generator01	RandomValue_Int16	RandomValue_Int16	true	Int16	66
Random-Integer-Generator01	RandomValue_Int8	RandomValue_Int8	false	Int8	99
Random-Boolean-Generator01	RandomValue_Boolean	RandomValue_Boolean	true	Boolean	true
Random-Float-Generator01	RandomValue_Float64	RandomValue_Float64	true	Float64	59.87567099553303
Random-Float-Generator01	RandomValue_Float32	RandomValue_Float32	true	Float32	23.188938
Random-UnsignedInteger-Generator01	RandomValue_Uint64	RandomValue_Uint64	true	Uint64	70
Random-UnsignedInteger-Generator01	RandomValue_Uint32	RandomValue_Uint32	true	Uint32	30
Random-UnsignedInteger-Generator01	RandomValue_Uint16	RandomValue_Uint16	true	Uint16	64
Random-UnsignedInteger-Generator01	RandomValue_Uint8	RandomValue_Uint8	true	Uint8	9

Get Int8 value again:



A screenshot of a REST client showing the response for the Int8 value. The response is a JSON object with fields: device, origin, and readings. The readings array contains an object with fields: origin, device, name, and value. The value field is '99', which is highlighted with a red box.

```
{
  "device": "Random-Integer-Generator01",
  "origin": "1551199150606",
  "readings": [
    {
      "origin": "1551199150603",
      "device": "Random-Integer-Generator01",
      "name": "RandomValue_Int8",
      "value": "99"
    }
  ]
}
```


THANK YOU