**(ICBMS-3세부) 사물 가상화, 분산 자율지능 및 데이터 연계/분석을 지원하는 IoT 기반 플랫폼 기술 개발**

**SI (서비스 연동 프레임워크)**

**SI oneM2M Server**

**메뉴얼**

**Version 1.0**



**문서승인**

**주문서의 부속문서로 승인을 받는 경우 주문서의 서명으로 대체함.**

**수행사: ㈜헤리트**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

이인석 이사 Date

개발팀

**주관사: ㈜파인원커뮤니케이션즈**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

김태철 팀장 Date

개발팀

**제.개정 이력서**

|  |  |  |
| --- | --- | --- |
| 개정번호 | 개정 페이지 및 내용 | 개정일자 |
| 0.1 | 초안 작성 | 2018/01/10 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

목차

[1. SI oneM2M Server 7](#_Toc503951792)

[1.1. Introduction 7](#_Toc503951793)

[1.2. SI oneM2M Server 7](#_Toc503951794)

[1.2.1. System Structure 7](#_Toc503951795)

[1.2.2. Connectivity Structure 8](#_Toc503951796)

[1.2.3. Software Structure 8](#_Toc503951797)

[1.2.4. Supported Addressing Format 9](#_Toc503951798)

[1.2.5. Supported Protocol Bindings 9](#_Toc503951799)

[1.2.6. Supported Serialization data format 10](#_Toc503951800)

[1.2.7. Documentation 10](#_Toc503951801)

[1.2.7.1. SI oneM2M Server source code package 10](#_Toc503951802)

[1.2.7.2. Library Dependencies 10](#_Toc503951803)

[1.2.8. Installation 11](#_Toc503951804)

[1.2.9. 설치 순서 12](#_Toc503951805)

[2. SI oneM2M Server Installation 13](#_Toc503951806)

[2.1. Pre-requisites Installation for Windows 13](#_Toc503951807)

[2.1.1. MongoDB Installation 13](#_Toc503951808)

[2.1.1.1. Download 13](#_Toc503951809)

[2.1.1.2. Install 13](#_Toc503951810)

[2.1.1.3. Configuration(초기 환경 설정) 13](#_Toc503951811)

[2.1.2. Java Installation 14](#_Toc503951812)

[2.1.2.1. Download 14](#_Toc503951813)

[2.1.2.2. Install 15](#_Toc503951814)

[2.1.3. Eclipse Installation 15](#_Toc503951815)

[2.1.3.1. Download 15](#_Toc503951816)

[2.1.3.2. Configuration(Installed JREs) 16](#_Toc503951817)

[2.1.4. Postman Installation 17](#_Toc503951818)

[2.1.4.1. Download and Installation 17](#_Toc503951819)

[2.1.4.2. Configuration(초기 환경 설정) 18](#_Toc503951820)

[2.1.5. Mosquitto 19](#_Toc503951821)

[2.1.5.1. Download 19](#_Toc503951822)

[2.1.5.2. Install 20](#_Toc503951823)

[2.2. SI oneM2M Server Installation 24](#_Toc503951824)

[2.2.1. Download 24](#_Toc503951825)

[2.2.2. Source Import 24](#_Toc503951826)

[2.2.3. Maven Clean 25](#_Toc503951827)

[2.2.4. Maven Update 27](#_Toc503951828)

[2.2.5. Project Build 28](#_Toc503951829)

[2.2.6. Maven Install 29](#_Toc503951830)

[2.3. SI oneM2M Server Configuration 30](#_Toc503951831)

[2.3.1. Database Configuration 30](#_Toc503951832)

[2.3.2. Binding Protocol Configuration 30](#_Toc503951833)

[2.3.3. CSE Configuration 31](#_Toc503951834)

[2.3.4. DM Configuration 31](#_Toc503951835)

[2.4. Run SI oneM2M Server 32](#_Toc503951836)

[2.4.1. Run MongoDB 32](#_Toc503951837)

[2.4.2. Run SI oneM2M Server 33](#_Toc503951838)

[3. SI oneM2M Server Test 35](#_Toc503951839)

[3.1. Test for HTTP Protocol 35](#_Toc503951840)

[3.1.1. Postman Usage 35](#_Toc503951841)

[3.1.1.1. 항목 선택 35](#_Toc503951842)

[3.1.1.2. URI 입력 35](#_Toc503951843)

[3.1.1.3. Headers 설정 36](#_Toc503951844)

[3.1.1.4. Body 설정 36](#_Toc503951845)

[3.1.2. CSE-Base Retrieve 36](#_Toc503951846)

[3.1.2.1. Request URI 36](#_Toc503951847)

[3.1.2.2. Request Header 36](#_Toc503951848)

[3.1.3. AE Creation 37](#_Toc503951849)

[3.1.3.1. Request URI 37](#_Toc503951850)

[3.1.3.2. Request Header 37](#_Toc503951851)

[3.1.3.3. Request Body 37](#_Toc503951852)

[3.1.4. Container Creation 38](#_Toc503951853)

[3.1.4.1. Request URI 38](#_Toc503951854)

[3.1.4.2. Request Header 38](#_Toc503951855)

[3.1.4.3. Request Body 38](#_Toc503951856)

[3.1.5. contentInstance Creation 38](#_Toc503951857)

[3.1.5.1. Request URI 38](#_Toc503951858)

[3.1.5.2. Request Header 38](#_Toc503951859)

[3.1.5.3. Request Body 39](#_Toc503951860)

# SI oneM2M Server

## Introduction

SI oneM2M Server는 IoT 국제 표준인 oneM2M 기반의 디바이스 및 애플리케이션 연동을 지원하는 서버 프레임워크(IN-CSE)입니다. SI oneM2M Server 소스를 활용하여 oneM2M 기반의 디바이스 및 애플리케이션 연동 서버를 구축할 수 있을뿐만 아니라 oneM2M 코어 소스를 활용하여 AE, MN-CSE등 다양한 oneM2M 컴포넌트를 개발 할 수 있습니다.

## SI oneM2M Server

### System Structure



### Connectivity Structure



### Software Structure



### Supported Addressing Format

* Structured SP relative addressing format
  + hierarchical하게 구성되어 있는 service provider 기준의 주소체계로서, 리소스 URI 규격은 다음과 같다.
  + /{cse-id}/{cse-name}/{resource-name}/{resource-name}}/….
* Structured CSE relative addressing format
  + Hierarchical 하게 구성되어 있는 CSE 기준으로 구성된 주소체계로서, 다음과 같은 리소스 URI 규격을 갖는다.
  + /{cse-name}/{resource-name}/{resource-name}}/….
* Structured Absolute addressing format
  + 고유의 도메인으로 중심으로 Hierarchical하게 구성된 주소체계로서, 리소스 URI 규격은 다음과 같다.
  + //www.memprovider.com/{cse-id}/{cse-name}/{resource-name}/{resource-name}/….
* Unstructured SP relative addressing format
  + Hierarchical 하지 않게 구성되어 있는 Service Provider 관점에서의 주소체계로서, 다음과 같은 리소스 URI 규격을 갖는다.
  + /{cse-id}/{cse-name}/{resource-id}
* Unstructured CSE relative addressing format
  + Hierarchical 하지 않게 구성되어 있는 CSE 관점에서의 주소체계로서, 다음과 같은 리소스 URI 규격을 갖는다.
  + /{cse-name}/{resource-id}
* Unstructured Absolute addressing format
  + 고유의 도메인으로 중심으로 Hierarchical하지 않게 구성된 주소체계로서, 리소스 URI 규격은 다음과 같다.
  + //www.memprovider.com/{cse-id}/{cse-name}/{resource-id}

### Supported Protocol Bindings

* HTTP
  + oneM2M 규격 중 TS-0009-HTTP-Protocol Binding 문서 참조
* CoAP
  + oneM2M 규격 중 TS-0008-CoAP-Protocol Binding 문서 참조
* MQTT
  + oneM2M 규격 중 TS-0010-MQTT-Protocol Binding 문서 참조
* WebSocket
  + oneM2M 규격 중 TS-0020-WebSocket-Protocol Binding 문서 참조

### Supported Serialization data format

* XML
  + 위의 4종 프로토콜 바인딩 규격에 모두 적용 가능
* JSON
  + 위의 4종 프로토콜 바인딩 규격에 모두 적용 가능
* CBOR
  + 위의 4종 프로토콜 바인딩 규격에 모두 적용 가능

### Documentation

#### SI oneM2M Server source code package

|  |  |
| --- | --- |
| Package | Role and Function |
| net.herit.iot.db.mongo | MongoDB 커넥션 풀을 담당하는 패키지. |
| net.herit.iot.message.onem2m | oneM2M 상수값들이 모인 패키지이다. oneM2M의 Content-type, CSE type, Request type, Member type등이 있으며, Operation에 대한 request 처리와 처리 결과를 알려주는 Response Code도 정의되어 있다. |
| net.herit.iot.onem2m.bind | Binding Protocol을 정의한 패키지. HTTP, CoAP, MQTT, WebSocket이 정의되어 있으며, 각각의 프로토콜은 MQTT를 제외하고 Client-Server model로 구현되어있다. (MQTT의 Server인 Broker는 지원하지 않는다.) |
| net.herit.iot.onem2m.core | 여러형태의 Data를 처리하기 위한 Data Format Convertor와 Utility를 정의한 패키지이다. JSON Convertor와 XML Convertor가 있으며, CBOR의 경우 라이브러리를 활용하여 Encode/Decode 동작이 수행된다. |
| net.herit.iot.onem2m.incse | oneM2M resource의 초기화 및 CRUDN을 담당하는 manager 패키지와 oneM2M, TR-069의 DM Adaptor와 DM Controller가 존재하는 dm 패키지 등으로 구성되어 있다. |
| net.herit.iot.onem2m.resource | oneM2M 표준을 기반한 resource를 정의한 패키지. oneM2M resource는 각각 하나의 객체로써 java파일로 1:1 매핑되어 구성되며, oneM2M.org에서 정의한 xsd schema를 참조하여 구현된다. |

#### Library Dependencies

The list of library dependencies for SI oneM2M Server

* scandium-1.0.0.jar
* element-connector-1.0.0.jar
* org.eclipse.paho.client.mqttv3-1.0.2.jar
* slf4j-api-1.7.13.jar
* logback-classic-1.1.3.jar
* logback-core-1.1.3.jar
* commons-codec-1.10.jar
* httpclient-4.5.1.jar
* httpcore-4.4.3.jar
* commons-logging-1.2.jar
* netty-all-4.0.33.Final.jar
* org.eclipse.persistence.core-2.4.2.jar
* org.eclipse.persistence.asm-2.4.2.jar
* org.eclipse.persistence.moxy-2.4.2.jar
* org.eclipse.persistence.antlr-2.4.2.jar
* joda-time-2.8.2.jar
* mongo-java-driver-3.1.1.jar
* commons-configuration-1.6.jar
* commons-collections-3.2.1.jar
* commons-lang-2.4.jar
* commons-digester-1.8.jar
* commons-beanutils-1.7.0.jar
* commons-beanutils-core-1.8.0.jar
* Java-WebSocket-1.3.0.jar
* jackson-mapper-asl-1.8.5.jar
* jackson-core-asl-1.8.5.jar
* xstream-1.4.7.jar
* xmlpull-1.1.3.1.jar
* xpp3\_min-1.1.4c.jar
* json-20140107.jar
* json-simple-1.1.jar
* jackson-databind-2.8.7.jar
* jackson-annotations-2.8.0.jar
* jackson-core-2.8.7.jar
* jackson-dataformat-cbor-2.8.7.jar
* cbor-0.7.jar
* commons-httpclient-3.1.jar
* quartz-1.7.3.jar
* spring-core-4.3.10.RELEASE.jar

### Installation

* MongoDB

스케일아웃 구조여서 쉽게 운용이 가능하며 JSON형태로 저장이 가능해서 직관적이다. 또한 문서지향적 Query Lnaguage를 사용하여 SQL만큼 강력한 Query 성능을 제공한다. 이와 같은 이유로 SI oneM2M Server의 Database로 사용되고 있다.

* Java 7

Native Language를 제외하고 가장 빠른축에 속하는 언어로, Native Language보다 개발 소요시간이 적어 효율이 매우 좋기에 메인 개발 언어로 선택되었다.

* Eclipse

개발툴로써 꼭 Eclipse를 써야 하는건 아니지만 Java개발에 최적화 되어 있어, 매우 편하다

* Postman

Chrome에서 제공하는 HTTP메시징 툴로써, 설치부터 사용까지 매우 간편하다. SI oneM2M Server의 설치 완료 확인 및 테스트하는 용도로 사용된다.

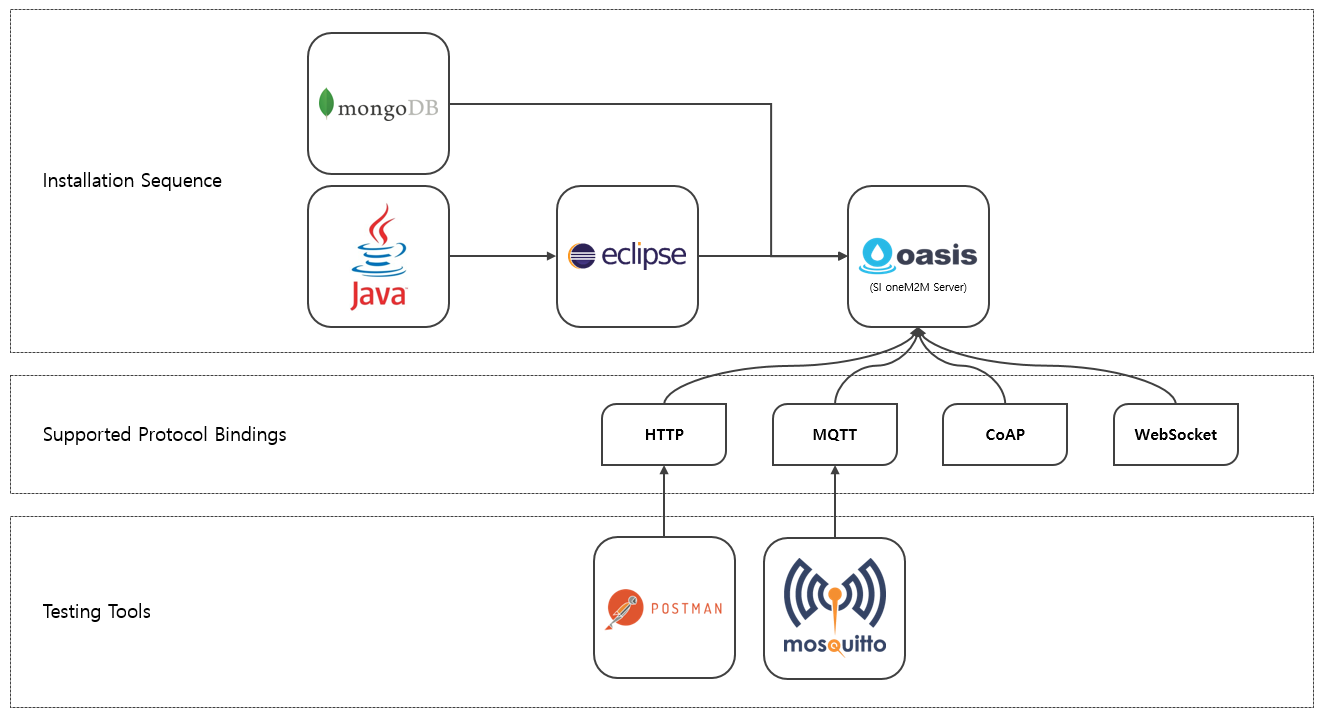
* Mosquitto

MQTT 프로토콜은 메시지를 발행(publishing)하고 관심있는 주제(topic)를 구독(subscribe)하는 것을 기본 원칙으로 하는데, 이 때 Publisher와 Subscriber는 모두 Broker에 대한 클라이언트로 작동한다. Publisher는 토픽을 발행하기 위한 목적으로, Subscriber는 토픽을 구독하기 위한 목적으로 Broker 서버에 연결하며 이 Broker 역할을 하는게 Mosquitto이다.

* SI oneM2M Server

OASIS 플랫폼에서의 SI(System Integration)영역의 Server로, 국제 표준인 oneM2M을 기반으로 만들어진 Server이다. (oneM2M에서의 IN-CSE역할)

### 설치 순서



# SI oneM2M Server Installation

## Pre-requisites Installation for Windows

### MongoDB Installation

#### Download

* <https://www.mongodb.com/download-center#community> 에서 다운로드

|  |
| --- |
|  |
| 그림-1. 다운로드 페이지 일부 |

#### Install

* Next 버튼을 눌러서 설치 진행

|  |
| --- |
|  |
| 그림-2. 설치 화면 |

#### Configuration(초기 환경 설정)

* MongoDB가 데이터를 저장할 경로를 지정해줘야 하므로 C드라이브(MongoDB가 설치된 메인 드라이브)에 ‘data’폴더를 생성하고 그 안에 ‘db’폴더를 생성한다.
* (‘그림-3’의 경우 MongoDB가 D드라이브에 설치되었기에 D드라이브에 폴더를 생성함)

|  |
| --- |
|  |
| 그림-3. 폴더 생성 화면 |

### Java Installation

#### Download

* [http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase7-521261.html](http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase7-521261.html%20) 에서 JDK 7u80 버전 다운로드
* (JDK 7을 사용하지 않을 경우 SI oneM2M Server의 일부 기능이 동작하지 않을 수 있음)

|  |
| --- |
|  |
| 그림-4. OS별 JDK 7u80리스트 |

#### Install

* Next 버튼을 눌러 설치 진행

|  |
| --- |
|  |
| 그림-5. 설치 화면 |

### Eclipse Installation

#### Download

* <http://www.eclipse.org/downloads/packages/release/Mars/2> 에서 ‘Eclipse IDE for Java Developer’ 다운로드(JDK7과 잘 호환되는 Eclipse 버전인 Mars로 사용)

|  |
| --- |
|  |
| 그림-6. Eclipse Donwload Page 일부 |

#### Configuration(Installed JREs)

* ‘Window’ 메뉴 아래의 ‘Preference’ 메뉴 클릭

|  |
| --- |
|  |
| 그림-7. Preferences 메뉴 |

* 왼쪽 검색창에 ‘Installed JREs’를 검색하여 우측의 ‘Edit…’ 버튼 클릭

|  |
| --- |
|  |
| 그림-8. Preferences의 Installed JREs 설정 |

* ‘JRE home’에 jre가 설정되어 있다면 ‘Directory…’버튼을 눌러 jdk로 변경

|  |
| --- |
|  |
| 그림-9. JRE home 수정전(왼쪽)과 수정후(오른쪽) |

### Postman Installation

#### Download and Installation

* <https://chrome.google.com/webstore/detail/postman/fhbjgbiflinjbdggehcddcbncdddomop> 에서 다운로드(Chrome 브라우저 필요)

|  |
| --- |
|  |
| 그림-10. Postman 다운로드 화면 |

#### Configuration(초기 환경 설정)

* 스크립트 추가를 위해 Postman 실행 후 Import 버튼 클릭

|  |
| --- |
|  |
| 그림-11. Postman 스크립트 추가 버튼 위치 |

* SI oneM2M Server 소스에 들어있는 postman-collections 폴더내의 collection을 그림-9에서 누른 ‘Import’ 버튼 클릭하면 나오는 창에 드래그 & 드랍으로 등록

|  |
| --- |
|  |
| 그림-12. collection을 드래그&드랍 하는 모습 |

|  |
| --- |
|  |
| 그림-13. collection이 정상적으로 import된 모습 |

### Mosquitto

#### Download

* <https://mosquitto.org/download> 에서 mosquitto 다운로드

|  |
| --- |
|  |
| 그림-14. 다운로드 페이지 일부 |

#### Install

* 그림-15에서 Next를 누르면 그림-16과 같은 화면이 나오는데, OpenSSL의 dll 파일과 pthreads의 pthreadVC2.dll이 필요하므로, 이 파일들이 없다면 받으라는 안내메시지이다.

|  |
| --- |
|  |
| 그림-15. 설치 화면-1 |

|  |
| --- |
|  |
| 그림-16. 설치 화면-2 |

* OpenSSL 설치가 필요하므로 <http://slproweb.com/products/Win32OpenSSL.html>에 접속하여 'Win32 OpenSSL [버전] Light’를 다운 받아 설치

|  |
| --- |
|  |
| 그림-17. OpenSSL download 페이지 일부 |

* 다운받아 설치하다보면 그림-18과 같은 화면이 나오는데, 아래쪽의 The OpenSSL binaries(/bin) directory를 선택하여 설치한다.(dll파일을 찾기 쉽게)

|  |
| --- |
|  |
| 그림-18. OpenSSL 설치 과정 일부 |

* pthreads의 library가 필요하므로 <ftp://sources.redhat.com/pub/pthreads-win32/dll-latest/dll/x86/>에 접속하여 ‘pthreadVC2.dll’ 파일을 다운로드

|  |
| --- |
|  |
| 그림-19. pthreads library가 있는 ftp 디렉터리 |

* 처음으로 돌아가 다시 mosquitto 설치를 마저 진행한다

|  |
| --- |
|  |
| 그림-20. mosquitto 설치 완료 화면 |

* mosquitto가 설치된 폴더에 아까 받은 OpenSSL의 dll파일과 pthreadVC2.dll을 복사-붙여넣기 한다.(libeay32.dll, ssleay32.dll이 OpenSSL설치경로의 bin폴더에 없다면 Win32 OpenSSL v1.0.2n Light를 설치)

|  |
| --- |
|  |
| 그림-21. mosquitto 폴더로 복사한 dll파일들 |

## SI oneM2M Server Installation

### Download

* <https://github.com/iotoasis/SI> 에서 ‘Clone or download’ 버튼을 클릭하여 download

|  |
| --- |
|  |
| 그림-22. github 화면 일부 |

### Source Import

* Eclipse 실행 후 다운 받은 소스 import

|  |
| --- |
|  |
| 그림-23. Eclipse Import 과정-1 |

|  |
| --- |
|  |
| 그림-24. Eclipse Import 과정-2 |

|  |
| --- |
|  |
| 그림-25. Eclipse Import 과정-3 |

### Maven Clean

* 프로젝트 우클릭 -> Run As -> Maven clean

|  |
| --- |
|  |
| 그림-26. Maven clean 선택 화면 |

### Maven Update

* Alt+F5를 눌러서 프로젝트 업데이트

|  |
| --- |
|  |
| 그림-27. Update Maven Project |

### Project Build

* 프로젝트 우클릭 -> Build Project

|  |
| --- |
|  |
| 그림-28. Build Project |

### Maven Install

* 프로젝트 우클릭 -> Run As -> Maven install

|  |
| --- |
|  |
| 그림-29. Maven Install 화면 |

|  |
| --- |
|  |
| 그림-30. Maven Install 완료 화면 |

## SI oneM2M Server Configuration

### Database Configuration

|  |
| --- |
| <database>  <host>localhost</host>  <port>27017</port>  <dbname>IITP-IOT</dbname>  <user>herit</user>  <password>qrwe1423</password>  </database> |

### Binding Protocol Configuration

|  |
| --- |
| <http>  <port>8080</port>  <sec-port>8443</sec-port> <!-- https port -->  <rest-port>8081</rest-port>  <netty>  <boss-threadPool-size>0</boss-threadPool-size>  <worker-threadPool-size>1024</worker-threadPool-size>  </netty>  </http> |
| ( HTTP ) |

|  |
| --- |
| <mqtt>  <supported>yes</supported> <!-- yes or no -->  <keepalive>1200</keepalive> <!-- seconds -->  <broker>tcp://10.10.202.146:1883</broker>  </mqtt> |
| ( MQTT ) |

|  |
| --- |
| <coap>  <supported>yes</supported> <!-- yes or no -->  <port>5683</port>  <sec-port>4433</sec-port> <!-- dtls port -->  <cse-unstruct>false</cse-unstruct> <!-- cse-relative unstructured addressing -->  </coap> |
| ( CoAP ) |

|  |
| --- |
| <websocket>  <supported>yes</supported> <!-- yes or no -->  <port>8887</port>  </websocket> |
| ( WebSocket ) |

### CSE Configuration

* SI oneM2M Server정보 및 poa등 설정

|  |
| --- |
| <cse>  <host>10.10.0.23</host>  <baseName>herit-cse</baseName>  <resourceId>herit-in</resourceId>  <poa>http://10.10.0.23:8080</poa>  <default-resource-expiration-time>3</default-resource-expiration-time> <!-- days -->  <allowed-subgroup-depth>3</allowed-subgroup-depth>  </cse> |

|  |  |
| --- | --- |
| 항목 | 역할 |
| host | SI oneM2M Server가 동작하는 Server의 IP |
| baseName | SI oneM2M Server가 사용할 CSE-Base Name |
| resourceId | SI oneM2M Server가 사용할 CSE Resource ID |
| poa | poa |
| default-resource-expiration-time | oneM2M Resource 생성시점부터 기간만료까지 걸리는 시간(et) |
| allowed-subgroup-depth | 그룹내의 그룹을 생성할 때의 최대 depth |

### DM Configuration

* DM 관련 기능을 위한 설정으로, DM을 사용하지 않는 다면 특별한 값을 입력하지 않아도 상관없다.

|  |
| --- |
| <dms>  <hitdm>  <address>http://10.101.101.107:8888</address>  </hitdm>  <tr69dm>  <address>http://10.10.0.23:7557</address>  <timeout>3000</timeout>  </tr69dm>  <onem2mAgent>  <address>http://10.10.0.81:1337</address>  </onem2mAgent>  </dms> |

|  |  |
| --- | --- |
| 항목 | 역할 |
| hitdm.address | Herit DM 서버 주소 |
| tr69dm.address | TR-069의 Server인 ACS의 address |
| tr69dm.timeout | ACS의 응답 대기 시간(timeout)으로 기본적으로 3000을 사용한다. |
| onem2mAgent.address | oneM2M Agent의 address |

## Run SI oneM2M Server

### Run MongoDB

* MongoDB(mongod.exe) 실행

|  |
| --- |
|  |
| 그림-31. MongoDB 실행 화면 |

### Run SI oneM2M Server

* 프로젝트 우클릭 -> Run As -> Java Application

|  |
| --- |
|  |
| 그림-32. Run As Java Application |

* ‘incse’를 선택 후 ‘OK’버튼 클릭하여 실행

|  |
| --- |
|  |
| 그림-33. Run InCse |

# SI oneM2M Server Test

## Test for HTTP Protocol

### Postman Usage

#### 항목 선택

|  |
| --- |
|  |
| 그림-34. 항목 선택 |

#### URI 입력

|  |
| --- |
|  |
| 그림-35. URI 입력 |

#### Headers 설정

|  |
| --- |
|  |
| 그림-36. Headers 설정 |

#### Body 설정

|  |
| --- |
|  |
| 그림-37. Body 설정 |

### CSE-Base Retrieve

#### Request URI

<http://localhost:8080/herit-in/herit-cse>

#### Request Header

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Value | Description | Mandatory |
| Accept | application/json | Response Body Type | Optional |
| HOST | 10.10.101.193 | The FQDN or IP address of the Receiver CSE of the next hop in multi-hop commnunication senarios. | Optional |
| X-M2M-RI | pm\_1506031019180001 | The X-M2M-RI Header shall be mapped to the *Request Identifier* parameter of request and response privitives and vice versa. | Mandatory |
| X-M2M-Origin | //iot.herit.net/herit-cse/CAE0001 | The X-M2M-Origin header shall be mapped to the *From* parameter of request and response primitives and vice versa, if applicable. | Mandatory |

### AE Creation

#### Request URI

<http://localhost:8080/herit-in/herit-cse>

#### Request Header

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Value | Description | Mandatory |
| Accept | application/json | Response Body Type | Optional |
| HOST | 10.10.101.193 | The FQDN or IP address of the Receiver CSE of the next hop in multi-hop commnunication senarios. | Optional |
| X-M2M-RI | pm\_1506031019180001 | The X-M2M-RI Header shall be mapped to the *Request Identifier* parameter of request and response privitives and vice versa. | Mandatory |
| X-M2M-Origin | //iot.herit.net/herit-cse/CAE0001 | The X-M2M-Origin header shall be mapped to the *From* parameter of request and response primitives and vice versa, if applicable. | Mandatory |
| Content-Type | application/vnd.onem2m-res+json; ty=2 | application/vnd.onem2m-res+[data\_type]; ty=[resource\_type] | Mandatory |

#### Request Body

|  |
| --- |
| {  "m2m:ae" : {  "lbl" : [ "hubiss", "admin" ],  "et" : "20181103T122321",  "apn" : "onem2mPlatformAdmin",  "poa" : [ "10.10.0.23" ],  "api" : "testapp",  "rr" : false  }  } |

### Container Creation

#### Request URI

<http://localhost:8080/herit-in/herit-cse/device0001>

#### Request Header

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Value | Description | Mandatory |
| Accept | application/json | Response Body Type | Optional |
| HOST | 10.10.101.193 | The FQDN or IP address of the Receiver CSE of the next hop in multi-hop commnunication senarios. | Optional |
| X-M2M-RI | pm\_1506031019180001 | The X-M2M-RI Header shall be mapped to the *Request Identifier* parameter of request and response privitives and vice versa. | Mandatory |
| X-M2M-Origin | //iot.herit.net/herit-cse/CAE0001 | The X-M2M-Origin header shall be mapped to the *From* parameter of request and response primitives and vice versa, if applicable. | Mandatory |
| Content-Type | application/vnd.onem2m-res+json; ty=3 | application/vnd.onem2m-res+[data\_type]; ty=[resource\_type] | Mandatory |

#### Request Body

|  |
| --- |
| {  "m2m:cnt" : {  “rn” : “temperature”,  "lbl" : [ "hubiss", "admin", “key1” ],  "et" : "20181103T122321",  "mni" : 100,  "mbs" : 1024000,  "mia" : "36000"  }  } |

### contentInstance Creation

#### Request URI

<http://localhost:8080/herit-in/herit-cse/device0001/temperature>

#### Request Header

|  |  |  |  |
| --- | --- | --- | --- |
| Key | Value | Description | Mandatory |
| Accept | application/json | Response Body Type | Optional |
| HOST | 10.10.101.193 | The FQDN or IP address of the Receiver CSE of the next hop in multi-hop commnunication senarios. | Optional |
| X-M2M-RI | pm\_1506031019180001 | The X-M2M-RI Header shall be mapped to the *Request Identifier* parameter of request and response privitives and vice versa. | Mandatory |
| X-M2M-Origin | //iot.herit.net/herit-cse/CAE0001 | The X-M2M-Origin header shall be mapped to the *From* parameter of request and response primitives and vice versa, if applicable. | Mandatory |
| Content-Type | application/vnd.onem2m-res+json; ty=4 | application/vnd.onem2m-res+[data\_type]; ty=[resource\_type] | Mandatory |

#### Request Body

|  |
| --- |
| {  "m2m:cin" : {  "lbl" : [ "hubiss", "admin" ],  "et" : "20181103T122321",  "cnf" : “text/plain:0”,  "con" : 24  }  } |