


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

# SDA 프레임워크 오픈 소스 메뉴얼

요약 : SDA 프레임워크를 적용하여 개발하는 방법을 설명한다.

 <b>PINEONE</b>	<b>SDA 프레임워크 개발가이드</b>		
프로젝트명:	단계:	담당팀:	

## 문서승인

주문서의 부속문서로 승인을 받는 경우 주문서의 서명으로 대체함.

수행사: (주)파인원커뮤니케이션즈

\_\_\_\_\_  
 배운호 책임  
 개발팀

\_\_\_\_\_  
 Date

주관사: (주)파인원커뮤니케이션즈

\_\_\_\_\_  
 김태철 팀장  
 개발팀

\_\_\_\_\_  
 Date

## 제.개정 이력서

개정번호	개정 페이지 및 내용	개정일자
0.1	1. 목차 정의	2016.11.28
1.0	내용 수정	2016.12.28
2.0	내용 수정 및 추가(구성도 및 SDA 모듈 내용 추가)	2018.01.10

## 목차

1. 개요	5
1.1 SDA (Semantic Data Analysis)	5
1.2 SDA (Semantic Data Analysis) 구성도	6
2. 주요기능	6
3. 개발 방법	7
3.1 Quick Start	7
3.1.1 Requirements	7
3.1.2 따라하기	7
3.2 SDA 서버 설정	9
3.3 SDA Framework Server Build	11
3.3.1 Requirements	11
4. Module	13
4.1 sda-web : RESTFul API 및 web 서비스등을 제공하기 위한 web 모듈	13
4.2 sda-client : 독립적으로 실행되는 client 프로그램을 가지고 있는 모듈	13
4.3 sda-common : sda 의 core 모듈로 공통적인 기능과 core 기능을 포함하고 있는 모듈	14
5. Q&A	15

# 개발 가이드

## 1. 개요

Oasis (Open-source Architecture Semantic lot Service-platform) 프로젝트는 국제 표준을 준용하는 오픈 소스 기반 지능형 사물 인터넷 서비스 플랫폼을 개발하는 것을 목표로 하고 있습니다.

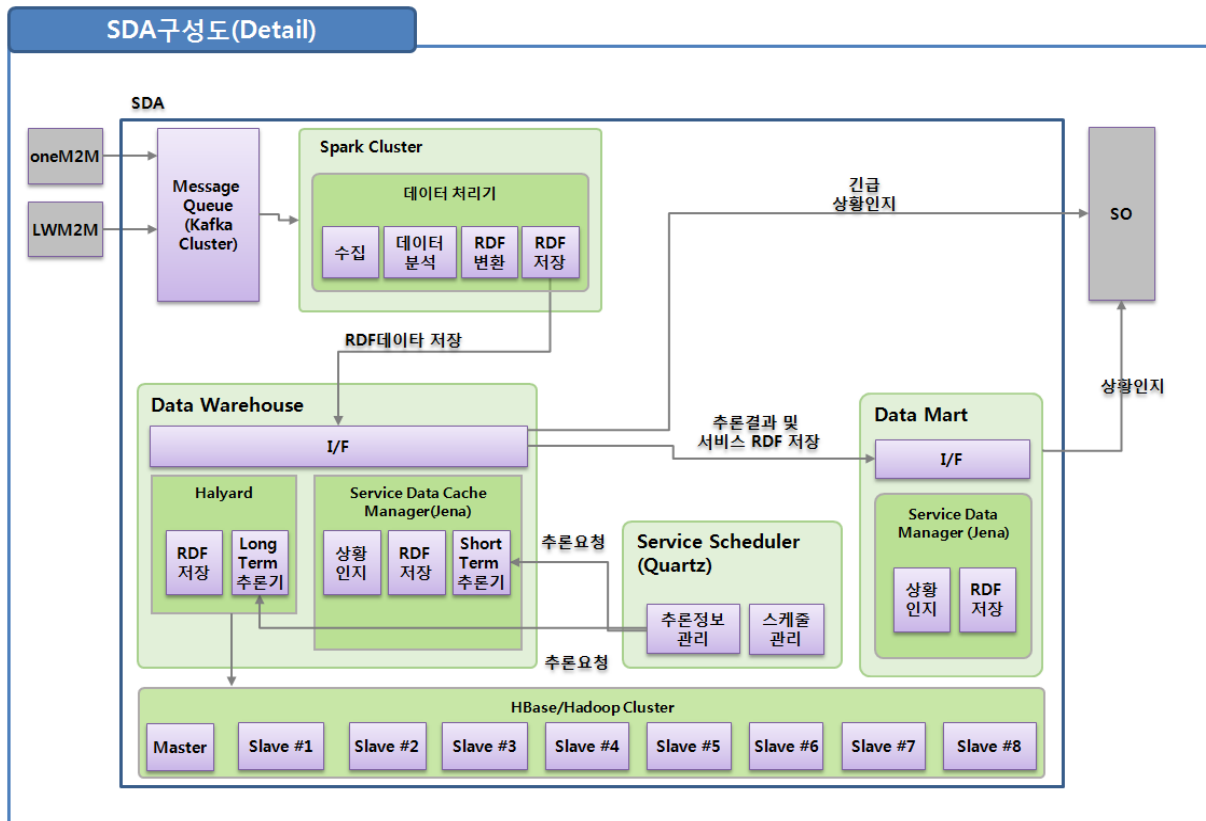
Oasis 프로젝트는 2015 년도 정보통신. 방송 기술개발사업 신규지원 대상과제 "(ICBMS-3 세부) 사물 가상화, 분산 자율지능 및 데이터 연계/분석을 지원하는 IoT 기반 플랫폼 기술 개발" 과제의 결과물로써 오픈소스로 제공됩니다.

본 Oasis 프로젝트는 오픈 소스 커뮤니티를 기반으로 오픈소스로써 계속적으로 성장해 나갈 계획입니다.

### 1.1 SDA (Semantic Data Analysis)

SDA 는 IoT 환경에서 발생하는 센서/디바이스 데이터, 레거시 데이터를 수집, 분석, 제공하는 기능을 지원하는 데이터 분석 프레임워크 입니다.

## 1.2 SDA (Semantic Data Analysis) 구성도



## 2. 주요기능

- OneM2M 표준 플랫폼 기반 데이터 수집
- RDF/OWL 등 시맨틱 웹 표준 준수
- Semantic Annotation
- 센서네트워크 온톨로지(SSNO), 상황인지(DUL) 등 IoT 도메인 전용 레퍼런스 모델 적용
- 실시간 상황인지를 위한 추론 기능

### 3. 개발 방법

#### 3.1 Quick Start

Oasis SDA FrameWork 를 처음 접하는 분들이 소스를 다운받고 쉽게 시험할 수 있도록 안내합니다.

SDA Server 시험은 아래의 순서로 진행할 수 있습니다.

1. MariaDB, JDK 다운로드 및 설치
2. Oasis SDA 소스 다운로드
3. MariaDB db 및 테이블 생성
4. SDA 빌드
5. SDA 설정 및 웹모듈 실행

##### 3.1.1 Requirements

- JDK 7+
- MariaDB 10.1.X+
- Maven 3.3.X+
- Tomcat 7.0.X+
- Windows / Linux

##### 3.1.2 따라하기

###### 3.1.2.1 MariaDB 다운로드 및 설치, JDK 다운로드 및 설치

- [MariaDB 설치안내](#)
- [JDK 설치안내](#)

###### 3.1.2.2 Oasis SDA 소스 다운로드

- [릴리즈 페이지](#)에서 SDA 소스 및 설치관련 파일을 다운받는다.

### 3.1.2.3 MariaDB 기본 셋팅

- [다운로드](#)에서 다운받은 mariadb 스크립트파일(mariadb\_script.txt)를 실행시켜준다.

### 3.1.2.4 SDA 빌드

- [릴리즈 페이지](#)에서 다운받은 SDA 소스를 이클립스에서 불러와서 Build 한다.(Maven 이용함)
- SDA 소스를 빌드하는 방법은 [소스 Build 방법](#)페이지를 참고한다.

### 3.1.2.5 SDA 설정 및 실행

- 다운 받은 소스의 sda-common/resources/system.properties 파일을 오픈하여 SDA 설정을 수정한다.
- SDA 설정방법은 [SDA 서버 설정방법](#)페이지를 참고한다.



### 3.2 SDA 서버 설정

SDA 서버 설정방법을 설명합니다.

설정은 properties 파일로 작성되며 sda-common 폴더의 resources 폴더에 존재합니다.

```
system.properties
```

설정항목별 의미는 아래와 같습니다.

- jena 운영  
com.pineone.icbms.sda.knowledgebase.host=XXX.XXX.XXX.XXX  
com.pineone.icbms.sda.knowledgebase.public.host=XXX.XXX.XXX.XXX
- fuseki 운영  
com.pineone.icbms.sda.knowledgebase.sparql.endpoint=<http://XXX.XXX.XXX.XXX:포트/icbms> com.pineone.icbms.sda.knowledgebase.host.port=포트  
com.pineone.icbms.sda.knowledgebase.uri=<http://www.iotoasis.org>
- SI 운영  
com.pineone.icbms.sda.mongodb.server=XXX.XXX.XXX.XXX  
com.pineone.icbms.sda.mongodb.port=포트 com.pineone.icbms.sda.mongodb.db=db 명
- data 수집 1 회 처리개수  
com.pineone.icbms.sda.mongodb.read\_limit=20000
- DB 정보운영  
com.pineone.icbms.sda.ss.db.server=XXX.XXX.XXX.XXX com.pineone.icbms.sda.ss.db.port=포트  
com.pineone.icbms.sda.ss.db.name.timetable=db 명  
com.pineone.icbms.sda.ss.db.name.device=db 명 com.pineone.icbms.sda.ss.db.user=사용자계정  
com.pineone.icbms.sda.ss.db.pass=패스워드

- fuseki 의 s-post 위치  
com.pineone.icbms.sda.triple.regist.bin=/fuseki 설치위치/bin/s-post
- triple 파일 저장위치  
com.pineone.icbms.sda.triple.save\_path=/triple 저장폴더/triples
- SDA 운영서버의 콜백 URI  
com.pineone.icbms.sda.si.notification\_uri=<http://XXX.XXX.XXX.XXX:포트/sda/subscribe/callback>
- SI 운영(description 설정용)  
com.pineone.icbms.sda.si.subscription\_uri=<http://XXX.XXX.XXX.XXX:포트번호/경로/>
- SO 운영(callback 용)  
com.pineone.icbms.sda.so.callback\_result\_uri=<http://XXX.XXX.XXX.XXX:포트번호/경로/>
- riot 모드  
--skip : riot 적용하지 않음  
--check : 전체 파일을 체크하여 오류가 나는 부분 추출 → 개발중에 활용  
--validate : 파일을 스캔하다가 문법의 오류가 있으면 바로 파싱을 멈추고 오류 발생시 종료  
com.pineone.icbms.sda.riot.bin=/jena 설치위치/bin/riot com.pineone.icbms.sda.riot.mode=--skip com.pineone.icbms.sda.riot.result.save\_path=/riot 결과저장폴더/riot-result
- kafka 쓰레드 개수  
com.pineone.icbms.sda.kafka.thread.count=3
- lasttestedContentInstance 를 구하는 기준시간을 구할때 사용되는 마이너스 할 시간을 ms 단위로 지정한다.  
(현재시간 - 10 초) 부터 유효하게 하려면 1000\*10 값을 지정한다.  
com.pineone.icbms.sda.init.adjust.ms=10000

-통계 쿼리를 수행할 db 접속정보

com.pineone.icbms.sda.stat.db.server=XXX.XXX.XXX.XXX com.pineone.icbms.sda.stat.db.port=포트 번호

프로젝트명:

단계:

담당팀:

com.pineone.icbms.sda.stat.db.name=db 명 com.pineone.icbms.sda.stat.db.user=사용자계정  
com.pineone.icbms.sda.stat.db.pass=패스워드

-mongodb 정보(통계용)

com.pineone.icbms.sda.mongo.db.server=XXX.XXX.XXX.XXX

com.pineone.icbms.sda.mongo.db.port=포트번호 com.pineone.icbms.sda.mongo.db.name=db 명

com.pineone.icbms.sda.mongo.db.collection.name=collection 명

### 3.3 SDA Framework Server Build

다운받은 SDA Framework 서버 소스를 Eclipse 를 이용해서 Build 하는 방법을 설명합니다.

Build 를 위한 요구조건은 아래와 같습니다.

#### 3.3.1 Requirements

- JDK 7+
- Windows / Linux
- Eclipse
- Maven

Build 실행 순서는 아래와 같습니다.

1. 이클립스로 소스 import
2. Maven 을 이용해서 소스 컴파일
3. 설정 파일 추가 및 수정
4. sda-web 모듈 실행

##### 3.3.1.1 이클립스로 소스 import

- 메뉴에서 File/Import 메뉴를 선택하여 Import 창을 열어 Maven Project 를 선택한 후 소스가 저장된 폴더를 선택하여 import 합니다.

### 3.3.1.2 설정 파일 추가 및 수정

- Release 페이지에서 다운받은 설정파일(system.properties)을 수정 합니다.
- DB 설정 등 로컬환경에 맞게 설정파일을 수정합니다. 설정파일에 대한 자세한 내용은 [SDA 서버설정](#) 페이지를 참고하세요.

### 3.3.1.3 Maven 을 이용한 Build

- Alt+Shift+X,M 로 Maven Build 실행.
- Package 를 입력하고 build 수행

### 3.3.1.4 SDA 웹모듈 실행

- target 폴더(sda-web/target/sda-web-2.0.war)를 tomcat 의 webapps 에 sda.war 로 이름을 변경하여 복사하고 tomcat 를 기동해준다.

## 4. Module

SDA Framework 는 세 개의 서버 모듈로 구성됩니다.

### 4.1 sda-web : RESTFul API 및 web 서비스등을 제공하기 위한 web 모듈

1. sda-web/com.pineone.icbms.sda.itf : cm, ci에 대한 CRUD 클래스
  - 가. sda-web/com.pineone.icbms.sda.itf.ci : ci에 대한 CRUD
  - 나. sda-web/com.pineone.icbms.sda.itf.ci : cm에 대한 CRUD
  - 다. sda-web/com.pineone.icbms.sda.itf.cmi : cm/ci에 대한 CRUD(주로 같이 사용해야하는 경우)
2. sda-web/com.pineone.icbms.sda.logger : 공통적으로 적용되는 로그지정
3. sda-web/com.pineone.icbms.sda.sch : SI로 부터 데이터를 수집하거나 내부컨텍스트 스케줄 작업을 수행
  - 가. sda-web/com.pineone.icbms.sda.sch.comm : 스케줄러 공통 클래스
  - 나. sda-web/com.pineone.icbms.sda.sch.controller : 스케줄러 controller클래스
  - 다. sda-web/com.pineone.icbms.sda.sch.dao : 스케줄러 dao클래스
  - 라. sda-web/com.pineone.icbms.sda.sch.service : 스케줄러 service클래스
  - 마. sda-web/com.pineone.icbms.sda.sch.service.CollectDataFromSIJobService : SI로 부터 데이터 수집
  - 바. sda-web/com.pineone.icbms.sda.sch.service.기타클래스 : 내부컨텍스트 용 클래스
4. sda-web/com.pineone.icbms.sda.sf : 상황인지 RESTFul API제공을 위한 클래스
5. sda-web/com.pineone.icbms.sda.subscribe : SI쪽 subscribe API제공을 위한 클래스(현재 사용되지 않음)
6. sda-web/resources/config : spring framework용 설정파일(\*.xml)
7. sda-web/resources/mapper : mybatis용 쿼리파일(\*.xml)

### 4.2 sda-client : 독립적으로 실행되는 client 프로그램을 가지고 있는 모듈

1. sda-client/com.pineone.icbms.sda : kafka broker에 등록된 onem2m 데이터를 가져와서 triple data file로 만들고 fuseki의 s-put.sh를 이용하여 triple repository에 저장
2. sda-client/com.pineone.icbms.sda.hbase.onem2m : onem2m데이터를 수집하는 패키지인데 사용되지는 않음

3. sda-client/com.pineone.icbms.sda.kafka.onem2m.AvroOneM2MDataSubscribe : onem2m데이터를 kafka broker로 부터 가져와서 triple data file로 만들고 triple repository에 저장

## 4.3 sda-common : sda 의 core 모듈로 공통적인 기능과 core 기능을 포함하고 있는 모듈

1. sda-common/com.pineone.icbms.sda.comm : sda-web혹은 sda-client등에서 공통적으로 사용되는 dto, exception정의 클래스, kafka avro전송 클래스
  - 가. sda-common/com.pineone.icbms.sda.comm.kafka.onem2m : sda-client의 AvroOneM2MDataSubscribe.java파일이 사용하는 클래스
  - 나. sda-common/com.pineone.icbms.sda.comm.kb : onem2m 데이터를 triple data로 변환하는 모듈
  - 다. sda-common/com.pineone.icbms.sda.comm.sf : 상황인지를 위한 dao, dto, service등을 포함하는 모듈
2. sda-common/resources/mariadb : sda-common의 모듈에서 log및 context model정보등을 조회하기 위해서 사용하는 db커넥션 클래스및 정보
3. sda-common/resources/system.properties : 공통적으로 사용하는 설정값등을 가지고 있는 config파일

## 5. Q&A