


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

SO 프레임워크 오픈 소스 메뉴얼

요약 : SO 프레임워크를 적용하여 개발하는 방법을 설명한다.

 PINEONE	SO 프레임워크 개발가이드		
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

[illegible]

 PINEONE	SO 프레임워크 개발가이드		
프로젝트명:	단계:	담당팀:	

목차

1. 개요	5
2. 주요기능	5
3. 개발 방법	6
3.1 Quick Start	6
3.1.1 Requirements	6
3.1.2 따라하기	6
3.2 SO 서버 설정	8
3.3 SODA Framework Server Build	10
3.4 Requirements	10
4. Module	14
4.1 SODA Framework Server TEST	14
5. Q&A	16

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 지속적으로 성장해 나갈 계획입니다.

2. 주요기능

- IOT 환경에서 동적으로 변화하는 서비스와 서비스 구동 환경 그리고 이벤트 처리 가능
- 서비스 설명, 서비스 목표, 서비스 상세, 비즈니스 명세등의 표준 기반 아키텍처
- 다중 서비스를 하나의 독립적인 서비스로 활용
- 주어진 목표를 수행하기 위해 적절한 서비스를 선택, 구성, 협업 하기 위한 구성요소와 프로세스
- SDA 연동 기능 지원 , 상황 발생 조회 및 인지
- MongoDB 기반의 리소스 데이터 관리

3. 개발 방법

3.1 Quick Start

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

SO Server 시험은 아래의 순서로 진행할 수 있습니다.

1. MongoDB, JDK 다운로드 및 설치
2. Oasis SO FrameWork 소스 다운로드
3. mongoDB 기본 셋팅
4. SO Server 빌드
5. SO Server 설정 및 실행
6. HTTP 애물레이터(PostMan)를 이용한 시험

3.1.1 Requirements

- JDK 7+
- MongoDB 3.x
- Windows / Linux

3.1.2 따라하기

3.1.2.1 MongoDB, JDK 다운로드 및 설치

- [mongodb 설치안내](#)
- [JDK 설치안내](#)

3.1.2.2 Oasis SO Server 소스 다운로드

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

3.1.2.3 mongoDB 기본 셋팅

- [릴리즈 페이지](#)에서 다운받은 mongodb 스크립트파일(mongodb_script.txt)를 실행시켜서 기본 컬렉션 및 색인을 생성한다.

3.1.2.4 SO Server 빌드

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

3.1.2.5 SO Server 설정 및 실행

- 다운 받은 소스의 so-web/application.properties 파일을 오픈하여 SO 설정을 수정한다.
- SO Server 설정방법은 [SO 서버 설정방법](#) 페이지를 참고한다.

3.1.2.6 HTTP 애플레이터(PostMan)을 이용한 시험

- PostMan 프로그램을 설치한다. [PostMan 다운로드](#)
- PostMan 에서 URL 의 "localhost" 부분을 SO 서버를 실행한 서버의 IP 로 수정하여 메시지 전송을 시험한다. [Test 페이지](#)를 참조한다.

3.2 SO 서버 설정

SO 서버 설정방법을 설명합니다.

설정은 properties 파일로 작성되며 so-web 폴더의 resources 폴더에 존재합니다.

application.properties

설정항목별 의미는 아래와 같습니다.

Item	Description
spring.data.mongodb.database=so	데이터 베이스 이름
spring.data.mongodb.host	데이터베이스 호스트명
spring.data.mongodb.port	데이터베이스 포트번호
server.context-path	서버 main-path 설정
server.port	서버 포트 설정
spring.jackson.time-zone	서버 시간 설정 (Asia/Seoul 추천)

아래는 설정파일 샘플입니다.

```
spring.data.mongodb.database=so
spring.data.mongodb.host=101.010.101.01
spring.data.mongodb.port=10011
```


프로젝트명:

단계:

담당팀:

```
spring.data.mongodb.repositories.enabled=true  
server.context-path=/so  
server.port=10012  
spring.jackson.time-zone=Asia/Seoul
```

3.3 SODA Framework Server Build

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

Build 를 위한 요구조건은 아래와 같습니다.

3.4 Requirements

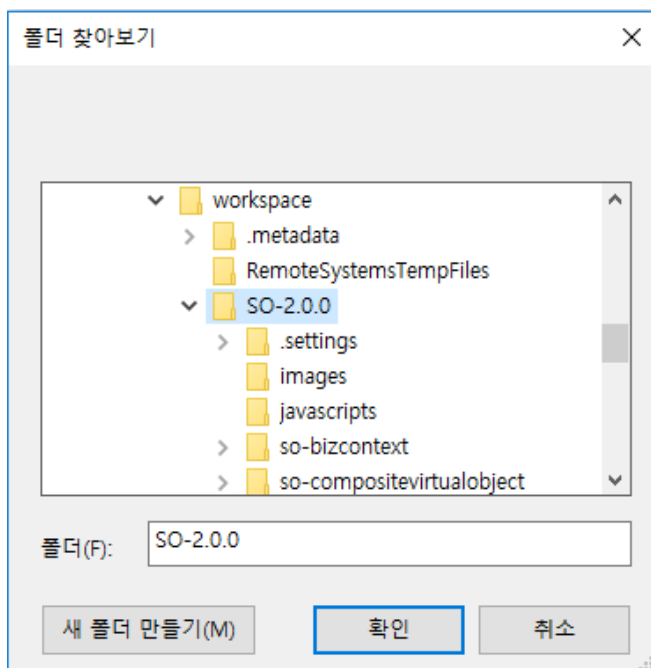
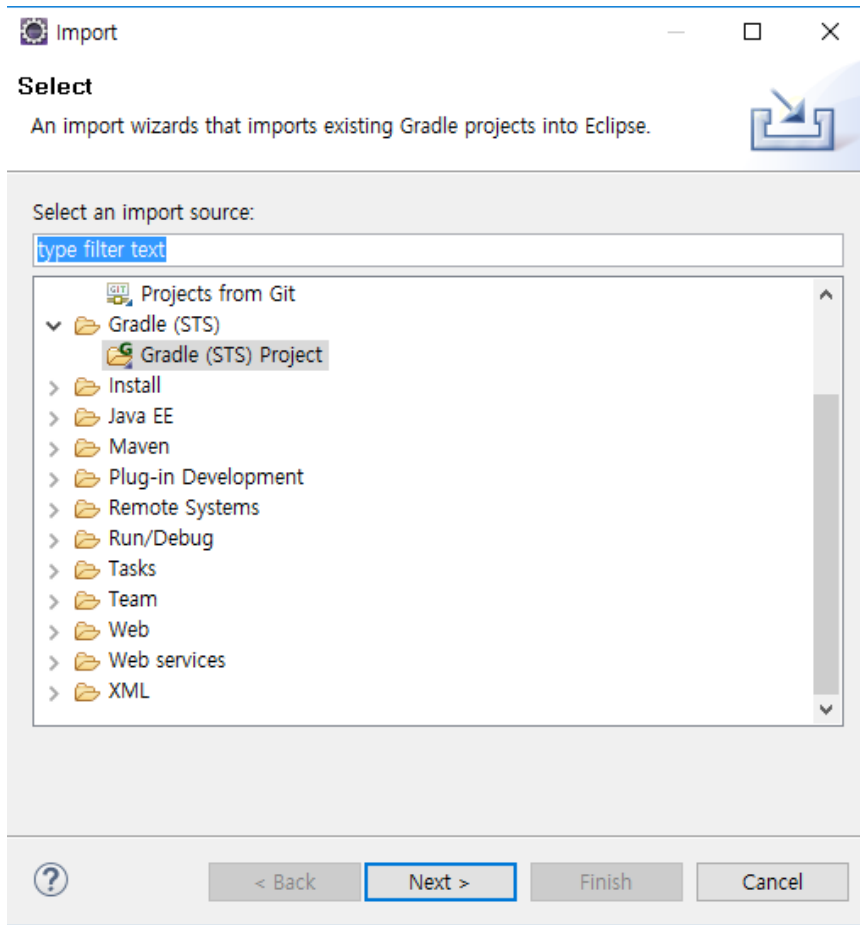
- JDK 7+
- Windows / Linux
- eclipse
- gradle

Build 실행 순서는 아래와 같습니다.

1. 이클립스로 소스 import
2. Gradle 을 이용해서 소스 컴파일
3. 설정 파일 추가 및 수정
4. SODA Framework 서버 실행
5. PostMan 을 활용한 SODA Framework 서버 시험

3.4.1.1 (1) 이클립스로 소스 import



- 메뉴에서 File/Import 메뉴를 선택하여 Import 창을 열어 Gradle(STS)/Gradle(STS) Project 를 선택한 후 소스가 저장된 폴더를 선택합니다.
- Import Gradle Project 화면에서 오른쪽 Build Model 버튼을 클릭하여 프로젝트의 dependency 및 lib 를 받는다.
- Import 된 프로젝트는 아래와 같이 표시됩니다.













프로젝트명:

단계:

담당팀:


Import Gradle Project


Root folder:

Project	Description
<input checked="" type="checkbox"/>  SO	
<input checked="" type="checkbox"/>  so-bizcontext	
<input checked="" type="checkbox"/>  so-compositevirtualobject	
<input checked="" type="checkbox"/>  so-contextinformation	
<input checked="" type="checkbox"/>  so-contextmodel	
<input checked="" type="checkbox"/>  so-creator	
<input checked="" type="checkbox"/>  so-device	
<input checked="" type="checkbox"/>  so-domain	
<input checked="" type="checkbox"/>  so-profile	
<input checked="" type="checkbox"/>  so-scheduler	

☒ Auto-select subprojects

Import options

☒ Run before
☒ Run after


☒ Enable dependency management
☒ Create resource filters

☐ Use hierarchical project names
☒ Add to workingset 'SO'

Additional working sets

☐ Add project to working sets

Working sets:

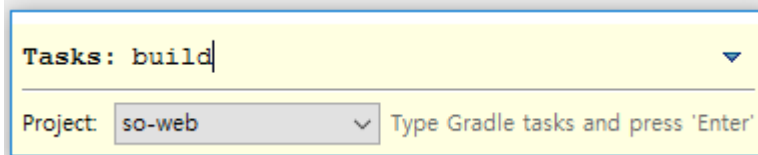


3.4.1.2 (2) 설정 파일 추가 및 수정

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

3.4.1.3 (3) Gradle 을 이용한 Build

- Ctrl+Alt+Shift+R 로 Gradle Build 실행.
- Project 를 so-web 선택. Tasks 에 build 명령어 입력.



3.4.1.4 (4) SO 서버 실행

- target 폴더(so-web/build/libs)에서 아래 명령으로 서버를 실행합니다.
- ```
java -jar service-orchestration-2.0.0-SANPSHOT.jar
```

## (5) PostMan 을 활용한 SO 서버 시험

- POSTMAN 을 활용하여 SO 서버를 시험합니다.
- POSTMAN 시험을 위한 스크립트 파일은 release 페이지에서 다운받을 수 있습니다.
- POSTMAN 을 활용한 SO 서버 시험에 관한 자세한 내용은 [Test 페이지](#)에서 확인할 수 있습니다.

## 4. Module

- SO Framework 는 하나의 서버 모듈로 구성됩니다.
- SO Framework 시험하기 위해 가상의 서비스를 이용하여 테스트 할 수 있습니다.

### 4.1 SODA Framework Server TEST

다운 받은 SODA Framework 서버를 테스트 하기 위한 방법을 설명 합니다.

TEST 실행 순서는 아래와 같습니다.

1. SO 서버 설정을 한다.
2. mongodb script 를 이용하여 TEST DB 생성 한다.
3. SO Service 실행 한다.
4. POSTMAN 으로 생성한 SO(Service Orchestration)를 실행한다.

#### 4.1.1.1 (1) SO 서버 설정

- SO 서버 설정 페이지에서 설정한다.
- ex)

```
spring.data.mongodb.database=sotest
spring.data.mongodb.host=127.0.0.1
spring.data.mongodb.port=27017
spring.data.mongodb.repositories.enabled=true
server.context-path=/so
server.port=10080
spring.jackson.time-zone=Asia/Seoul
```

## 4.1.1.2 (2) mongodb 생성

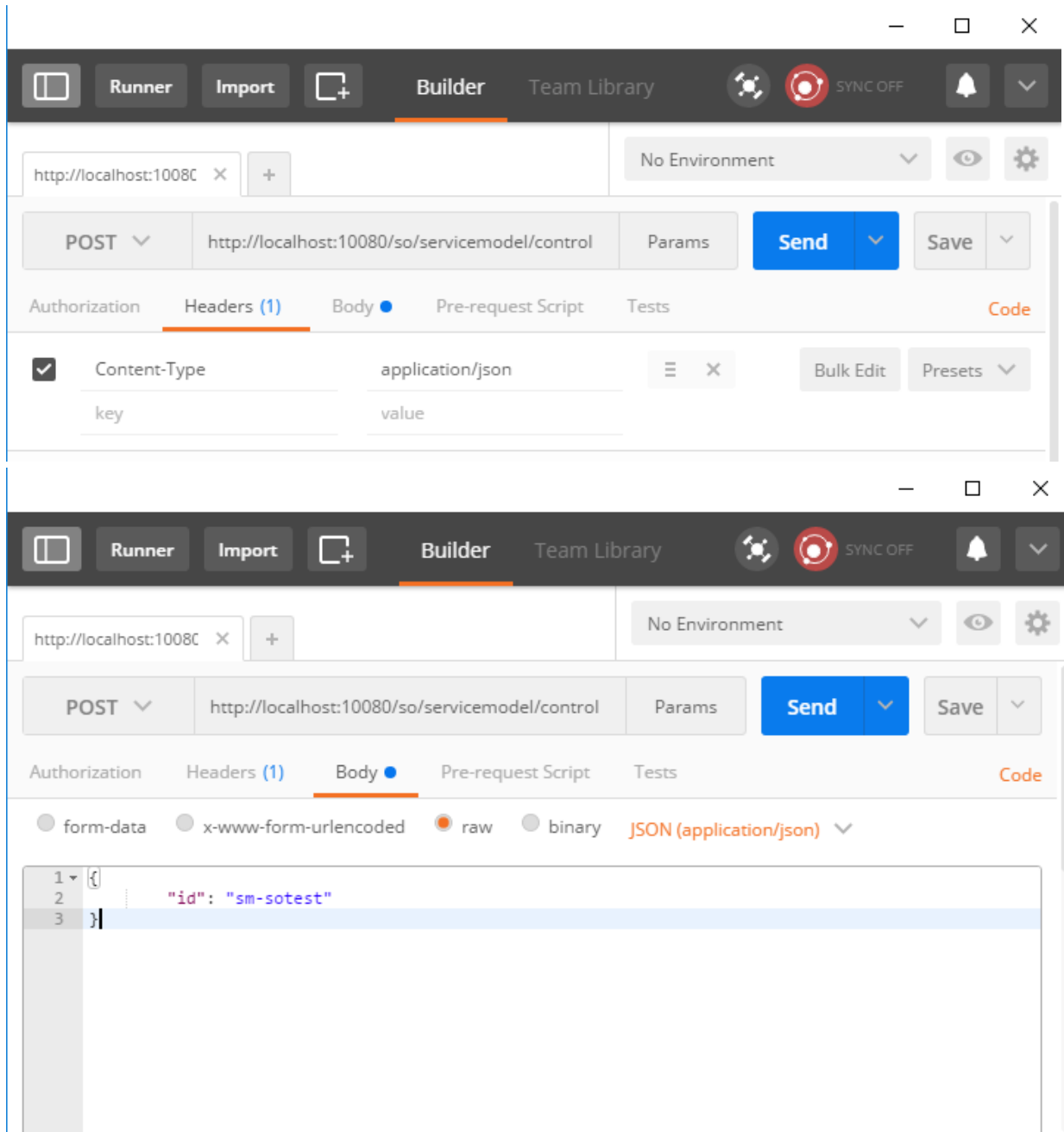
- Release 페이지에서 다운 받은 소스에서 [MongoDB Script](#)를 복사 하여 MongoDB Shell 에 실행 시킨다.
- 생성된 DB 를 확인 한다.

## 4.1.1.3 (3) SO Server 실행

- target 폴더(so-web/build/libs)에서 아래 명령으로 서버를 실행합니다.
- ```
java -jar service-orchestration-2.0.0-SANPSHOT.jar
```

4.1.1.4 (4) POSTMAN 으로 생성한 SO(Service Orchestration)를 실행

- Headers 에 **Content-Type** 는 **application/json** 으로 설정
- POST 에 **./so/servicemodel/control** 입력
- Body 에 { **"id": "sm-sotest"** } 입력
- Send 버튼 클릭
- SO Server 가 실행 되는 것을 확인 한다.



5. Q&A