
01

개요



개요

가. 개요

1) 목적 :

가) 무선 IoT 단말기의 데이터 수신 처리 서버 구현

2) 개발 목표 :

가) 데이터 센싱과 무선 송수신을 이용한 IoT 플랫폼 서비스의 수신 서버 구현

3) 개발 환경

가) IDE : Eclipse

나) DBMS : MariaDB

(1) 구축 서버 : Raspberry pi 4, 4GB

다) IoT Server Application

(1) Framework :

(가) Netty

(나) Spring Framework

(다) MyBatis

개요

4) 개발 환경 상세

가) IDE : Eclipse

- (1) 버전 정보 : Version: Oxygen.3a Release (4.7.3a)

나) DBMS : MariaDB

- (1) 구축 서버 :

- (가) 하드웨어 : Raspberry pi 4, RAM 4GB, SD 32GB

- (나) OS : Raspbian GNU/Linux 10

- (2) 버전 정보 : 10.3.22-MariaDB-0+deb10u1

- (3) 관련 라이브러리 :

- a. mysql-connector-java 8.0.19

다) IoT Server Application

- (1) Framework :

- (가) Netty

- (1) 버전 정보 : 4.0.31.Final

- (나) Spring Framework

- (1) 버전 정보 : 5.2.5.RELEASE

- (2) 관련 라이브러리 :

- a. spring-context 5.2.5

- b. spring-jdbc 5.2.5

- c. spring-test 5.2.5

- (다) MyBatis

- (1) 버전 정보 : 3.5.4

- (2) 관련 라이브러리 :

- a. mybatis-spring 2.0.4

- b. spring-tx 3.5.4

라) 기타

- (1) 엑셀처리 : poi-ooxml 4.1.2

- (2) lombok : lombok 1.18.12

- (3) logging : log4j 1.2.17

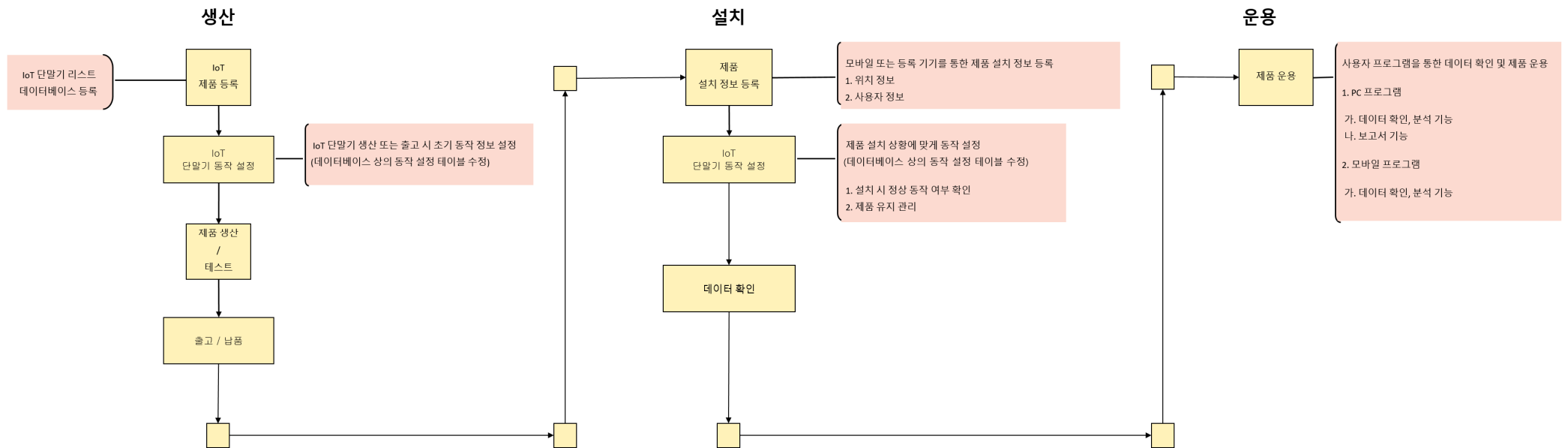
- (4) test : junit 4.13

- (5) java 1.8

- (6) maven 프로젝트

개요

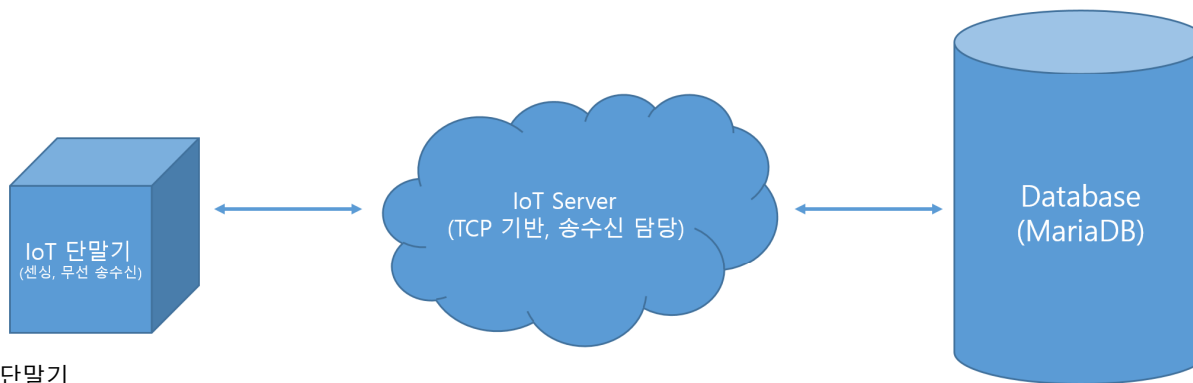
5) 일반사항 : IoT 서비스 시스템



개요

6) IoT 서버 어플리케이션 시스템 구성

시스템 구성도



※ IoT 단말기

- 통신 모듈이 장착된 펌웨어 기반 단말기
- Java Application으로 대체

02

일정

일정

1) 개발 일정

프로젝트명	IoT 서버 어플리케이션 구축
시작일정	2020-04-15
종료일정	2020-05-23
기간	39

업무	세부 업무	기간	시작일	종료일	상태	4/15	4/16	4/17	4/18	4/19	4/20	4/21	4/22	4/23	4/24	4/25	4/26	4/27	4/28	4/29	4/30	5/1	5/2	5/3	5/4	5/5	5/6	5/7	5/8	5/9	5/10	5/11	5/12	5/13	5/14	5/15	5/16	5/17	5/18	5/19	5/20	5/21	5/22	5/23	5/24																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																														

03

DB 설계

DB 설계

customer_info						
Field	Type	Collation	Null	Key	Default	Extra
id	varchar(50)	utf8_general_ci	NO	PRI		
password	varchar(50)	utf8_general_ci	NO			
name	varchar(50)	utf8_general_ci	NO			
email	varchar(50)	utf8_general_ci	NO			
phone_number	varchar(50)	utf8_general_ci	NO			
company_name	varchar(50)	utf8_general_ci	YES			
insert_date	datetime		YES		current_timestamp()	
update_date	datetime		YES		current_timestamp()	

device_list						
Field	Type	Collation	Null	Key	Default	Extra
serial_number	varchar(11)	utf8_general_ci	NO	PRI		
sw_version	varchar(13)	utf8_general_ci	NO			
hw_version	varchar(13)	utf8_general_ci	NO			
prod_histroy	varchar(50)	utf8_general_ci	YES			
note	varchar(50)	utf8_general_ci	YES			
customer_id	varchar(50)	utf8_general_ci	YES	MUL		
insert_date	datetime		YES		current_timestamp()	
update_date	datetime		YES		current_timestamp()	

1. 테이블 : customer_info

2. 목적 : 제품 구매 정보 관리

3. 사용 :

- 가. 생산 시 가장 먼저 리스트업 진행
- 나. 구매자 정보 미 존재 시 자체 계정 생성하여 관리

1. 테이블 : device_list

2. 목적 : 제품 정보 저장, 생산 정보 추적 관리

3. 사용 :

- 가. customer_id 테이블에서 id 필드를 외래키로 사용
- 나. 어플리케이션에서 사용 가능한 제품(시리얼번호의 정상) 여부 판단에 사용

DB 설계

installation_info						
Field	Type	Collation	Null	Key	Default	Extra
uid	bigint(20)		NO	PRI		
serial_number	varchar(11)	utf8_general_ci	YES	UNI		
inst_latitude	varchar(15)	utf8_general_ci	YES			
inst_longitude	varchar(15)	utf8_general_ci	YES			
insert_date	datetime		YES		current_timestamp()	
update_date	datetime		YES		current_timestamp()	

user_info						
Field	Type	Collation	Null	Key	Default	Extra
idx	bigint(20)		NO	PRI		auto_increment
owner	varchar(50)	utf8_general_ci	YES			
nt_address	varchar(50)	utf8_general_ci	YES			
ot_address	varchar(50)	utf8_general_ci	YES			
note	varchar(50)	utf8_general_ci	YES			
mobile	varchar(50)	utf8_general_ci	YES			
uid	bigint(20)		NO	MUL		
insert_date	datetime		YES		current_timestamp()	
update_date	datetime		YES		current_timestamp()	

1. 테이블 : installation_info

2. 목적 : 제품 설치 위치 정보 저장

3. 사용 :

- 가. 제품 설치 시 모바일 또는 휴대용 단말기 이용
- 나. 설치된 위치 정보 저장

1. 테이블 : user_info

2. 목적 : 설치 위치 관련 실 사용자 정보 저장 및 관리

3. 사용 :

- 가. 제품 설치 위치에 따른 사용자는 둘 이상 존재
- 나. installation_info 테이블에서 uid 필드를 외래키로 사용
- 다. installation_info의 uid 기준 사용자 정보 등록
- 라. 모바일 또는 휴대용 단말기, PC 프로그램 등을 이용, 사용자 정보 등록

DB 설계

communication_info						
Field	Type	Collation	Null	Key	Default	Extra
idx	bigint(20)		NO	PRI		auto_increment
serial_number	varchar(11)	utf8_general_ci	NO			
uid	bigint(20)		NO			
ptl_version	varchar(13)	utf8_general_ci	NO			
header_data	longblob		YES			
dev_latitude	varchar(15)	utf8_general_ci	YES			
dev_longitude	varchar(15)	utf8_general_ci	YES			
error	varchar(13)	utf8_general_ci	YES			
insert_date	datetime		YES		current_timestamp()	

1. 테이블 : communication_info

2. 목적 : 무선 통신 정보 저장, 제품 버전 정보 및 동작 상태 관리

3. 사용 :

가. 제품 동작 정보 및 관리에 따른 정보 사용

나. 동작 상태 관련 정보 저장(헤더 정보)

device_data						
Field	Type	Collation	Null	Key	Default	Extra
idx	bigint(20)		NO	PRI		auto_increment
serial_number	varchar(11)	utf8_general_ci	NO			
uid	bigint(20)		NO			
time	varchar(6)	utf8_general_ci	NO			
date	varchar(6)	utf8_general_ci	NO			
value	longblob		YES			
insert_date	datetime		YES		current_timestamp()	
update_date	datetime		YES		current_timestamp()	

1. 테이블 : device_data

2. 목적 : IoT 제품 센싱 데이터 저장

3. 사용 :

가. 사용자에게 필요한 센싱 데이터 사용

나. 프로그램을 이용, 사용자에게 데이터 제공

DB 설계

set_info						
Field	Type	Collation	Null	Key	Default	Extra
serial_number	varchar(11)	utf8_general_ci	NO	PRI		
opt1	varchar(10)	utf8_general_ci	YES			
opt2	varchar(10)	utf8_general_ci	YES			
opt3	varchar(10)	utf8_general_ci	YES			
opt4	varchar(10)	utf8_general_ci	YES			
opt5	varchar(10)	utf8_general_ci	YES			
insert_date	datetime		YES		current_timestamp()	
update_date	datetime		YES		current_timestamp()	

1. 테이블 : set_info

2. 목적 : IoT 단말기(제품) 동작 정보 및 원격 컨트롤을 위한 정보 저장

3. 사용 :

가. 제품 동작 상태 원격 설정

나. 제품 생산 및 설치 시 관리자 설정 필요

04

통신 프로토콜 설계

통신 프로토콜 설계

1) IoT 프로토콜

가) 데이터 프레임

(1) 구성 : 크게 세 부분으로 구성

(가) 헤더, 페이로드, 트레일러

PROTOCOL		
HEADER	PAYLOAD	TRAILER
1. 목적 : 프레임 해석 정보 제공 2. 크기 : 41 bytes 3. 특징 : 가. 고정 길이 나. 11개의 의미를 갖는 섹터로 구분	1. 목적 : IoT 단말기의 센싱 데이터 부분 2. 크기 : 가변 3. 특징 : 가. 가변 길이 나. 데이터에 따른 개별 프로토콜 존재	1. 목적 : 패킷의 에러 또는 제품 에러 정보 제공 2. 크기 : 3 bytes 3. 특징 : 가. 고정 길이 나. 3개의 의미를 갖는 섹터로 구분

통신 프로토콜 설계

(나) 헤더 프레임 상세

HEADER FRAME										
flag	protocol version	software version	hardware version	unique id	source address	destination address	header length	payload length	trailer length	reserved
3	3	3	3	4	4	4	4	4	4	5

1. flag : 패킷의 목적을 구분하는 용도로 사용

flag	의미	방향성
TST	테스트	단말기 -> IoT Server
RSP	정상 응답	단말기 <- IoT Server
ERR	에러 응답	단말기 <- IoT Server

* 페이로드 별 임의 정의

2. unique id : IoT 단말기에 부여된 유일한 시리얼번호(IMEI)

3. header length : 헤더 길이 정보 (41)

4. payload length : 페이로드 길이 정보 (가변)

5. trailer length : 트레일러 길이 정보 (3)

6. reserved : 예비 정보 구간

통신 프로토콜 설계

(다) 트레일러 프레임 상세

TRAILER FRAME

error 01	error 02	error 03
3	3	3

1. 바이트별 에러 정보를 표시

2. 바이트당 256가지 에러 표시

3. 에러 발생 시 3번의 재시도 이력 저장

4. 에러 값 테이블 이용

ex) 에러 값 테이블

ERROR 값	의미
0	센싱 실패
1	TCP 연결 실패
2	기지국 연결 실패
3	서버 응답 에러
.	.
.	.
.	.

(라) 페이로드 프레임 상세

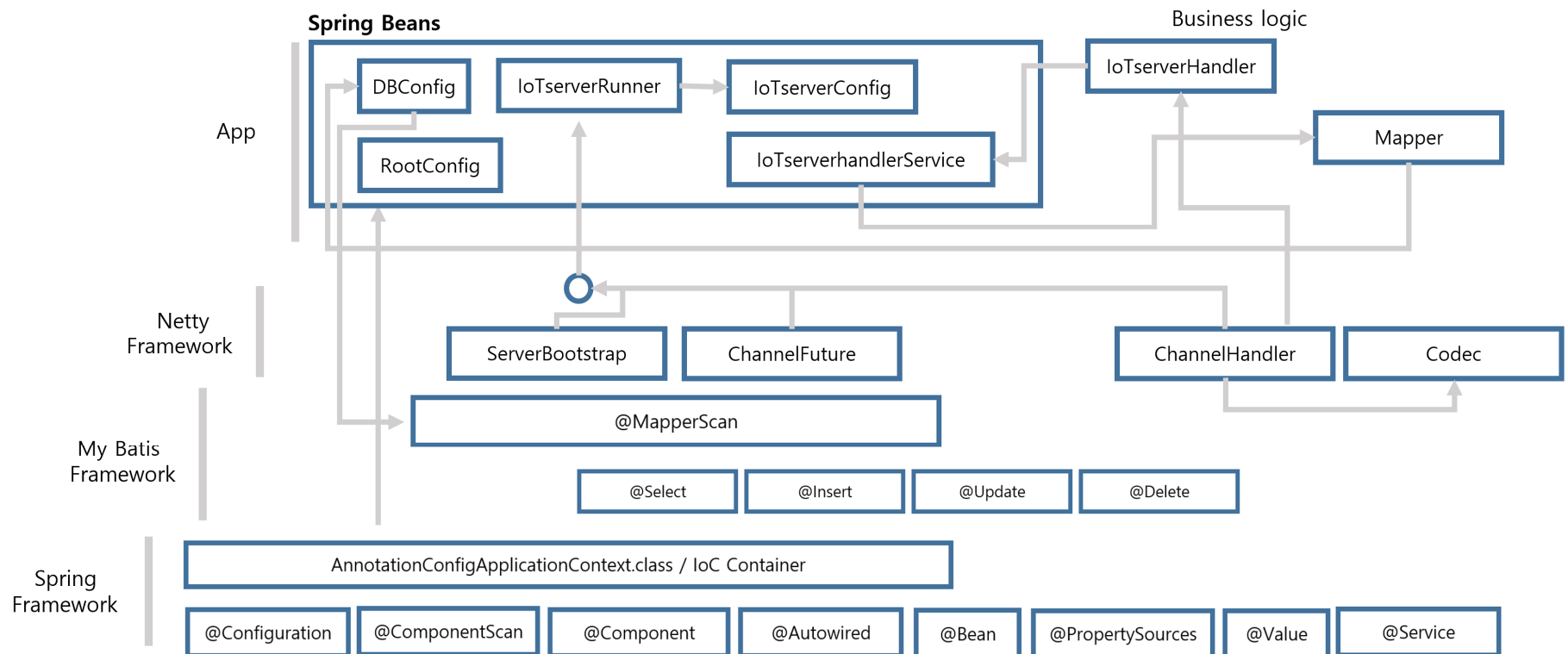
PAYLOAD FRAME(flag = RSP)				
OPT 01	OPT 02	OPT 03	OPT 04	OPT 05
4	4	4	4	4
1. flag 가 "RSP"인 경우의 페이로드 예시 2. IoT 단말기의 설정 정보 3. 데이터베이스의 set_info 테이블 참조				

05

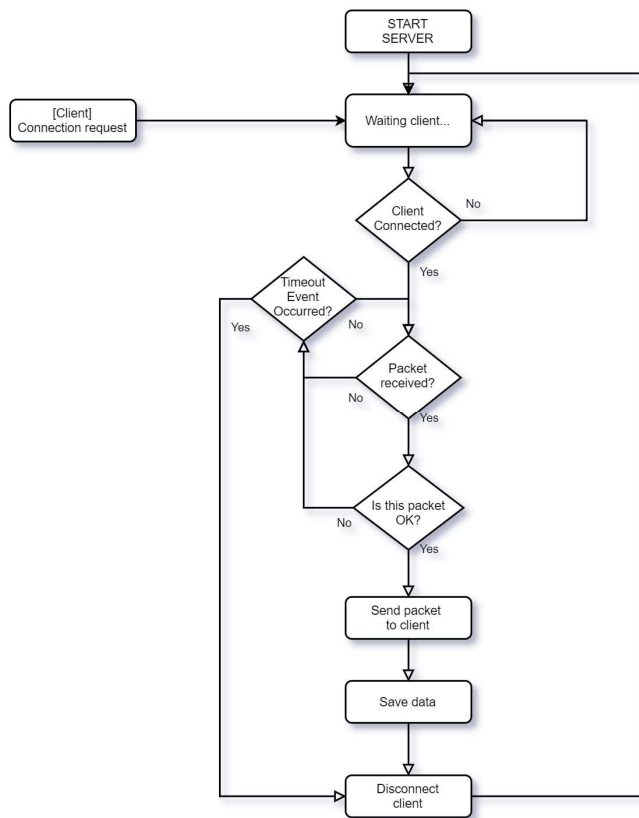
프로그램 아키텍처

프로그램 아키텍처

1) 프로그램 아키텍처



프로그램 아키텍처



06

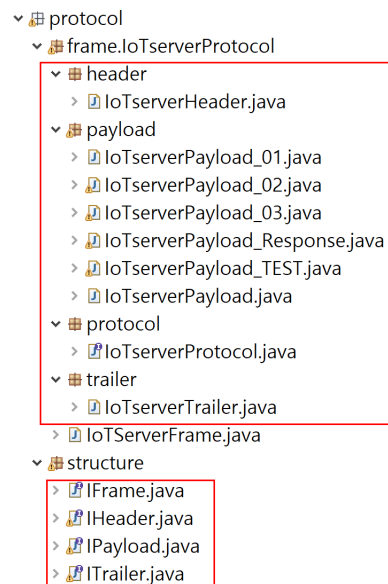
코드 리뷰

코드 리뷰

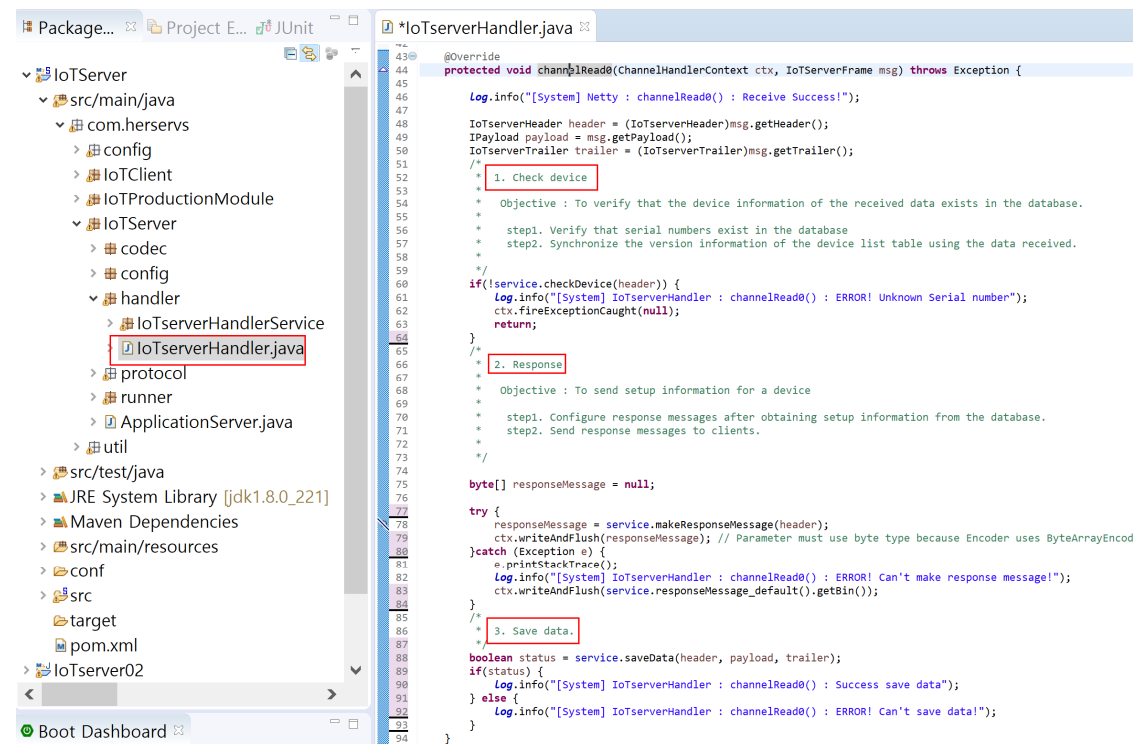
3) 코드 플랜

가) 프로토콜 다형성 고려

(1) 인터페이스를 통한 구현



나) 비즈니스 로직은 하나의 이벤트 처리 함수에서 모두 처리



코드 리뷰

다) 어플리케이션 동작 설정 값은 외부 파일을 이용

- (1) 서버 설정
- (2) DBMS 설정

