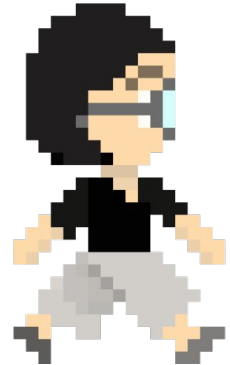


# HomeStatus

**Wisdom King**



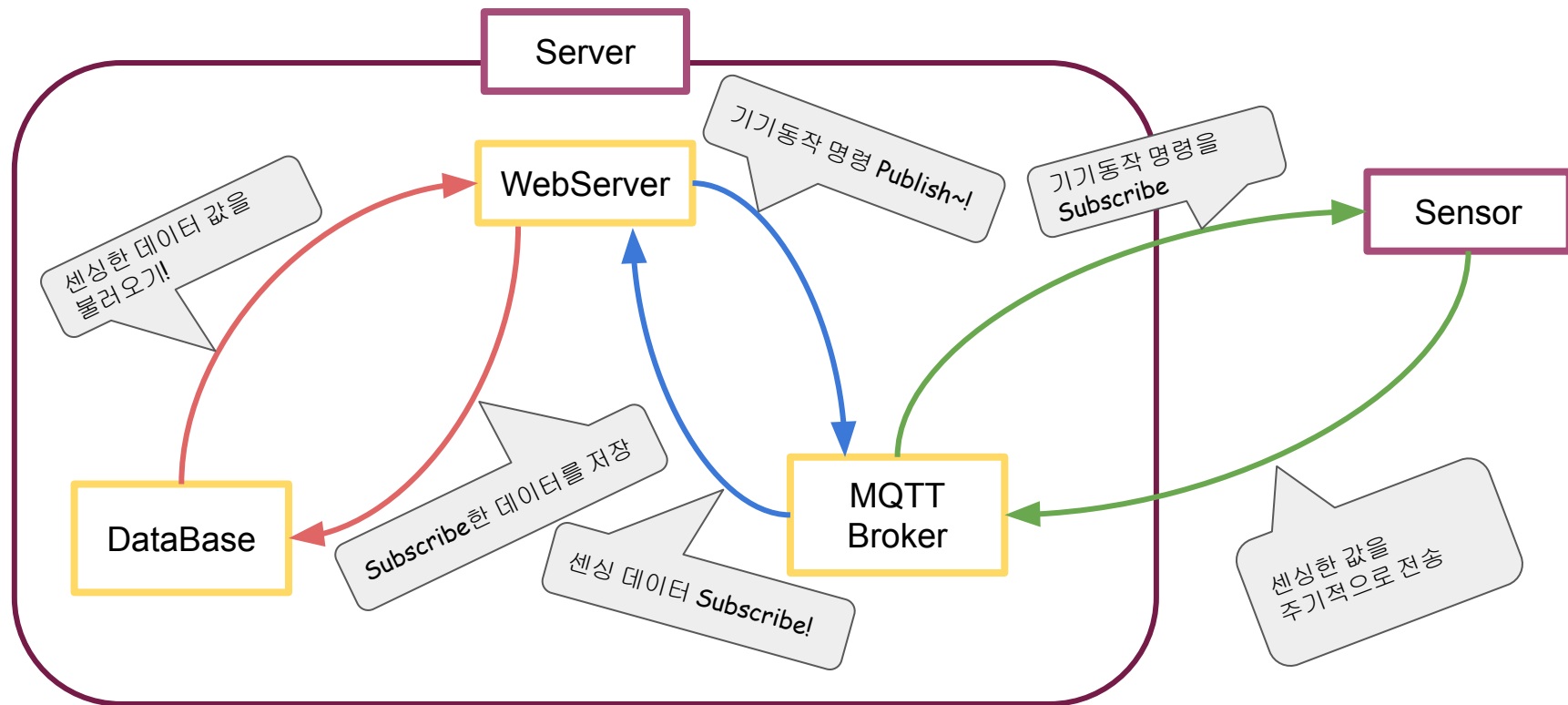
# HomeStatus 목적 및 기능

이전 학원과 직장에서 사용한 MQTT의 복습 및 다른 프레임워크도 익혀보기  
간단한 프로젝트.

기능의 MQTT를 사용하기 위한 기능들로 구성했습니다.

- 센싱한 온도, 습도, 먼지 값을 서버에 저장.
- 웹페이지에서 최근 온습도, 먼지 수치확인.
- 웹페이지에서 기기 ON/OFF.

# 프로젝트 구조~!





**SERVER**

# Server - 사용 툴

Front end framework:  
bootstrap

모바일로 규격에 최적화 되도록 구성

Web framework: express

- 간단하고 기본적인 프레임워크로 사용
- MVC 패턴과 유사하게 구성!

View 엔진: ejs

html 형식을 그대로 사용 가능한 이점!

라즈베리파이 OS는 Raspbian 경우  
프로그램들이 최신으로 지원을  
안하는 경우가 있어 Ubuntu  
Server를 사용하게 되었습니다.

Web Application : Node.js

빠르고 쉬운 구성을 위해 사용

DB : MongoDB

NoSQL로 구성이 쉽고 가볍고  
Session store로도 이용가능

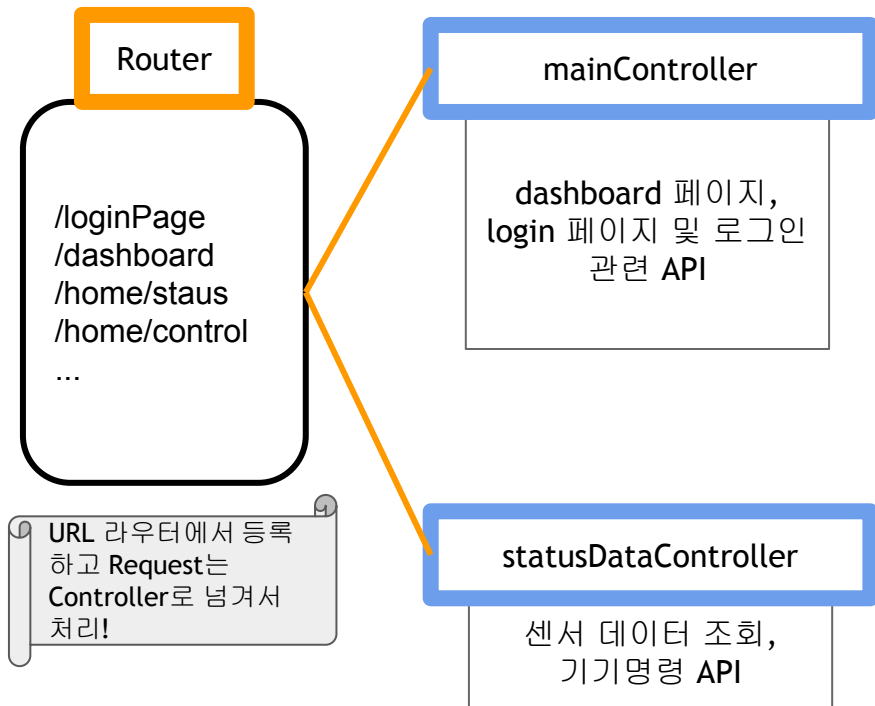
MQTT Broker  
: Mosquitto

운영 기기: 라즈베리 파이  
OS: Ubuntu Server

# WebServer!

Node.js는 로 express 프레임 워크를 이용  
MVC 패턴과 유사하게 만들었습니다.

## Controller



- View는 ejs엔진을 이용하여 html형식 사용!
- 로그인 관련 API는 JSON데이터 Response!
- 센서데이터 JSONArray로 return!
- 기기동작명령 API는 Request받은 명령을 해당 토픽으로 Publish!

# DataBase

- 사용DB: MongoDB
- Database명 : home
- Collection명: status

Node.js연동위해 mongoose  
라이브러리 이용  
Web에서 Session에서도  
이용.

MongDB는 NoSQL로 간단히 데이터 타입정도만 설정  
했습니다.

StatusData Key!

Key	Type	설명
temp	Number	온도값
humi	Number	습도값
dust	Number	먼지량
time	Date	저장시간

# MQTT 통신!

- MQTT Broker : Mosquitto
- Node.js 연동 라이브러리: mqtt.js
- QoS: publish, subscribe 모두 1

- 장치로 부터 주기적으로 수신 Topic 으로 JSON형태의 데이터를 받아 저장합니다.
- 명령은 설정되어있는 ID로 ON/OFF를 신호를 받아 작동합니다..

센서 데이터 수신	Topic	home/status
	데이터 형식	JSON 형식 String
기기명령	Topic	home/control/{동작기기ID}
	데이터 형식	char

ex) { "temp": 23, "humi": 34.2 , "dust": 23.2 }

ex) ID '1'로 설정된 기기를 ON 하려할때  
ON : 1 , OFF : 0  
topic : home/control/1  
data: 1