IBM Watson

Do It Yourself
인공지능 스피커

최의신
choies@kr.ibm.com

IBM

IBM Watson

Agenda

- 인공지능 스피커
- MQTT
- Watson Assistant (실습)
- 인공지능 스피커 만들기
- 인공지능 스피커 소스코드
- 확장하기

IBM

for WATSON

인공지능 스피커

인공지능 스피커

아마존 에코

구글 홈

SK 누구

네이버 웨이브

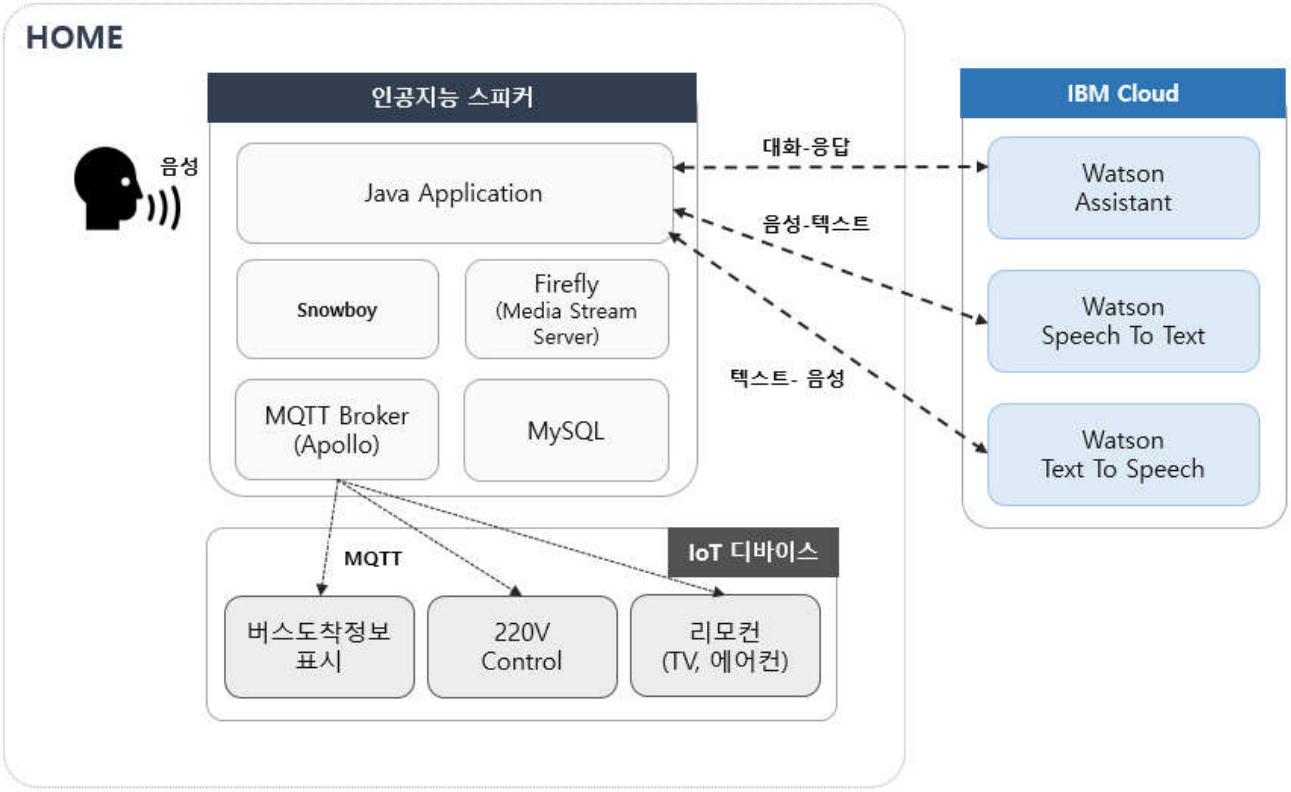
카카오 미니

기가 지니

Clova 프랜즈

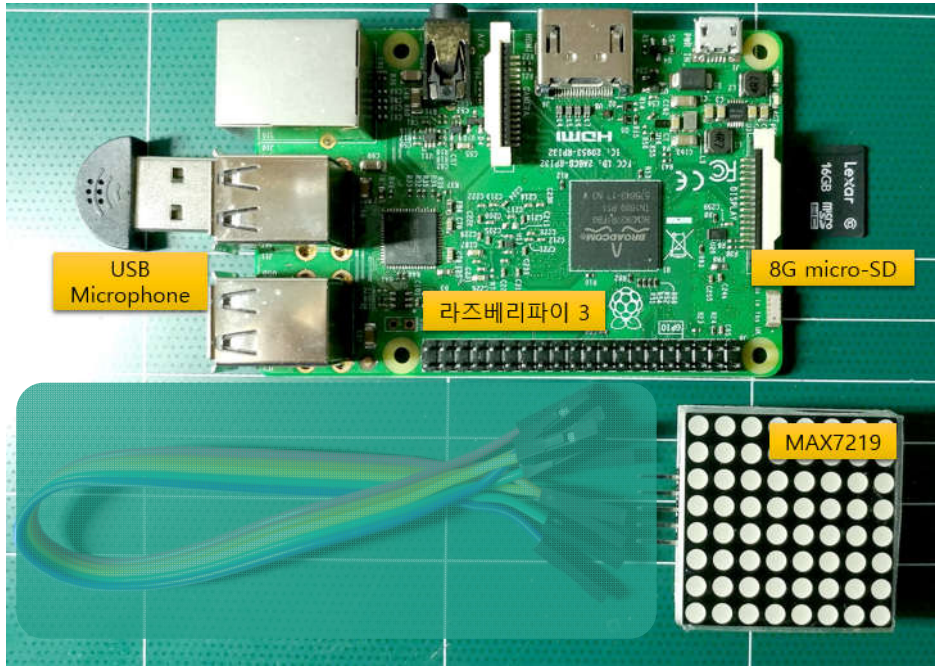
인공지능 스피커

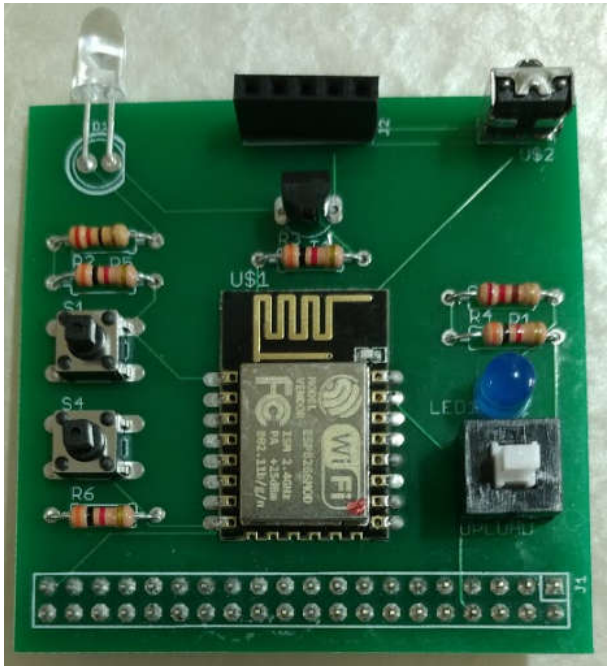
- 시스템 구성도



인공지능 스피커

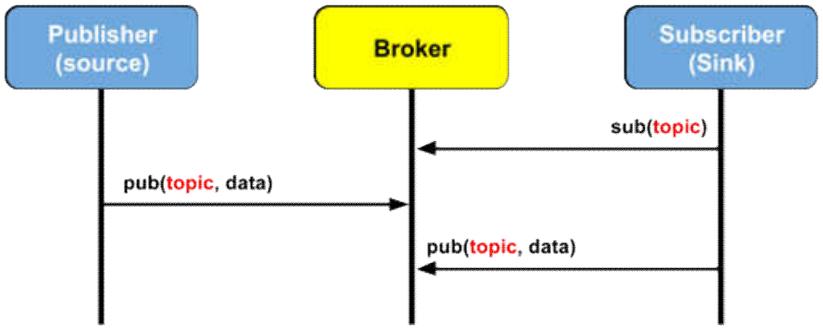
- 준비물





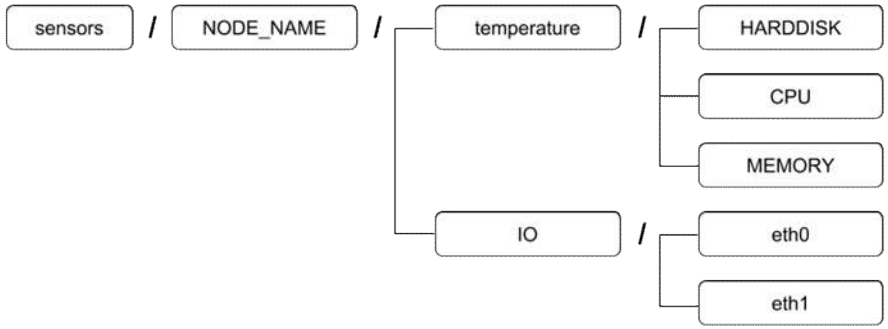
MQTT (Message Queue Telemetry Transport

- IBM 개발한 프로토콜
- 설계 의도
  - 느리고 품질이 낮은 네트워크의 장애와 단절에 대비
  - 프로토콜이 차지하는 리소스 점유(footprint)를 최소화
  - 클라이언트 애플리케이션 동작에 자원 활용이 극히 제한적임을 고려
- 특징
  - Publish/Subscribe



MQTT (Message Queue Telemetry Transport

- MQTT 특징
  - Topic



  - QoS (Quality of Service) 레벨
    - 0 : 메시지는 한번만 전달하며, 전달여부를 확인하지 않는다.
    - 1 : 메시지는 반드시 한번 이상 전달된다. 하지만 메시지의 핸드셰이킹 과정을 엄밀하게 추적하지 않기 때문에, 중복 전송 될 수도 있다.
    - 2 : 메시지는 한번만 전달된다. 메시지의 핸드셰이킹 과정을 추적한다.

MQTT (Message Queue Telemetry Transport

- MQTT 특징
  - 오버헤드를 최소화
    - 가장 작은 메시지 사이즈는 2byte (가변길이 MQTT헤더 + 애플리케이션 Payload)
    - Payload 데이터에 중립적 (애플리케이션 헤더 불필요)
  - TLS/SSL은 지원. X.509 인증서를 이용한 양방향 인증도 지원
  - Facebook에서 사용
  - 경량의 프로토콜로 M2M(machine-to-machine)와 IoT 환경에 적합

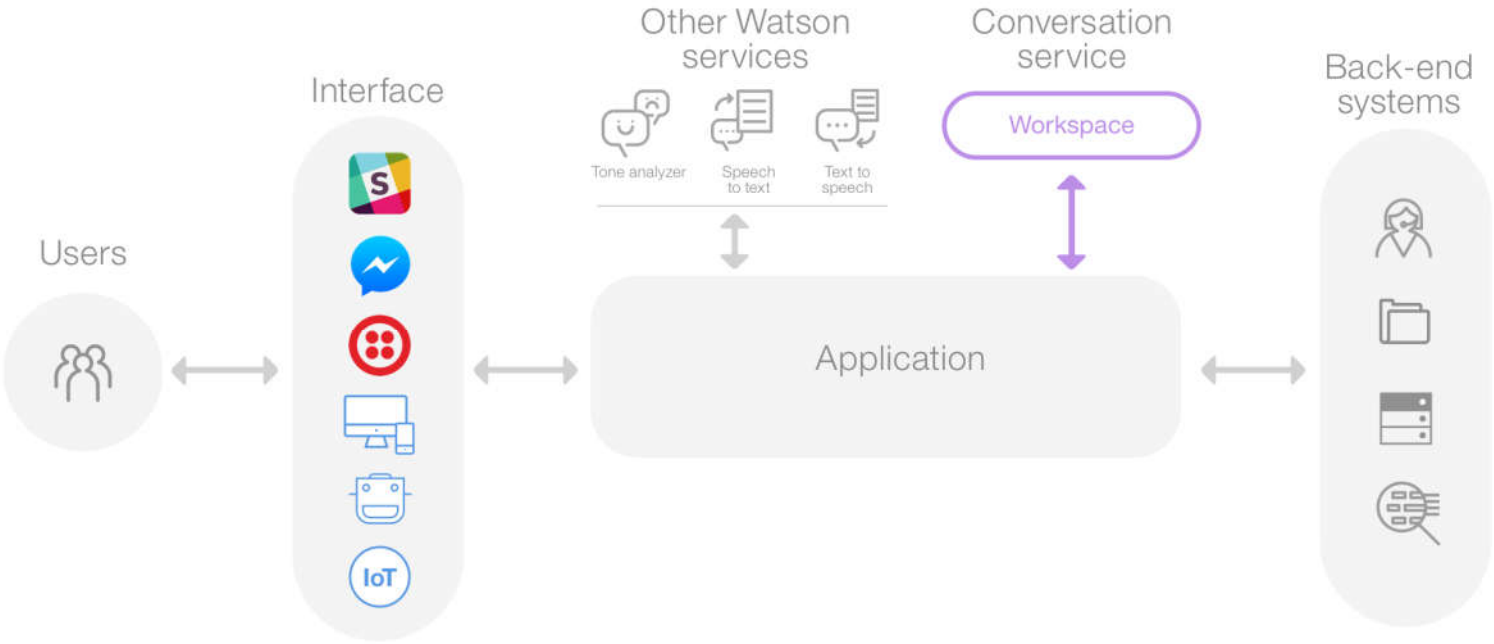
경량의 Publish/Subscribe(Pub/Sub) 기반의 메시징 프로토콜

Watson

ASSISTANT SERVICE

Watson Assistant

- 자연 언어 입력을 이해하고 고객에게 응답(대화)하는 응용 프로그램을 작성할 수 있다.



Watson Assistant

IBM

- Intent
  - 의도, 의향, 목적
  - 사용자의 입력이 무엇을 의미하는지 정의
  - 사용자의 입력에 대한 올바른 의도를 파악(선택)하는 것이 중요
  - 고객은 여러가지 방법으로 동일한 종류의 요청을 사용
    예)
    커피 가격이 얼마죠?
    커피 가격은?
    커피 가격 좀…
    얼마죠?

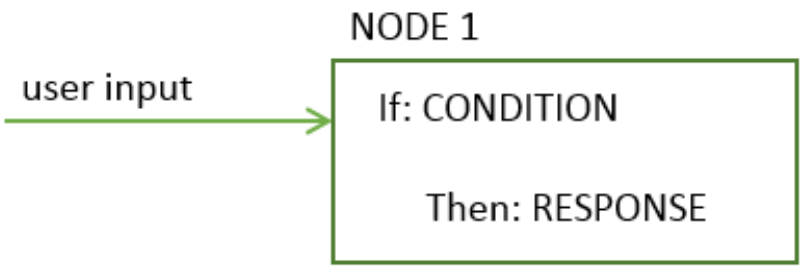
Watson Assistant

IBM

- Entity
  - 실체, 독립체
  - 사용자 입력 중에서 용어, 개체 또는 목적의 대상을 나타낸다.
  - 단일 의도에서 엔티티를 통해 여러 작업을 수행할 수 있다.
    예)
    - 전등을 꺼 주세요
    - TV를 꺼 주세요
    - '끄기' 라는 의도에서 대상이 되는 전등, TV를 구별한다.
  - 동의어를 지정할 수 있다. 예: TV = 텔레비젼

Watson Assistant

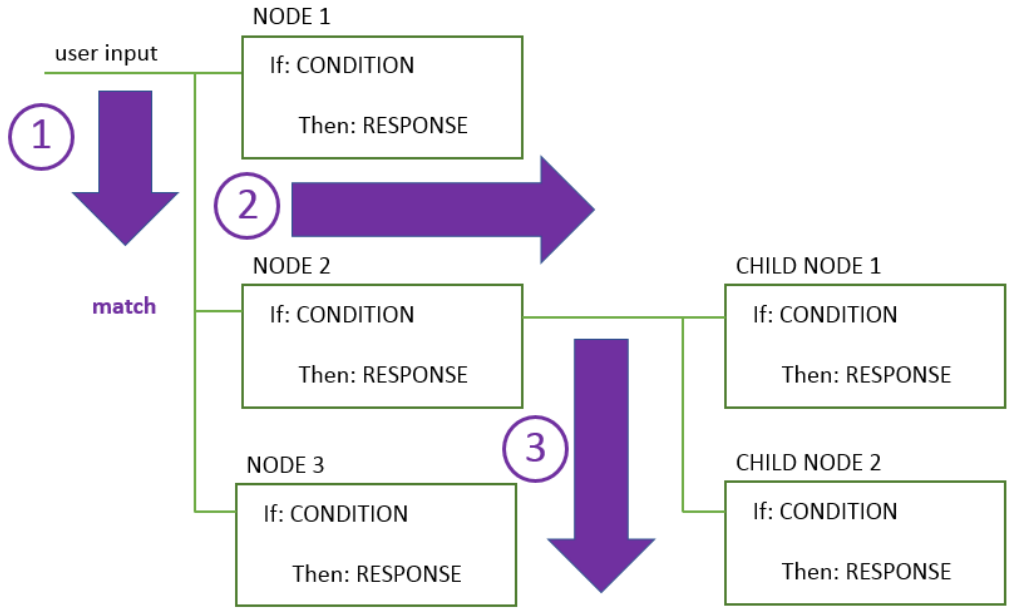
- Dialog
  - 대화 상자는 도구에서 그래픽으로 노드 트리를 구성한다.
  - 각 대화 상자 노드는 최소한 조건과 응답을 포함한다.



  - 노드를 if / then 구조로 생각할 수 있습니다. 조건이 참이면 응답을 반환합니다.

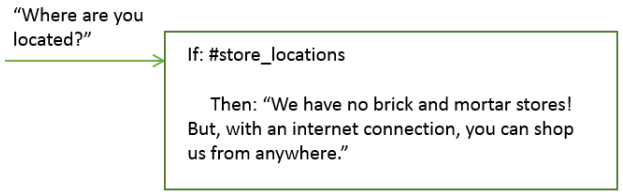
Watson Assistant

- Dialog
  - 서비스는 트리에서 위에서 아래로, 왼쪽에서 오른쪽으로 이어지는 분기의 마지막 노드에 도달 할 때까지 계속 작동합니다.

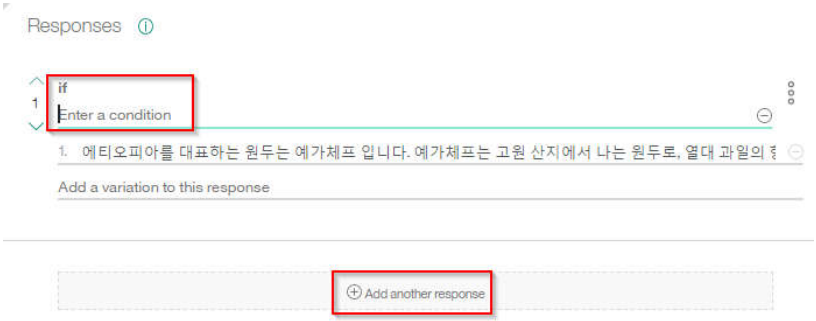


Watson Assistant

- Dialog – Response
  - 간단한 텍스트 응답



  - 복잡한 응답
    - 응답을 JSON 객체로 응답을 생성한다.
  - 여러 개의 조건 응답



Watson Assistant

IBM

기본 실습

- 목표
  - Intent 이해
  - Entity 이해
  - Dialog 이해
- 시나리오
  - 커피 문의
- 참조 문서
  - https://goo.gl/Xbk1LP, https://github.com/TechOrgg/ai_speaker4_watson.git (Conversation 기본 실습v1.1.pdf)
- 테스트
  - 케냐커피 맛나요?
  - 가격은 얼마죠?

Watson Assistant

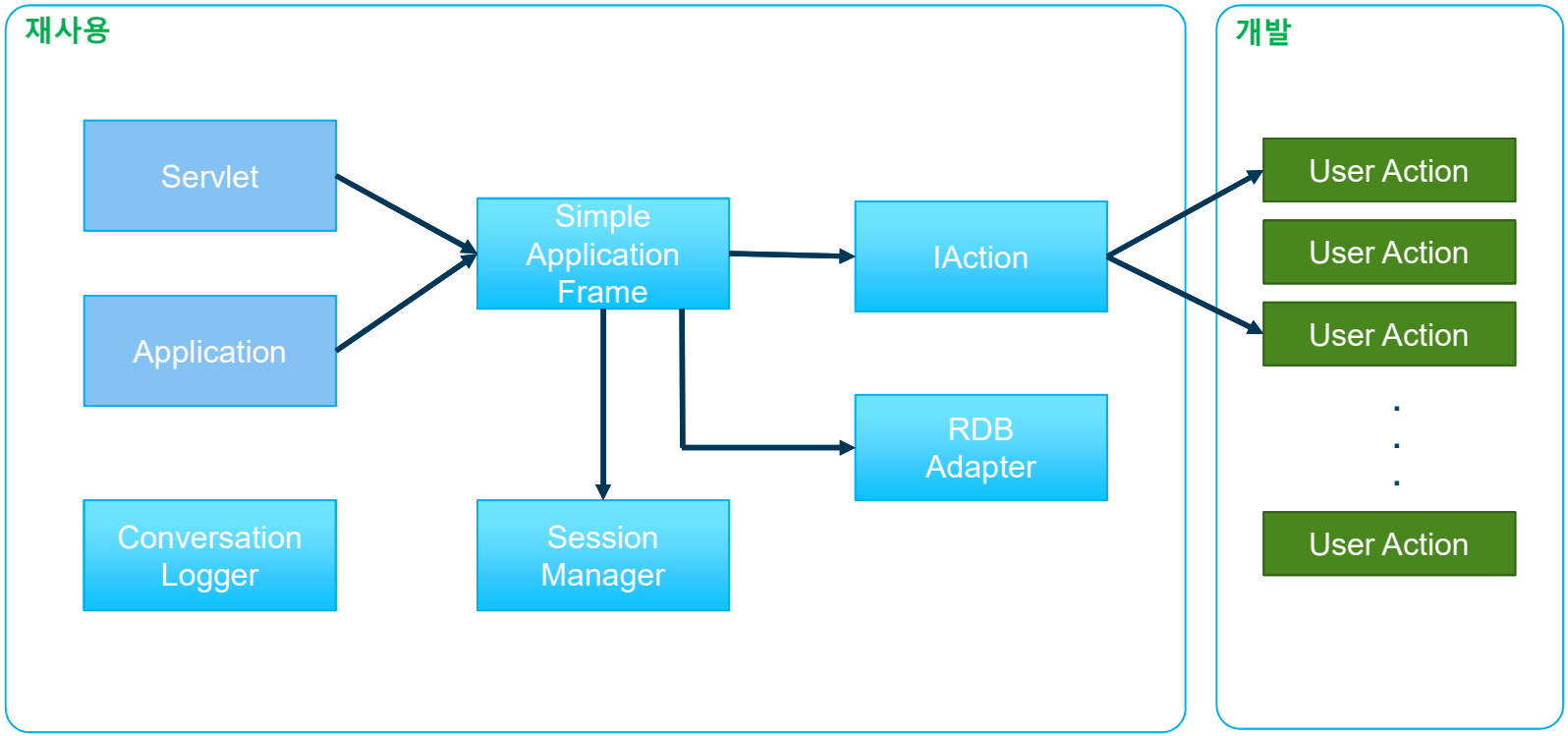
IBM

- Spring Expression (SpEL) Language
  - http://docs.spring.io/spring/docs/current/spring-framework-reference/html/expressions.html?cm_mc_uid=78859308332714736407473&cm_mc_sid_50200000=1496897051&cm_mc_sid_52640000=1496897065
- <? expression ?>
  - 확장
    “Your name is <? context.userName ?>”
    “Your name is $userName”
  - 숫자 증분
    “number: <? output.number + 1 ?>”
  - 메소드 호출
    “toppings”: “<? context.toppings.append(‘onions’) ?>”

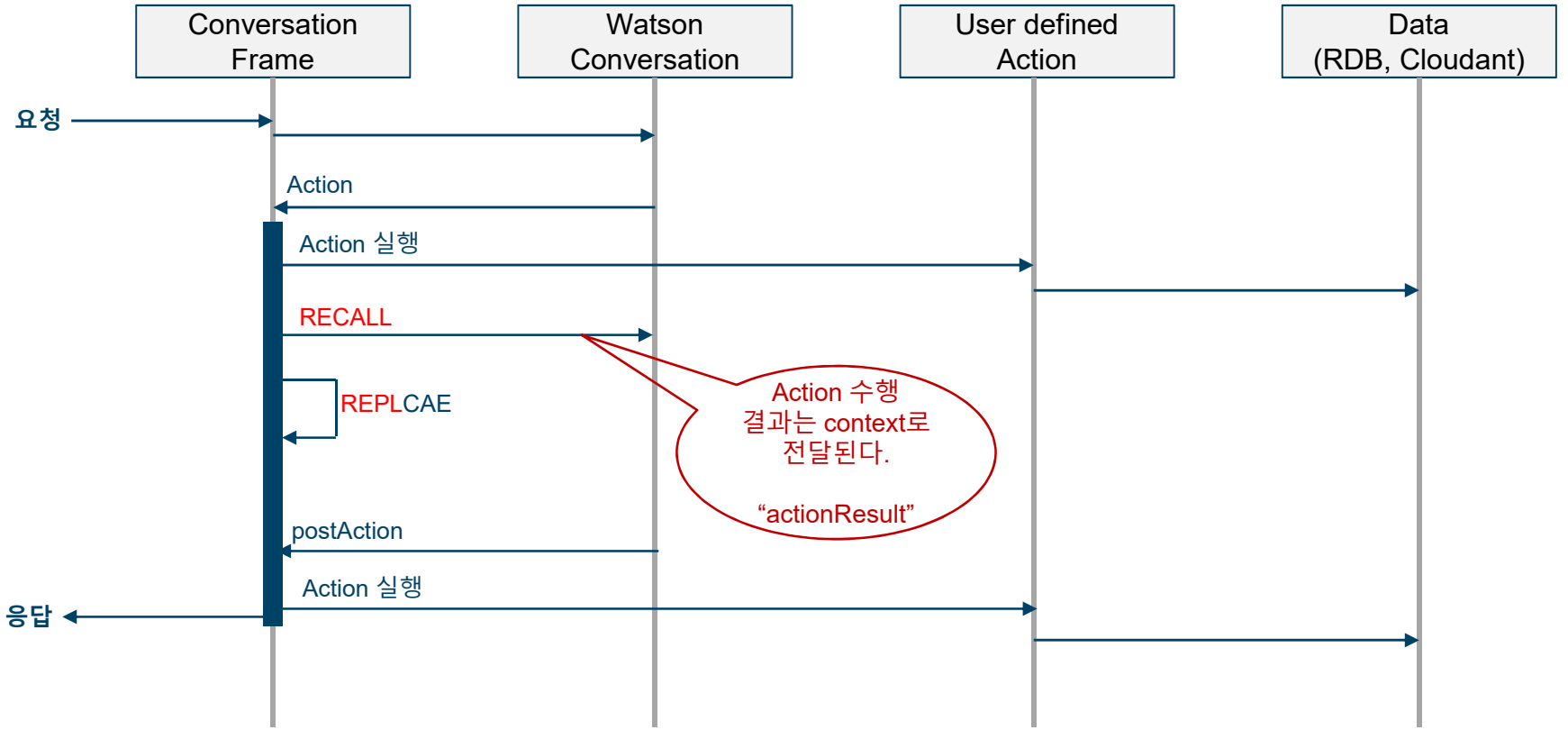
Simple Frame for Application

- 정의
  - 대화형 서비스를 빠르게 개발할 수 있는 간단한 어플리케이션 구조
- 목적
  - Watson conversation을 이용한 대화형 서비스를 빠르게 개발
  - 최소한의 수정으로 코드 재사용 극대화
  - 범용성을 가지지 않는다.
- ACTION 정의
  - Dialog 외부에서 수행해야 할 동작
    - Database Query, 외부 RESTful API 호출
  - action
    - REPL : 노드의 출력과 action 결과로 응답을 생성
    - RECALL : 노드를 다시 방문하여 응답을 생성
  - postAction : Conversation API 호출 후 실행되며, 실행 결과만 전달

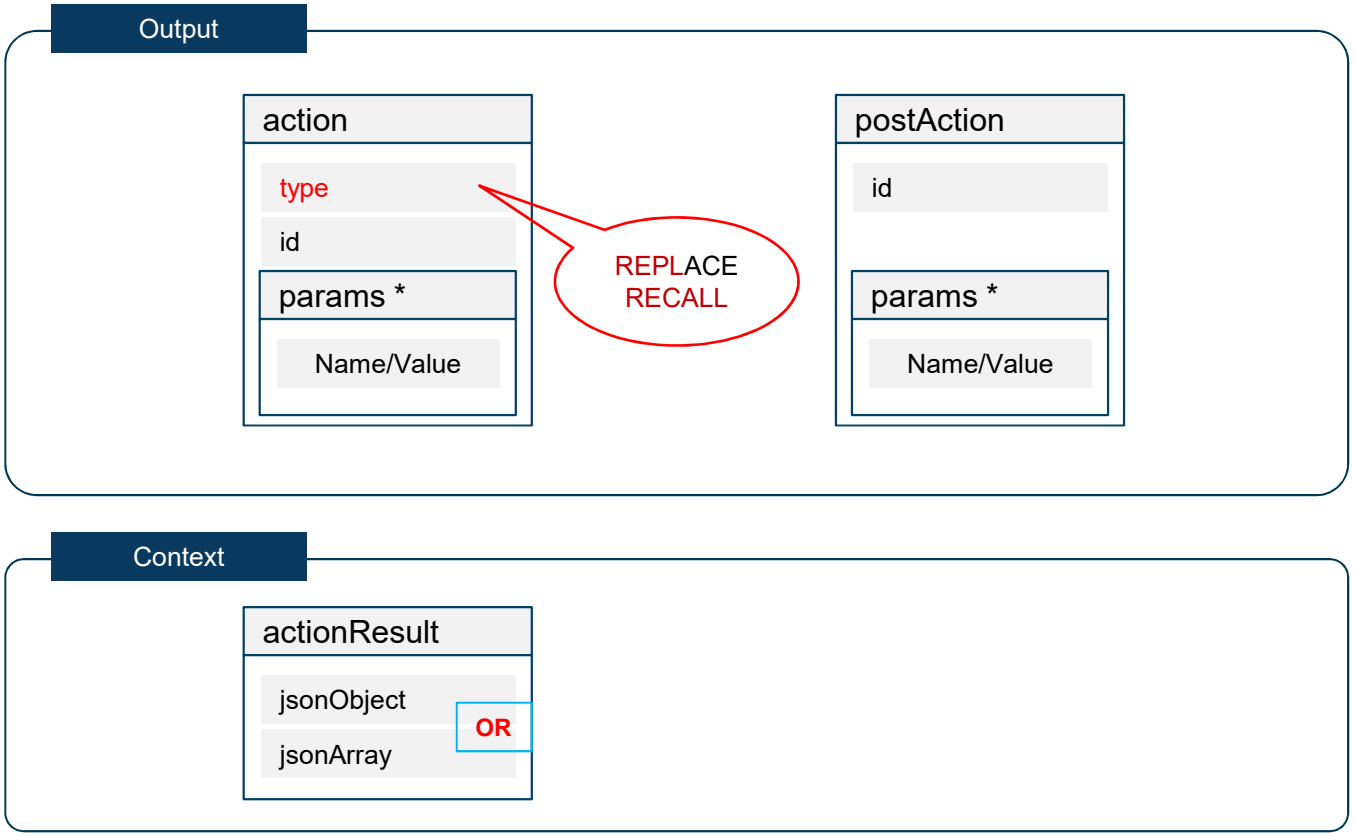
Simple Frame for Application



Simple Frame for Application



Simple Frame for Application



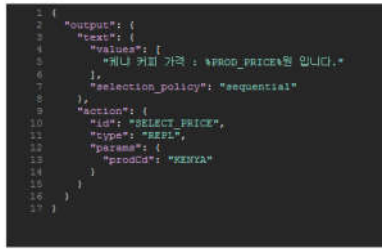
Simple Frame for Application

- Action Type : REPLACE

```
{
  "output": {
    "text": {
      "values": [
        "케냐 커피 가격 : %PROD_PRICE%원 입니다."
      ],
      "selection_policy": "sequential"
    },
    "action": {
      "id": "SELECT_PRICE",
      "type": "REPL",
      "params": {
        "prodCd": "KENYA"
      }
    }
  }
}
```

1. ‘SELECT_PRICE’로 정의된 Action을 검색한다.
2. params 항목을 파라미터로 Action을 수행한다.
3. output text에서 ‘% %’로 정의한 항목을 Action 수행결과에서 찾아 값으로 교체한다.

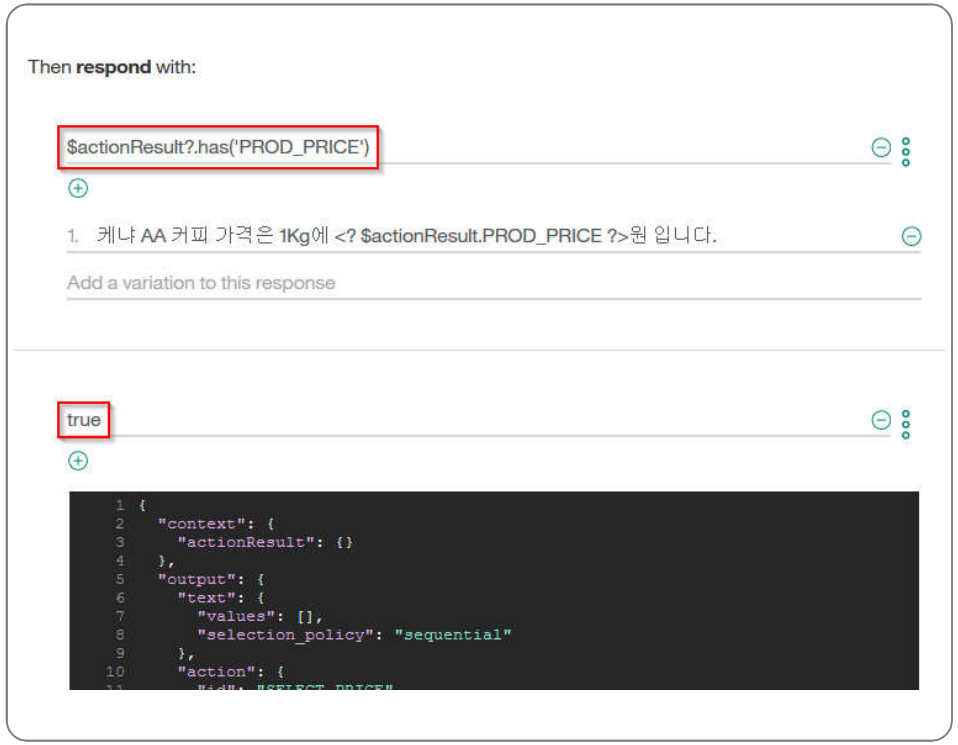
Then **respond** with:



Simple Frame for Application

- Action Type : RECALL

```
{
  "context": {
    "actionResult": {}
  },
  "output": {
    "text": {
      "values": [],
      "selection_policy": "sequential"
    },
    "action": {
      "id": "SELECT_PRICE",
      "type": "RECALL",
      "params": {
        "prodCd": "KENYA"
      }
    }
  }
}
```



Simple Frame for Application

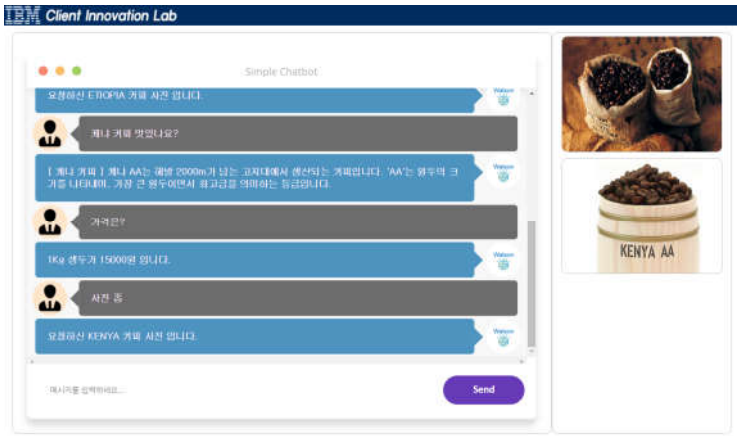
1. Dialog에서 필요한 action을 정의
   - 데이터 모델을 기준으로 사용할 컬럼 정의한다.
2. SQL 및 Action 프로그램 개발 (app.conf 파일에 등록)
   - SQL을 ibatis로 작성한다. (main.xml)
   - Action 프로그램에서 SQL을 실행하고 결과를 반환한다.
   - BaseAction 클래스를 상속받아 만든다.
3. 대화를 저장할 Logger 개발 (기본 : ConsoleLogger)
4. 어플리케이션 개발
   - SimpleAppFrame을 사용한다. (기능 추가 시 수정 필요)
   - SimpleFrameServlet을 사용한다. (기능 추가 시 수정 필요)
5. UI 개발
   - Chatbot 웹 어플리케이션 인터페이스를 구현한 index.html을 참조한다.

Simple Frame for Application

IBM

기본 실습

- 목표
  - Simple Frame for Application 환경 구성
  - 웹 페이지를 통한 챗봇 테스트



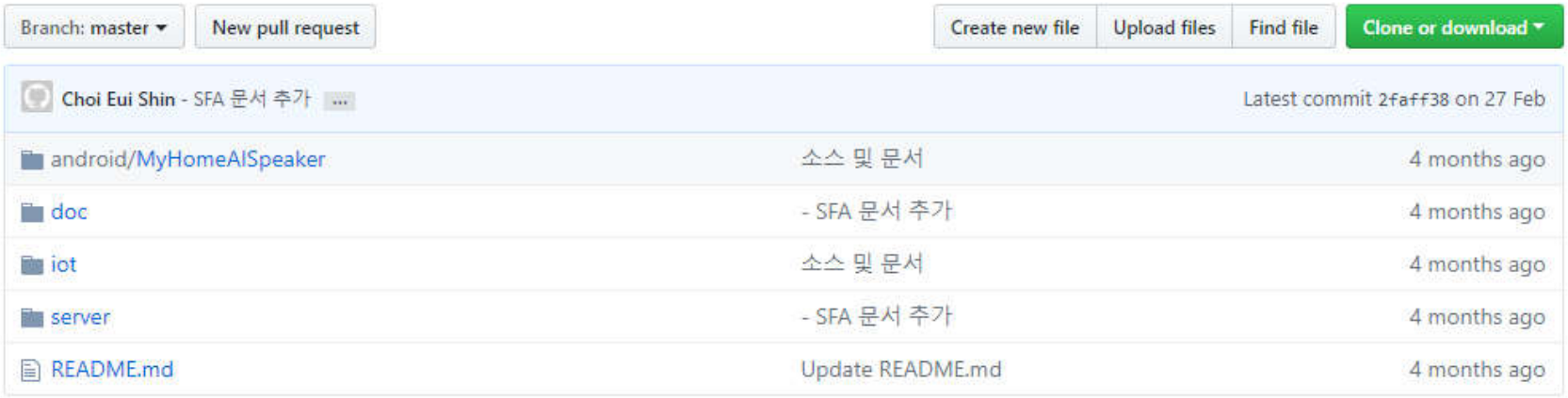
- 문서
  - https://goo.gl/Xbk1LP, https://github.com/TechOrgg/ai_speaker4_watson.git
    (Simple Frame for Application_v_2.pdf)

https://goo.gl/Xbk1LP

My AI Speaker for Watson.pdf 파일 참조

인공지능 스피커 소스코드

https://github.com/TechOrgg/ai_speaker4_watson.git



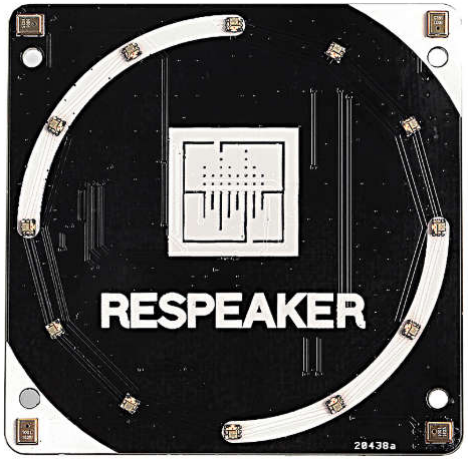
https://github.com/TechOrgg/ai_speaker_iot.git

인공지능 스피커 확장하기

IBM

- 마이크

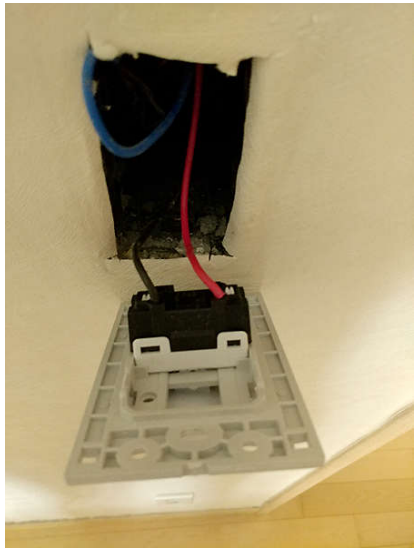
Playstation eye camera

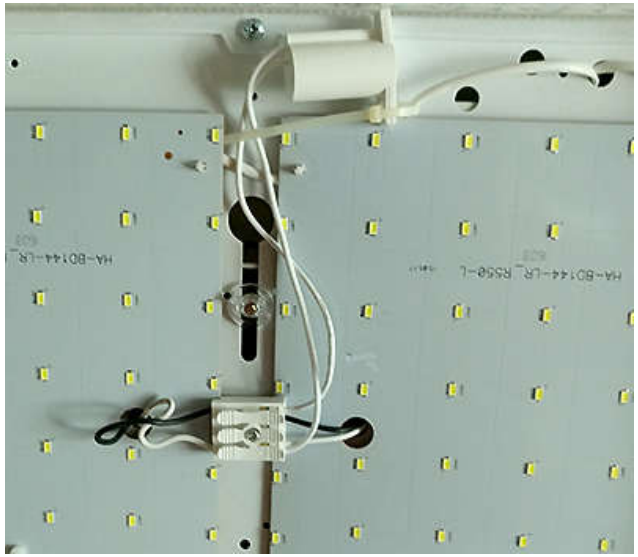


ReSpeaker 4-Mic Array

인공지능 스피커 확장하기

- 리모컨 – 스마트 스위치

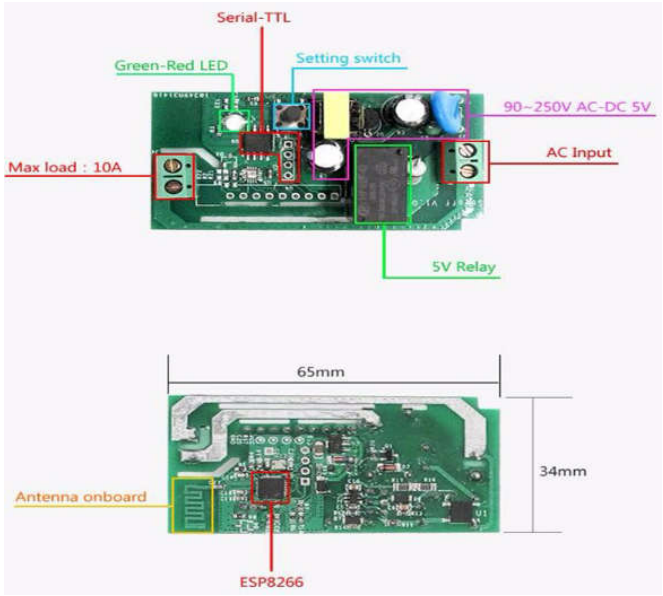




- 리모컨 – 써큘레이터

인공지능 스피커 확장하기

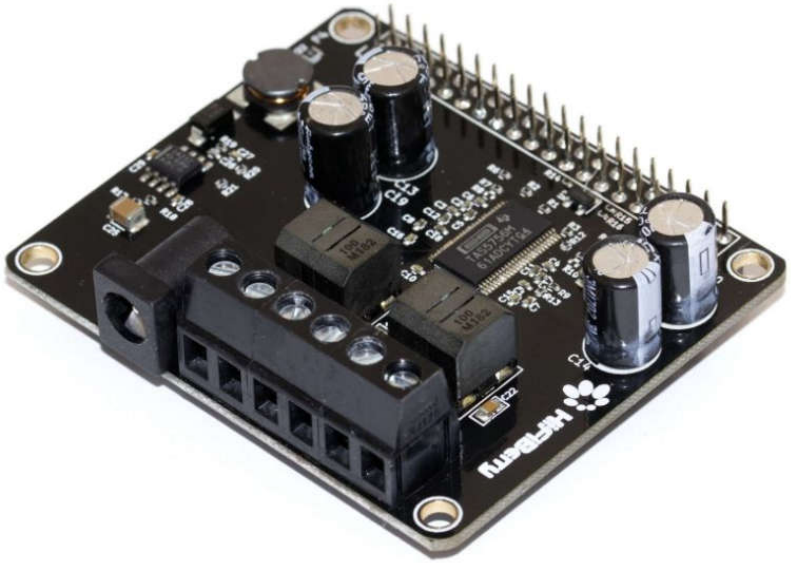
- ESP8266



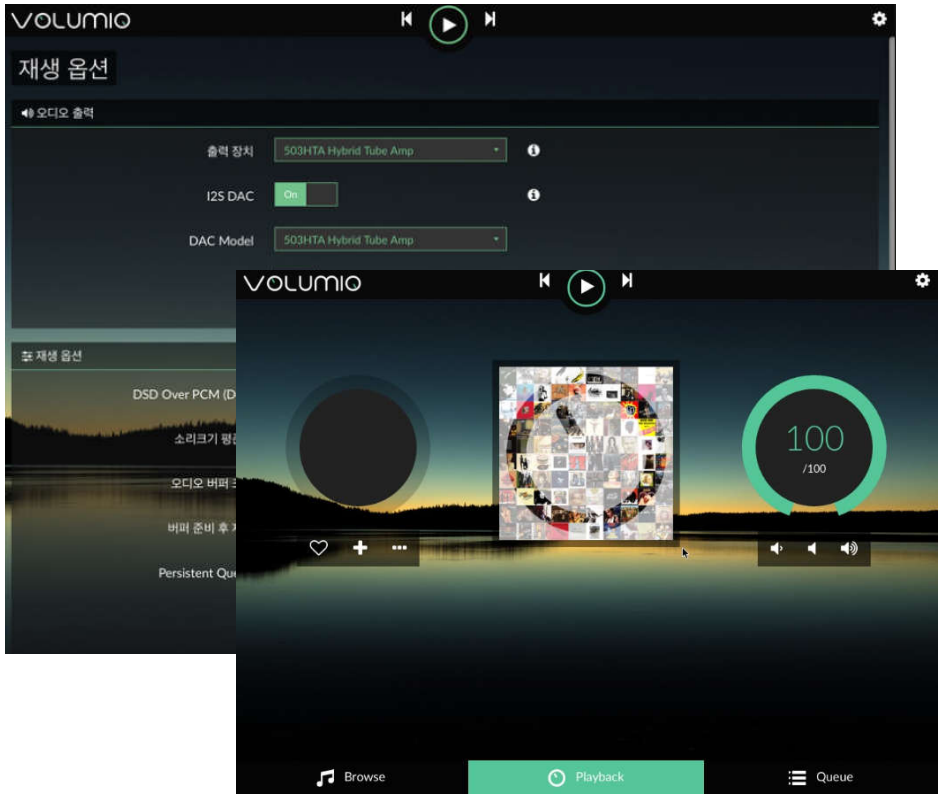
https://www.itead.cc/wiki/Sonoff

인공지능 스피커 확장하기

- 오디오 – HiFiBerry & Volumio



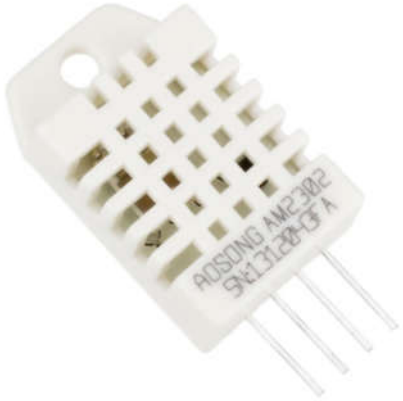
HiFiBerry Amp 2



인공지능 스피커 확장하기

- 센서 활용

DHT11



DHT22

SDS011

미세먼지 센서 - PM2.5 먼지 센서, 공기품질 체크

IBM Watson

감사합니다.

IBM