

# SW인공지능교육 - 인공지능 API를 이용한 서비스 개발 프로그램

3일차  
김성수

# 목차

1. Nugu Developers Api
2. 대화 시나리오 구축
3. 외부 Proxy 서버를 활용한 응답

# 준비하기

개발 언어 및 도구:

Python 3.x 버전

pip

GitHub 링크:

[https://github.com/kokokong/SSU\\_AI\\_AI\\_Seminar](https://github.com/kokokong/SSU_AI_AI_Seminar)



# 1 교시 -Nugu Developers Api

- Nugu Developers api 가입
- Nugu Play Kit 설명
- Custom Play 생성

# Nugu Developers api

누구나 직접 NUGU를 통하여 손쉽게 AI서비스를 개발할 수 있도록 개발자들에게 공개된 Api.

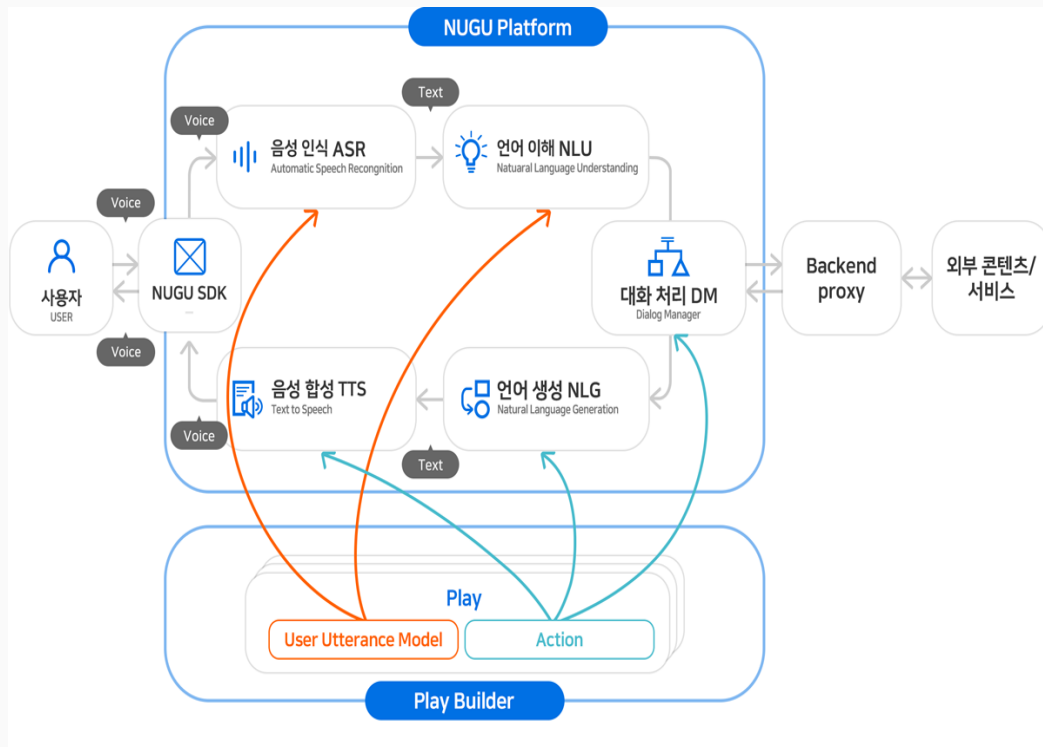
NUGU developers에서 여러분들만의 다양한 음성 시스템을 개발 가능



# Nugu Platform

## NUGU가 사용자 발화를 파악하는 과정

- 사용자 발화는 '음성 인식, 자연어 이해'를 통해 그 의도가 파악
- 대화 관리자는 파악된 의도에 맞는 액션을 수행, 사용자에게 전달할 응답을 생성
- 생성된 응답은 음성 합성 모듈을 거쳐 사용자에게 합성음으로 전달



# Nugu Play Kit

## NUGU의 서비스 단위인 Play 제작 도구

### 구성 요소

1. Play Builder
2. Intent와 Entity
3. Backend proxy
4. Capability Interface

### NUGU Play Kit

NUGU Play Kit은 NUGU의 서비스 단위인 **Play**를 제작하는 도구입니다.

개발자가 아니어도 대화 기반 서비스를 개발할 수 있는 통합 환경을 제공하여, 좋은 콘텐츠를 보유하고 있는 업체 또는 개인이 **NUGU** 사용자에게 자신의 서비스를 제공할 수 있도록 도와줍니다.

NUGU Play Kit은 다음과 같이 구성됩니다.

### Play를 제작하는 Play Builder

응답 생성에 필요한 정보를 외부 서버로부터 가져오기 위한 **Backend proxy API**  
오디오 플레이어 등 스피커 기능을 사용하기 위한 **Capability Interface**  
NUGU Play Kit을 활용하여 **Play**를 제작하는 과정은 다음과 같습니다.

**Play Builder**를 이용하여 **Play**를 만듭니다.

제작한 **Play**를 배포하기 위한 심사를 요청합니다.

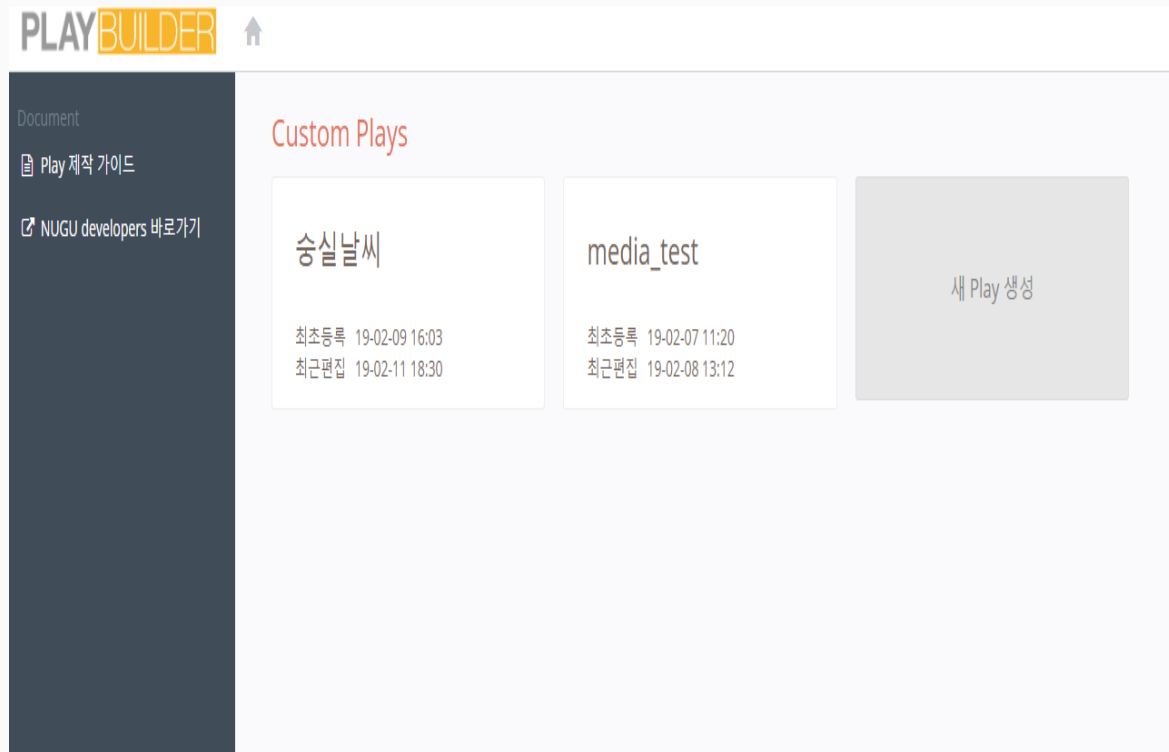
심사가 완료되면 배포를 통해 **NUGU** 사용자에게 서비스가 제공됩니다.

**Play** 및 **NUGU Play Kit**에 대한 상세한 설명은 개별 챕터에서 확인할 수 있습니다.

# Custom play 생성

NUGU Play Builder를 활용하여  
NUGU의 서비스 단위인 Play 제작

1. 새 **Play** 생성 클릭
2. play 이름 및 기본 사항 입력





# Custom play 생성

NUGU Play Builder를 활용하여  
NUGU의 서비스 단위인 Play 제작

1. 새 Play 생성 클릭
2. play 이름 및 기본 사항 입력

## Play 생성

### 1. 이름

Play 이름	<div>Aria</div> <p>PlayBuilder 내에서 사용할 이름입니다. 사용자에게 공개하게 될 Play의 이름은 별도로 등록하게 됩니다.</p>
---------	--

### 2. 기본 응답

Play 시작 안내/인사말	<div>안녕하세요. 아리아예요. 무엇을 도와드릴까요?</div> <p>예 : 안녕하세요. 누구 뮤직 이에요. 무엇을 도와드릴까요? Actions &gt; Built-in Actions &gt; <b>NUGU.ACTION.welcome</b> 에서 수정 가능합니다.</p>
Play 종료 안내/인사말	<div>아리아를 종료합니다.</div> <p>예 : 누구 뮤직을 종료할게요. Actions &gt; Built-in Actions &gt; <b>NUGU.ACTION.exit</b> 에서 수정 가능합니다.</p>
Play에서 처리할 수 없는 발화를 한 경우의 응답	<div>잘 모르겠어요 </div> <p>예 : 죄송해요. 제가 처리할 수 없는 요청이에요. 누구 뮤직을 종료할게요. Actions &gt; Built-in Actions &gt; <b>NUGU.ACTION.fallback</b> 에서 수정 가능합니다.</p>

✕ 취소

> 다음

# Custom play 생성

NUGU Play Builder를 활용하여  
NUGU의 서비스 단위인 Play 제작

1. 새 Play 생성 클릭
2. play 이름 및 기본 사항 입력
3. Play의 별칭을 변경합니다.

## Play 생성

### 1. 이름

Play 이름	<div>Aria</div> <p>PlayBuilder 내에서 사용할 이름입니다. 사용자에게 공개하게 될 Play의 이름은 별도로 등록하게 됩니다.</p>
---------	--

### 2. 기본 응답

Play 시작 안내/인사말	<div>안녕하세요. 아리아예요. 무엇을 도와드릴까요?</div> <p>예 : 안녕하세요. 누구 뮤직 이에요. 무엇을 도와드릴까요? Actions &gt; Built-in Actions &gt; <b>NUGU.ACTION.welcome</b> 에서 수정 가능합니다.</p>
Play 종료 안내/인사말	<div>아리아를 종료합니다.</div> <p>예 : 누구 뮤직을 종료할게요. Actions &gt; Built-in Actions &gt; <b>NUGU.ACTION.exit</b> 에서 수정 가능합니다.</p>
Play에서 처리할 수 없는 발화를 한 경우의 응답	<div>잘 모르겠어요 </div> <p>예 : 죄송해요. 제가 처리할 수 없는 요청이에요. 누구 뮤직을 종료할게요. Actions &gt; Built-in Actions &gt; <b>NUGU.ACTION.fallback</b> 에서 수정 가능합니다.</p>

✕ 취소

> 다음

# Custom play 생성

NUGU Play Builder를 활용하여  
NUGU의 서비스 단위인 Play 제작

1. 새 Play 생성 클릭
2. play 이름 및 기본 사항 입력
3. **Play**의 별칭을 변경합니다.

## 기본 정보

Play 이름	weather
Play 별칭	승원별칭

## Capability Interface 추가

AudioPlayer 제어	<input type="checkbox"/> off
----------------	------------------------------

## Prompt 발화 옵션

발화속도 (speed)	100% (Default)
발성음 높낮이 (pitch)	100% (Default)
문장 사이 묵음 구간 길이 (pause1)	600ms (Default)
끝어 읽기 뒤 묵음 구간 길이 (pause2)	300ms (Default)

[다러 돌아보기](#)[발화옵션 적용하기](#)

# 2교시 - 대화 시나리오 구축

- Intent
- Entity
- Action

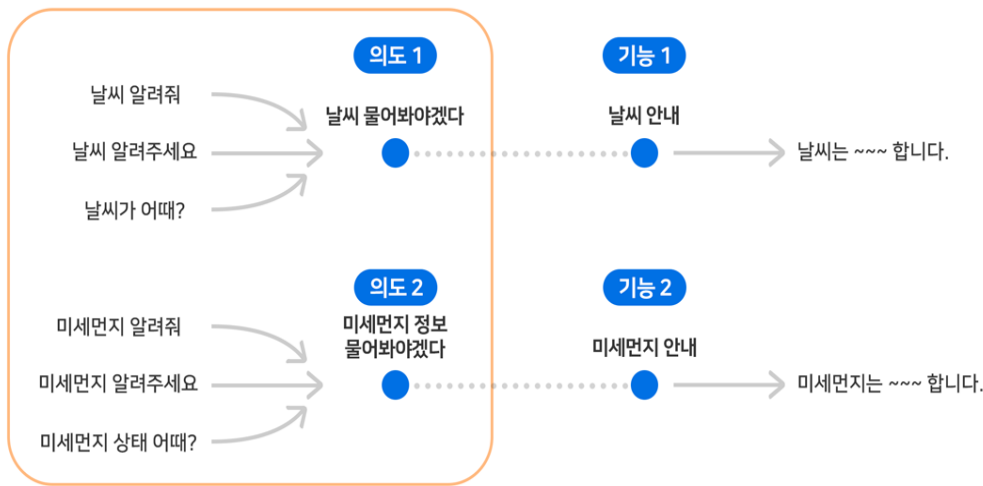
# 대화시나리오의 요소

- 대화 시나리오는 크게 3가지 요소로 구성되어 있다.
- Intent
- Entity
- Acton

Intent	Intent는 사용자가 발화를 통해 수행하고자 하는 기능을 구체적으로 구분한 범주이며, Play의 필수 구성 요소입니다.
Entity	Entity는 Intent의 예상 발화 정의 시 해당 발화문 내에서 특정 텍스트를 지정하여 정의합니다.
Action	NUGU에서는 하나의 Intent를 처리하는 최소 응답 단위

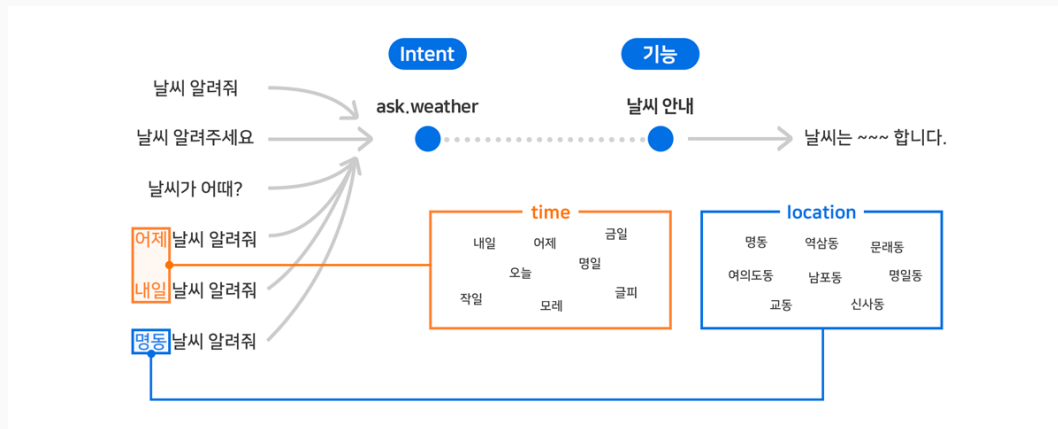
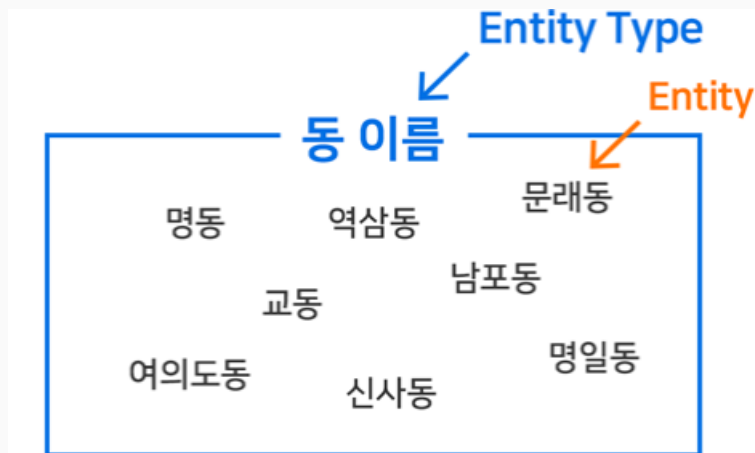
# Intent

1. Intent는 사용자가 발화를 통해 수행하고자 하는 기능을 구체적으로 구분한 범주
2. 입력된 예상 발화를 바탕으로 NLU 엔진은 자동으로 학습하여 사용자의 실제 발화를 분석합니다.
3. Intent를 정의하기 위해서는 기능 수행을 위한 사용자의 예상 발화를 일정 수준 이상 입력해야 합니다.
4. 동일한 Intent에 대해 사용자들은 각자의 상황/성향에 따라 여러 가지 형태 또는 문형으로 발화할 수 있으므로, 이를 고려하여 다양한 문형의 예상 발화를 입력하는 것이 좋습니다



# Entity

1. Entity: 부가 상세 정보를 의미하는 개체.
2. Entity: Intent의 예상 발화 정의 시 해당 발화문 내에서 특정 텍스트를 지정하여 정의.
3. 예상 발화에서 정의된 Entity는 사용자 발화에서 동일한 의미/유형의 다른 어휘들로 나타날 수 있음
4. 정의한 Entity가 사용자 발화에 누락 시 정상적으로 기능을 동작시키지 못하는 이 경우에는 Entity를 필수로 지정 가능



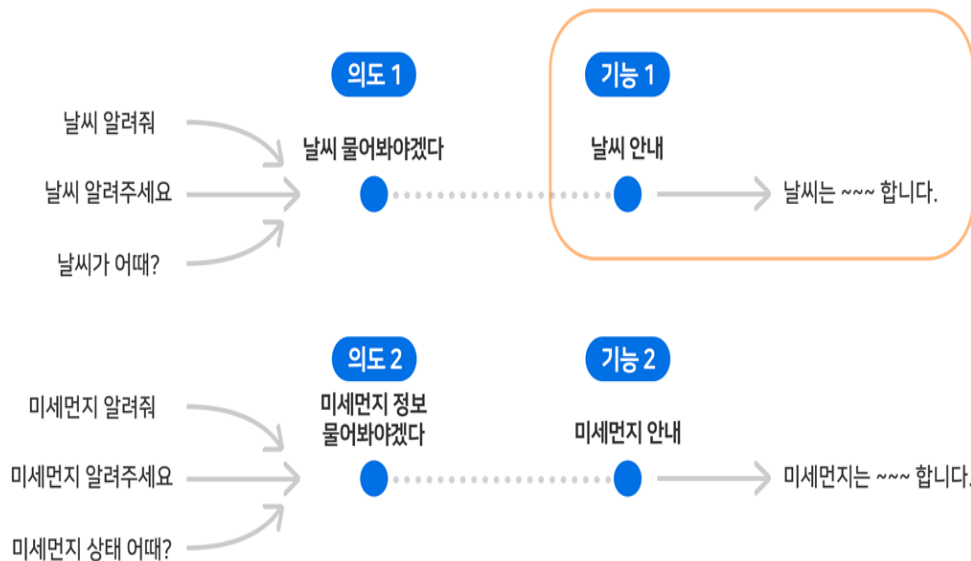
# Action

1. Action: NUGU의 하나의 Intent를 처리하는 최소 응답 단위

2. 'Action 등록'을 통해 사용자가 목적 달성을 위한 응답을 미리 지정

\*일반적으로 하나의 Intent에 대하여 하나의 Action을 정의하게 됩니다.

3. Action은 Custom Action과 Built-in Action으로 구분





# 3교시 - Backend Proxy

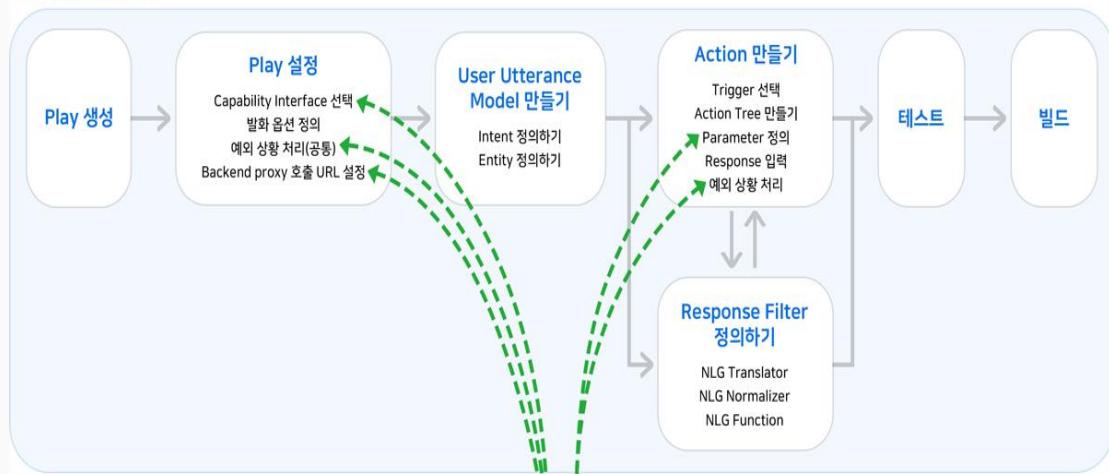
- Backend Proxy
- Backend Proxy 서버 생성 및 연결

# Backend Proxy

## Backend proxy를 사용하기 순서

1. Backend proxy 서버를 구축합니다.
2. Backend proxy를 Play와 연결합니다.
3. Play 개발자는 Backend proxy 개발자와 다음 사항을 공유해야 합니다.
  - 정의된 Entity Parameter 및 전달 될 값의 형태
  - Entity를 정규화 할 경우 해당 Entity의 대표값을 전달
  - 필요한 Backend Parameter와 어떤 값이 담길지를 안내
  - 예외 상황들에 대해 안내 각 상황별 code 요청
4. Backend Parameter를 활용, 예외 상황 Prompt 입력

## Play Builder 내부



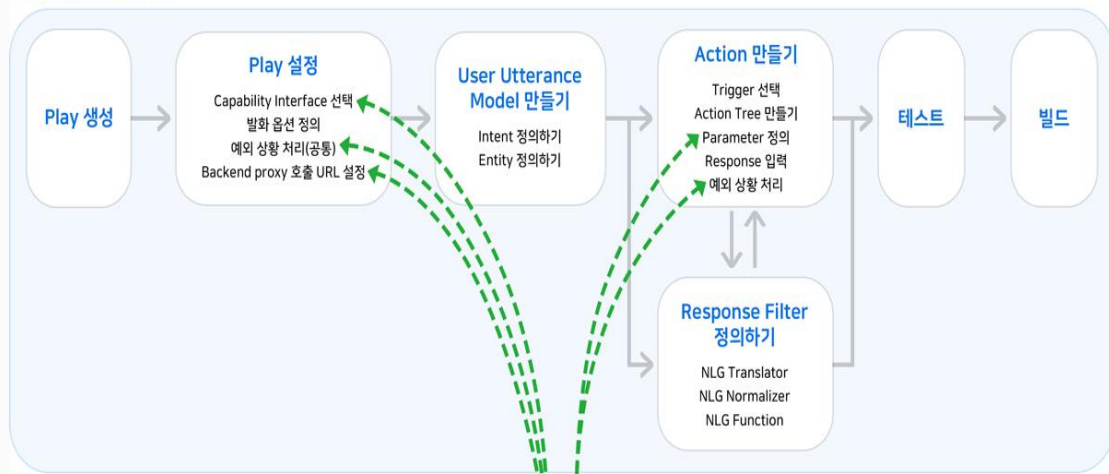
## Play Builder 외부



# Backend Proxy

Backend proxy: Play Builder에서 응답을 생성하기 위해 필요한 정보를 외부로부터 가져와야 하는 경우 NUGU 플랫폼에서 REST API를 통해 호출하는 서버

## Play Builder 내부



## Play Builder 외부



# Web Crawling

Nugu play가 외부 Back-end Proxy  
서버와 통신 할 때 웹 크롤링을 통해  
질문에 맞는 답을 얻는다.

파이썬의 BeautifulSoup를 사용하여 웹  
크롤링을 해보도록 한다.

```
pip install bs4
```

# BeautifulSoup



# Proxy 연결

proxy.py 파일을 실행하여 프록시 서버와 play를 연결시켜 봅니다.

python proxy.py

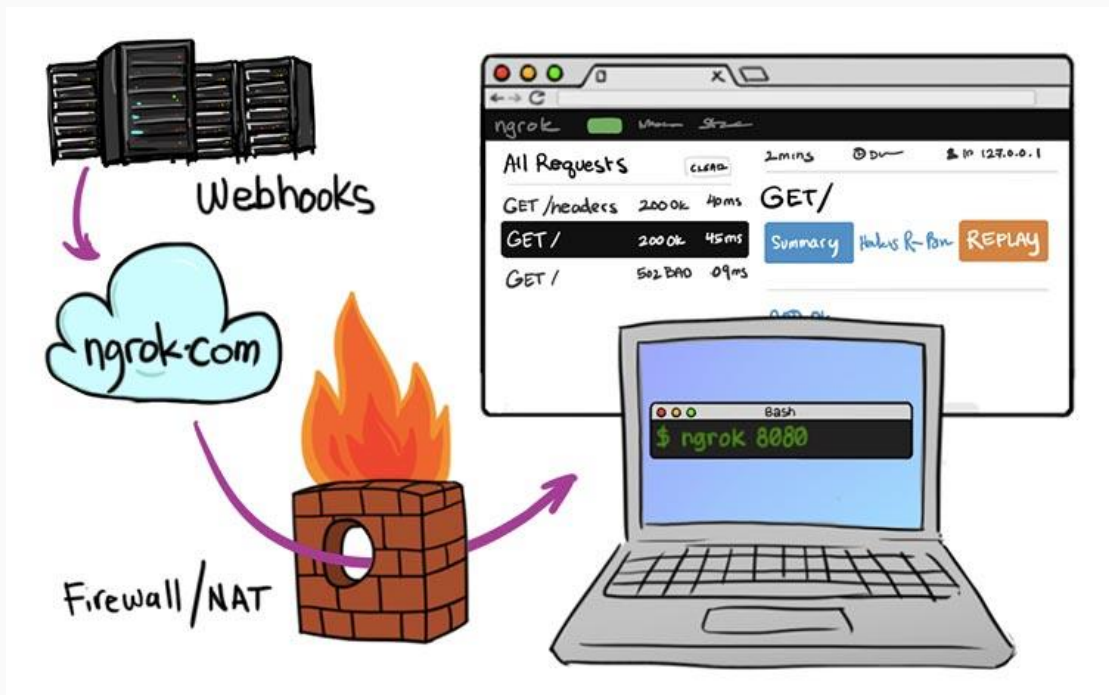
```
{  
  "version": "2.0",  
  "resultCode": "OK",  
  "output": {  
    "Answer": "msg"  
  }  
}
```



# Ngrok

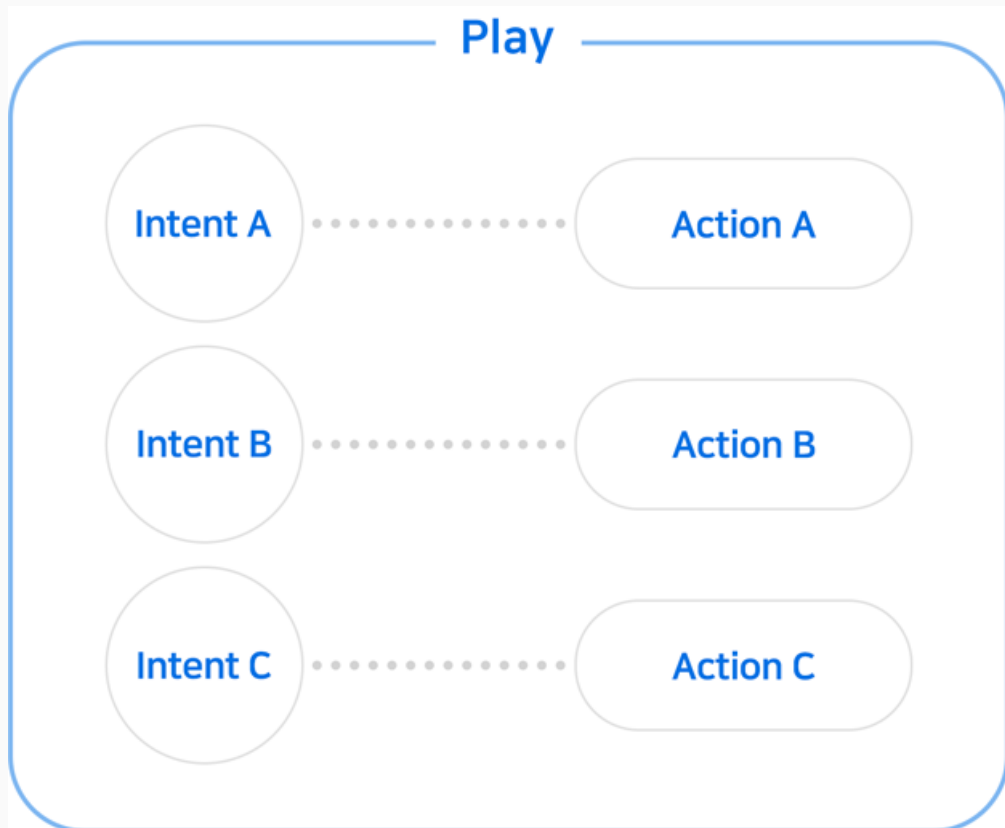
- Localhost 서버를 별도의 포트 포워딩 없이 웹 서버 환경으로 구축할 수 있게 해주는 프로그램
- Ssl 설치 없이 https로 통신하는 redirect url 생성 가능

<https://ngrok.com/>



# Branch Action

- 현재까지의 대화 시스템은 1번의 질문에 1번의 응답 후 종료
- 하나의 Intent라도 그것을 처리하는 로직이 복잡할 경우 하나의 Action만으로는 응답이 어려울 수 있음
- 생성된 응답은 음성 합성 모듈을 거쳐 사용자에게 음성으로 전달



# Branch Action

- 같은 의도이더라도 조금씩 다른 요구사항이나 조건들이 있을 경우 다양한 응답을 반환
- 한 Action 하위에 2개의 Action을 반환할 수 있음
- 생성된 응답은 음성 합성 모듈을 거쳐 사용자에게 합성음으로 전달





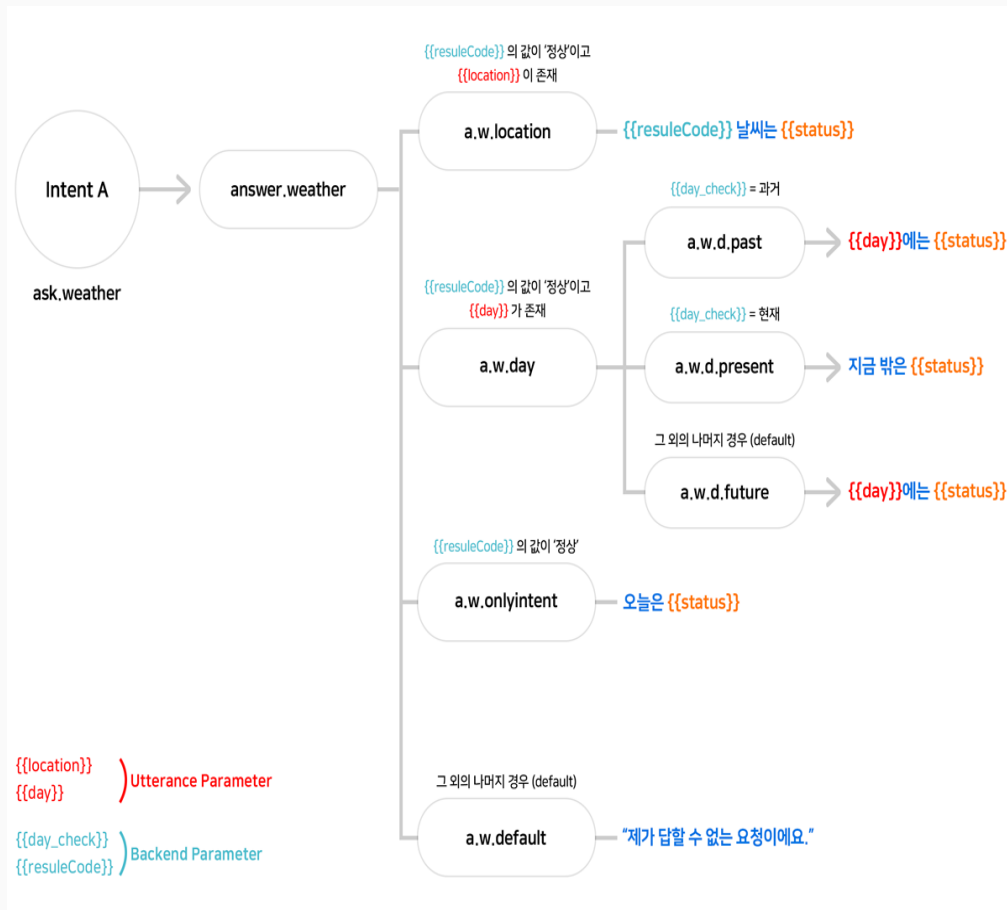
# Branch Action

- 같은 의도이더라도 조금씩 다른 요구사항이나 조건들이 있을 경우 다양한 응답을 반환
- 한 Action 하위에 2개의 Action을 반환할 수 있음
- 생성된 응답은 음성 합성 모듈을 거쳐 사용자에게 합성음으로 전달



# Branch Action

- 같은 의도이더라도 조금씩 다른 요구사항이나 조건들이 있을 경우 다양한 응답을 반환
- 한 Action 하위에 2개의 Action을 반환할 수 있음
- 생성된 응답은 음성 합성 모듈을 거쳐 사용자에게 합성음으로 전달



Q&A

감사합니다.

HCI Lab

학부연구원 김성수

[sdasd127@gmail.com](mailto:sdasd127@gmail.com)

