

카카오 오픈빌더로 챗봇 만들기 #02

Joonhwan Lee

Human-Computer Interaction+Design Lab. | Seoul National University

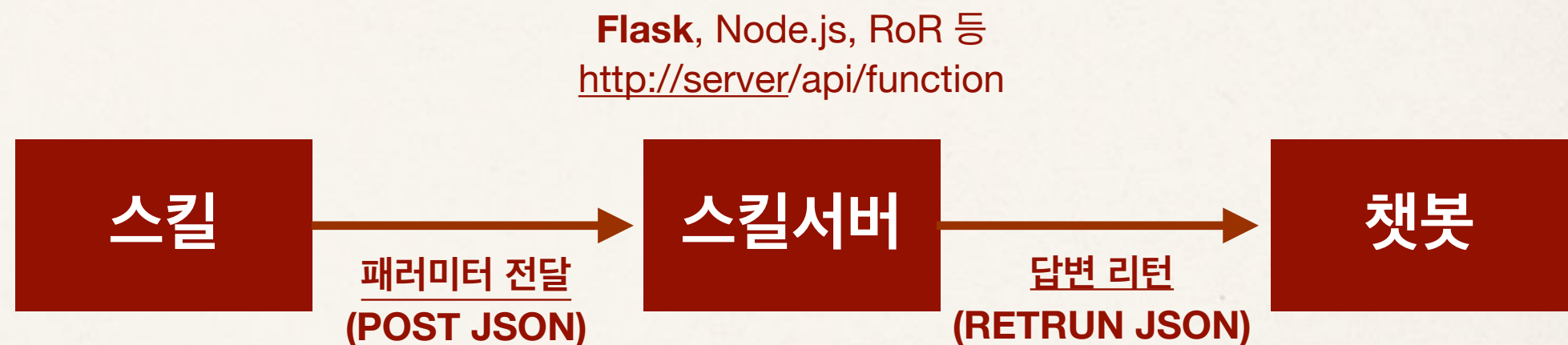
스킬

스킬이란?

- ♦ 블록의 출력과 비슷한 역할
- ♦ 블록은 미리 준비된 대답만 가능하지만 스킬은 상황에 따라 다른 답변을 제공
 - ♦ 오늘 미세먼지 수치는?
 - 시간마다 다른 정보를 수집해서 제공
 - ♦ 자하연 식당 전화번호는?
 - 데이터베이스에 저장된 여러 식당의 전화번호 중 요청한 번호를 제공
- ♦ 모든 답변을 다 준비해둘 수 없기 때문에 API 서버에서 답변을 처리하여 스킬로 넘겨주는 방법을 사용

스킬이란?

- 스킬서버를 사용하기 위해서는 원하는 정보를 서버에 요청해야 함 (패러미터 전달)



restaurant_name => '감골식당'

```
res = {"version": "2.0", "template": {  
  "outputs": [{"simpleText": {  
    "text": answer}}]}}
```

GET vs. POST method

- ✦ GET과 POST는 HTTP프로토콜을 이용해서 서버에 무언가를 전달할 때 사용하는 방식
- ✦ GET
 - ✦ https://news.naver.com/main/ranking/read.nhn?mid=etc&sid1=111&rankingType=popular_day&oid=293&aid=0000023882&date=20190407&type=1&rankingSeq=8&rankingSectionId=105
 - ✦ URL에 패러미터를 포함하여 특정 데이터를 서버에 요청
 - ✦ http://서버주소/post_id=101
 - ✦ URL에 길이제한이 있어서 많은양의 데이터는 보내기 어렵다

GET vs. POST method

✦ POST

- ✦ 데이터를 (주로 JSON 형태로) 서버에 포스팅
- ✦ 서버에 있는 함수가 데이터를 받아 처리하고 그 결과를 반환
- ✦ URL에는 패러미터가 포함되지 않음

GET vs. POST method

```
{
  "action": {
    "name": "7e0y10l37s",
    "clientExtra": null,
    "params": {
      "restaurant_name": "감골식당"
    },
    "id": "r6k5no5tfemwi0ms43srtss2",
    "detailParams": {
      "restaurant_name": {
        "origin": "감골식당",
        "value": "감골식당",
        "groupName": ""
      }
    }
  }
}
```

GET vs. POST method

```
{
  "action": {
    "name": "7e0y10l37s",
    "clientExtra": null,
    "params": {
      "restaurant_name": "감골식당"
    },
    "id": "r6k5no5tfemwi0ms43srtss2",
    "detailParams": {
      "restaurant_name": {
        "origin": "감골식당",
        "value": "감골식당",
        "groupName": ""
      }
    }
  }
}
```

`["action"]["detailedParams"]["restaurant_name"]["value"]`

API 서버 만들기

API Server

- ✦ API Server 는 POST 된 데이터를 처리한 후 결과를 반환하는 역할을 수행
- ✦ 최근의 트렌드는 Node.js, Flask, Django, Ruby on Rails 등을 사용
- ✦ 이 튜토리얼에서는 Python 기반의 Flask 를 사용할 예정

API Server

- ♦ Flask, Node.js 를 설치하기 위해서는 서버가 필요
- ♦ 간단히 계정 개설 만으로 서버 사용이 가능한 서버를 사용
 - ♦ 구름 IDE: <https://ide.goorm.io/>
- ♦ AWS를 사용하는 방법은 따로 소개할 예정

API Server: 구름 서버 컨테이너 만들기

[←](#) [→](#) [↺](#) [ide.goorm.io/container/new](#) [☆](#) [👤](#) [⋮](#)

[돌아가기](#) [컨테이너 생성](#) [생성 \(Ctrl + M\)](#)

이름 •

컨테이너 이름을 입력해주세요.

설명

컨테이너 설명을 입력해주세요.

지역

☐ 오리건 (미국) ☒ 서울 (한국) ☐ 프랑크푸르트 (독일) ☐ 뭄바이 (인도)

공개 범위

☒ Public ☐ Private

Public으로 설정 시 컨테이너 갤러리에 공개되어 누구나 이 컨테이너에 접속할 수 있습니다.
민감한 정보(서버 비밀번호, 개인 정보,...)를 다룰 경우 노출될 수 있음을 주의바랍니다.

템플릿

☒ Template ☐ ZIP / TAR ☐ Github ☐ Bitbucket ☐ Git / SVN ☐ Kakao Oven [NEW](#)

배포

☒ Not used ☐ Heroku ☐ AWS Elastic Beanstalk

소프트웨어 스택

C/C++

HTML/CSS/JS

Python

Django

Flask

PyTorch



API Server: 구름 서버 컨테이너 만들기

← → ↺

ide.goorm.io/container/new

☆

⋮

돌아가기

컨테이너 생성

생성 (Ctrl + M)

Containers

내 컨테이너 1

공유받은 컨테이너 0

● snubob

⚙

snubob server

소프트웨어 스택

저장 공간

공유 링크

마지막 실행 5/3/2020, 3:04:42 PM

👤

터미널 실행

Run

소프트웨어 스택

C/C++

HTML/CSS/JS

Python

Django

Flask

PyTorch

workspace / snubob / application.py



프로젝트

- snubob
 - application.py

Left sidebar icons: file explorer, terminal, search, resources monitor, lint, and settings.

```

1 import os
2 import sys
3 from flask import Flask, request, jsonify
4
5 app = Flask(__name__)
6
7 @app.route("/")
8 def home():
9     return "snubob api server"
10
11 @app.route("/hello")
12 def hello():
13     return "hellp, snubob"
14
15 if __name__ == "__main__":
16     app.run(host='0.0.0.0', port=int(sys.argv[1]))
17
    
```

디버그 **터미널** 검색 리소스 모니터 린트

```

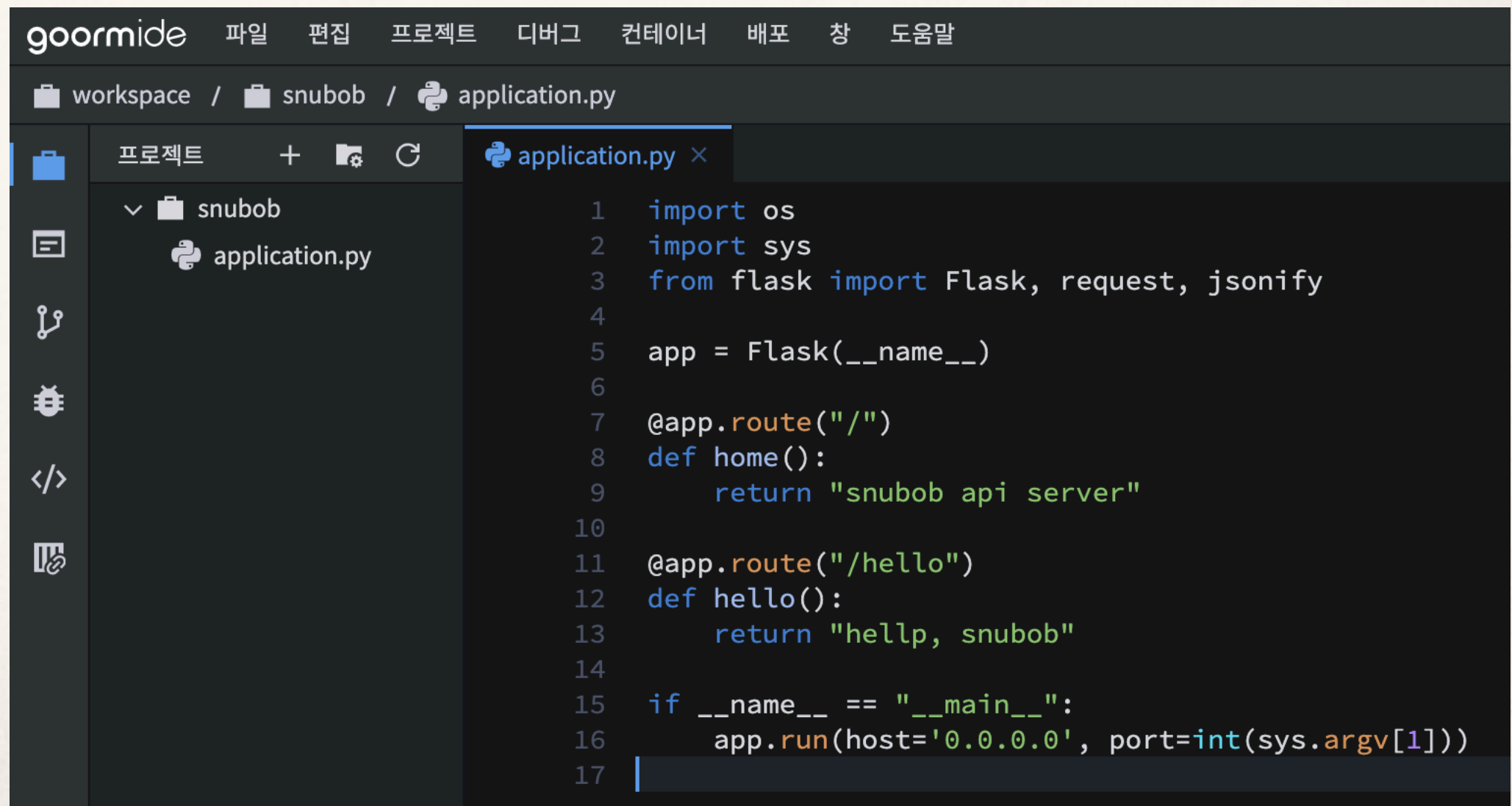
cd /workspace/snubob; clear export LANG=ko_KR.UTF-8;export LANGUAGE=ko_KR.UTF-8;locale-gen
;alias python=python3;alias pip=pip3;source ~/.profile; [ ! -f ~/.custom_env ] || source ~
/.custom_env; clear;
root@goorm:/workspace/snubob#
    
```

Flask 설치

- ✦ **`pip install Flask` (혹은 `pip3 install Flask`)**
 - ✦ 구름 IDE에서 기본 Python container를 만들면 python 3가 설치됨 (설정에 따라 python 2가 설치될 수도 있으니 확인해야)
 - ✦ 구름 IDE에는 기본적으로 Flask 가 설치되어 있으니 `pip list` 를 이용하여 설치 여부 확인.
 - ✦ 설치가 되어 있지 않다면 python 버전에 따라 Flask 설치
 - ✦ Python version 3: `pip3 install Flask`
 - ✦ Python version 2: `pip install Flask`

서버 앱 설치

- ✦ 컨테이너 아래에 기본으로 생성된 index.py 를 application.py 로 이름 변경 (이름 변경은 꼭 필요한 것은 아님)
- ✦ application.py 에 다음과 같이 코드 입력



```
goormide  파일  편집  프로젝트  디버그  컨테이너  배포  창  도움말
workspace / snubob / application.py

프로젝트 + [icons]
  ▼ snubob
    application.py

1  import os
2  import sys
3  from flask import Flask, request, jsonify
4
5  app = Flask(__name__)
6
7  @app.route("/")
8  def home():
9      return "snubob api server"
10
11 @app.route("/hello")
12 def hello():
13     return "hello, snubob"
14
15 if __name__ == "__main__":
16     app.run(host='0.0.0.0', port=int(sys.argv[1]))
17
```

서버 앱 설치

- ♦ Flask 임포트

```
import os
import sys
from flask import Flask, request, jsonify
```

- ♦ App 생성

```
app = Flask(__name__)
```

- ♦ App 실행

```
if __name__ == "__main__":
    app.run(host='0.0.0.0', port=int(sys.argv[1]))
```

API 서버 테스트

- ◆ 다음과 같이 API route 생성

```
@app.route('/')  
def hello():  
    return 'snubob api server'
```

- ◆ @app.route('/') → root (<http://server/>)
- ◆ @app.route('/hello') → <http://server/hello>
- ◆ 서버프로그램의 실행 (터미널에서 다음의 명령어 수행)
 - ◆ python3 application.py 80

API 서버 주소 확인

- ◆ 구름 IDE 메뉴에서 “프로젝트 > 실행 URL과 포트” 선택

실행 URL과 포트

서버 실행에 사용되는 도메인 주소를 등록할 수 있습니다.

URL	<input type="text" value="https://"/>	<input type="text" value=".run.goorm.io"/>
PORT	<input type="text" value="80 ~ 65000"/>	<button>등록</button>

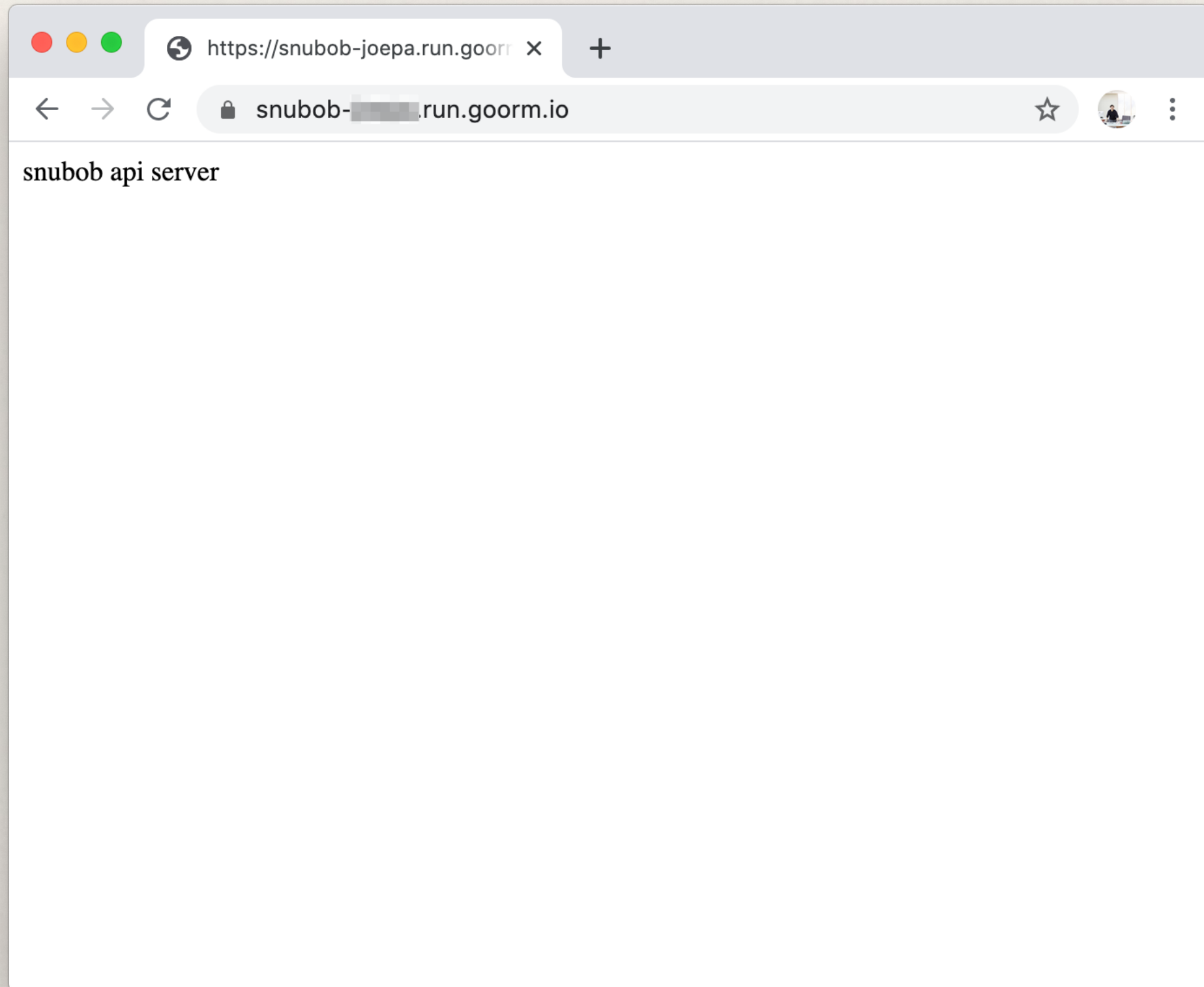
등록된 URL과 포트 (컨테이너당 최대 3개)

실행용	URL	Port
	https://snubob-goorm.io	80

웹URL



API 서버 테스트



스킬 만들기

스킬 페이지 설정

←

get_telephone_number

새 버전으로 저장

저장

기본 정보

ver. 1 | joonhwan | 3/22/2019 10:42:32 AM

☐ 기본 스킬로 설정

설명

설명을 입력해주세요

URL *

http://server_address/api/telephone

테스트 URL

URL을 입력해주세요

헤더값 입력

restaurant_n

@restaurant_name

Key

Value

Key

Value

테스트 헤더값 입력

Key

Value

Key

Value

Key

Value

블록 설정

telephone

저장

파라미터 설정

일반 파라미터

restaurant_name
@restaurant_name | \$restaurant_name

telephone
@telephone | \$telephone

필수 파라미터 ?

스킬에 보낼 파라미터 중 필수로 필요한 파라미터를 지정합니다.
파라미터 값이 채워지지 않으면, 되묻기 질문을 통해 값을 채우게 합니다.

파라미터 만들기

파라미터 그룹생성

봇 응답형식 설정

카카오 톡

스킬데이터 사용

스킬 API 코드

◆ 다음의 코드 추가

```
54 @app.route('/api/telephone', methods=['POST'])
55 def telephone():
56     phone = {
57         "감골식당": "02-880-5544",
58         "동원관": "02-880-8697",
59         "학생회관": "02-880-5543",
60     }
61     req = request.get_json()
62     rest_name = req["action"]["detailParams"]["restaurant_name"]["value"]
63     answer = "{}의 전화번호는\n{} 입니다.".format(rest_name, phone[rest_name])
64     res = {
65         "version": "2.0",
66         "template": {
67             "outputs": [
68                 {
69                     "simpleText": {
70                         "text": answer
71                     }
72                 }
73             ]
74         }
75     }
76     return jsonify(res)
```


스킬 API 코드

- ◆ API route 설정

- ◆ `@app.route('/api/telephone', methods=['POST'])`
- ◆ POST 방식, 주소는 `server_addr/api/telephone`

- ◆ 데이터 받아오기

- ◆ `req = request.get_json()`

- ◆ 필요한 데이터 추출

- ◆ `rest_name = req["action"]["detailParams"]
["restaurant_name"]["value"]`

스킬 API 코드

- ◆ 데이터 처리 후 response 생성
 - ◆ 코드 처음에 전화번호부 dictionary 생성
 - ◆

```
phone = {  
    "감골식당" : "02-880-5544",  
    "동원관" : "02-880-8697"  
}
```
 - ◆

```
answer = phone[rest_name]
```
 - ◆ SimpleText 형식으로 데이터 반환 (코드 참고)
 - ◆

```
return jsonify(res)
```
 - ◆ <https://i.kakao.com/docs/skill-response-format#skillpayload>

스킬 서버 테스트

스킬 테스트

● URL ○ Test URL

요청할 파라미터 값 입력



restaurant_name	샤반	string	✕
Key	Value	string	✕

JSON

직접 params 혹은 detailParams 수정 시 왼쪽 표는 사용할 수 없습니다.

```
21  }
22  },
23  "bot": {
24    "id": "5c81be135f38dd47672169c3",
25    "name": "봇 이름"
26  },
27  "action": {
28    "name": "hjrmgnf8mv",
29    "clientExtra": null,
30    "params": {
31      "restaurant_name": "샤반"
32    },
33    "id": "hptelddiuzgwfvtjkgxudxy",
34    "detailParams": {
35      "restaurant_name": {
36        "origin": "샤반",
37        "value": "샤반",
```

클립보드로 복사

스킬서버로 전송

응답 초기화

응답 미리보기

```
1  {
2    "template": {
3      "outputs": [
4        {
5          "simpleText": {
6            "text": "샤반의 전화번호는\n02-871-6933 입니다."
7          }
8        }
9      ]
10   }
```

응답 결과 로그

```
> 22:48:14 [INFO] 요청 형식을 서버로 전송하여 결과를 검증하는 중입니다...
> 22:48:15 [INFO] 올바른 스킬 서버 응답입니다.
>
```

Github Repository

- ✦ 이 강의에 사용된 코드 및 템플릿은 다음의 GitHub repository 에서 다운로드 가능.
 - ✦ <https://github.com/monot/snubob>

Questions...?
