

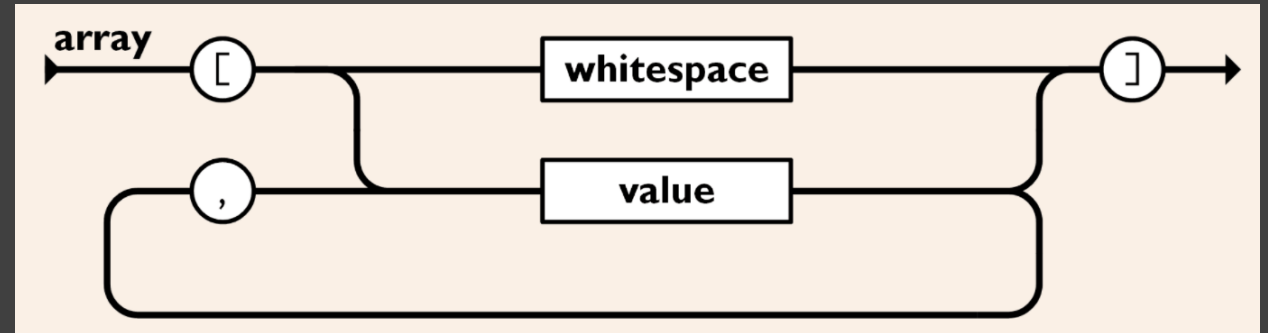
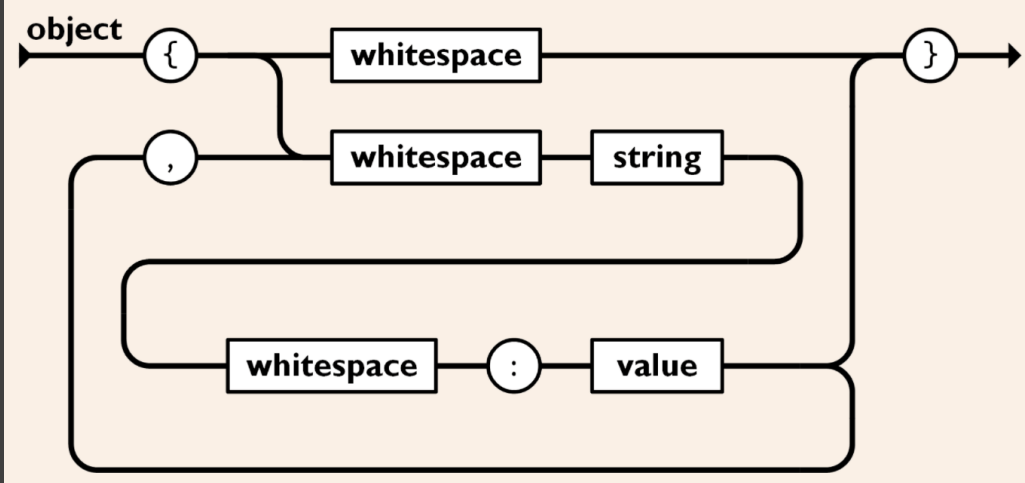
제13강 Messaging (1)

학습 목차

- JSON
- HTTP Request
- REST
- 텔레그램 메시지 주고받기
- 카카오톡 나에게 문자 보내기
- SMS 보내기

JSON(JavaScript Object Notation)

- JavaScript 프로그램의 표준 데이터 교환 형식
- 웹사이트들이 데이터를 JSON 형식으로 제공하는 경우가 많음.
- 파이썬 리스트 및 딕셔너리와 “거의” 동일
 - 큰따옴표만 허용, true, false,



JSON 데이터 문자열 처리

```
import json

json_data_string = '[1,"Suji",true,false, 3.14]'
data_list = json.loads(json_data_string)

data_list.append('Rose')
json.dumps(data_list)
```

JSON 데이터 파일 처리

```
import json

suji_data = \
{
    "City" : "Seoul",
    "Name" : "Suji",
    "Age" : 22,
    "Height" : 165.5
}

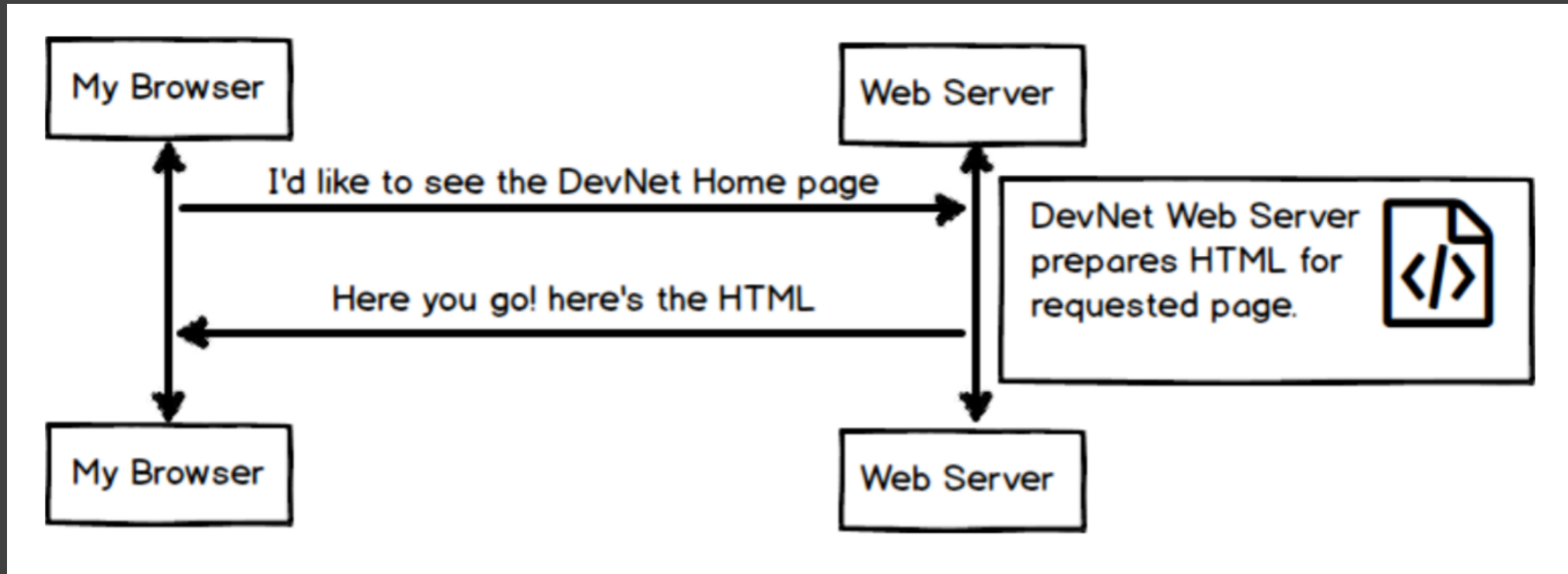
with open('sujidata.json', 'w') as f:
    json.dump(suji_data, f)

with open('sujidata.json') as f:
    data = json.load(f)

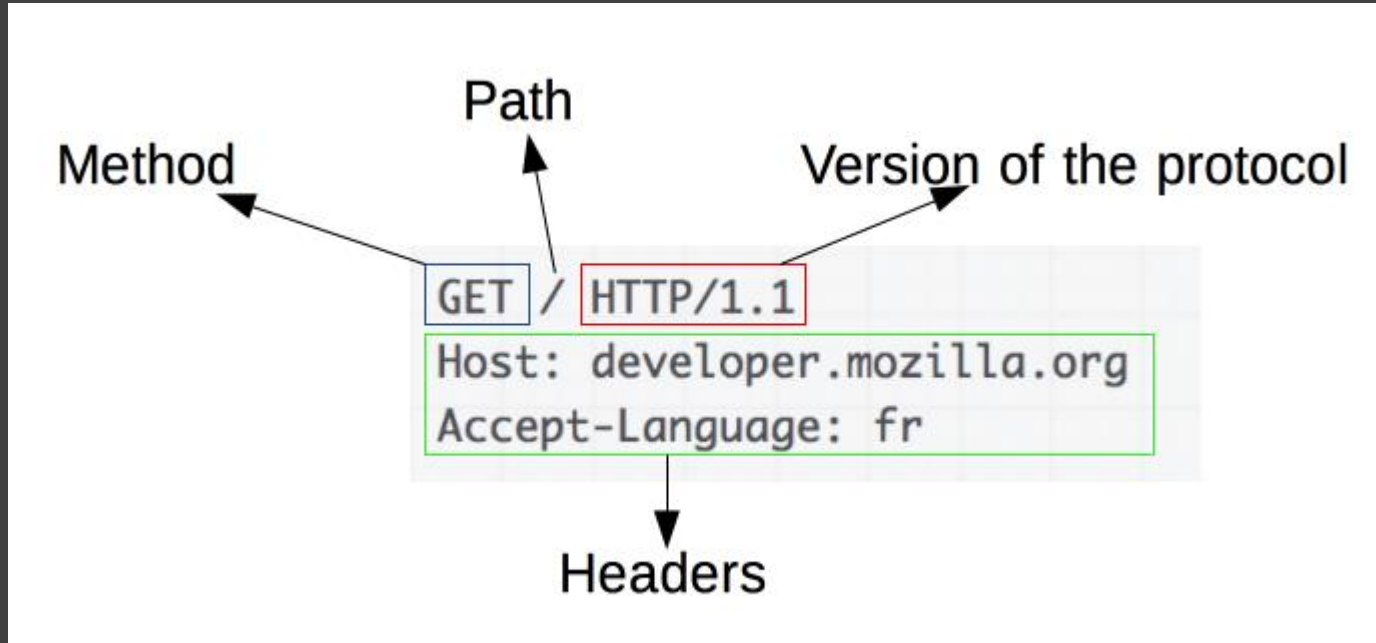
data
```

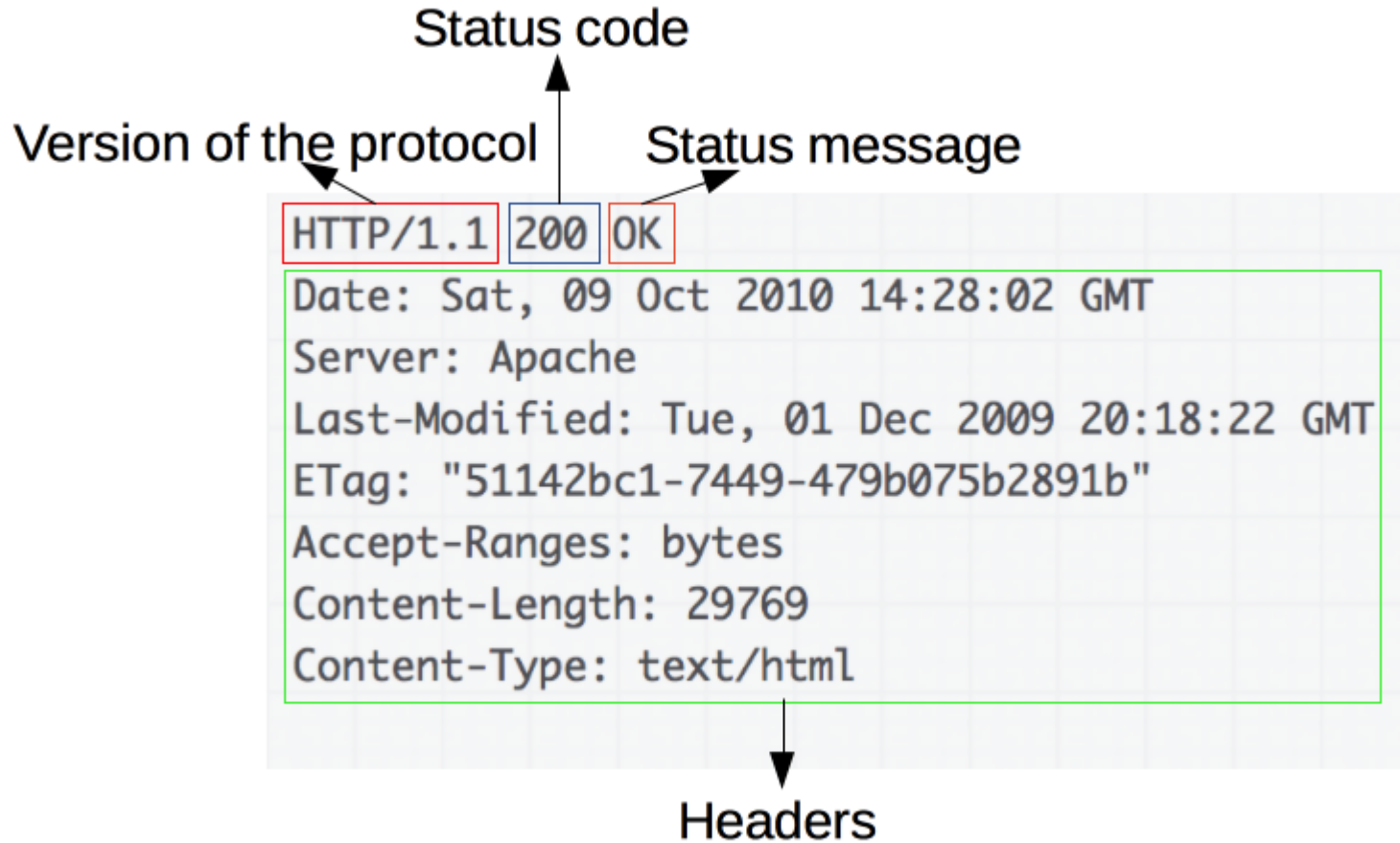
HTTP(HyperText Transfer Protocol)

- HTML 문서를 주고 받는 통신 규약
- 요청(request)를 하면 응답(response)을 받음.



요청 - 서버에 있는 리소스에 대한 처리 요청





요청(request)의 종류

- **GET**
 - 서버에 있는 리소스를 받아옴.
- **POST**
 - 서버에 리소스를 추가.
 - 동일한 요청은 리소스를 여러 개 만들어 냄.(생성)
- **PUT**
 - 서버에 리소스를 추가하거나 또는 업데이트(수정)
 - 동일한 요청에 대해서 결과가 동일.
- **DELETE**
 - 서버에 있는 리소스의 삭제

requests 모듈 – http library

```
import requests

def print_dict(d):
    for k, v in d.items():
        print(k, ': ', v)

r = requests.get('https://httpbin.org')

r.status_code
r.headers
r.headers['content-type']

r.text
r.encoding
r.url
r.content #binary
```

GET : URL로 인자 전달

```
params = {'City': 'Seoul', 'name': 'Suji'}  
r = requests.get('https://httpbin.org/get', params=params)
```

```
r.url  
r.ok  
r.headers  
r.json()
```

URL 에 인자가 표시

POST

```
r = requests.post('https://httpbin.org/post', data={'key': 'value'})  
  
r.url  
# 'https://httpbin.org/post'
```

URL 에 데이터 표시 X, 민감한 정보 제공.

JSON response

```
r = requests.post('https://httpbin.org/post', data = {'key': 'value'})  
  
r.headers  
r.headers['Content-Type']  
r.json()
```

API (Application Programming Interface)

- 소프트웨어 프로그램들이 서로 통신하는 방법
- 주요 API
 - REST
 - Representational State Transfer
 - HTTP 상에서 작동
 - XML, JSON 등의 형식이 이용됨.
 - SOAP
 - Simple Object Access Protocol
 - XML 형식 사용.
 - HTTP, SMTP, TCP, UDP 등을 이용하여 통신.
- 컴퓨터 언어
 - Java, Python, C 등으로 직접 구현.

REST(Representational State Transfer)

- 구성
 - 리소스 : URI 로 표시
 - 행위: HTTP method로 처리 – GET, POST, PUT, DELETE

METHOD	역할
POST	POST를 통해 해당 URI를 요청하면 리소스를 생성합니다.
GET	GET를 통해 해당 리소스를 조회합니다. 리소스를 조회하고 해당 도큐먼트에 대한 자세한 정보를 가져온다.
PUT	PUT를 통해 해당 리소스를 수정합니다.
DELETE	DELETE를 통해 리소스를 삭제합니다.

상태 코드

상태코드	
200	클라이언트의 요청을 정상적으로 수행함
201	클라이언트가 어떠한 리소스 생성을 요청, 해당 리소스가 성공적으로 생성됨(POST를 통한 리소스 생성 작업 시)
상태코드	
400	클라이언트의 요청이 부적절 할 경우 사용하는 응답 코드
401	클라이언트가 인증되지 않은 상태에서 보호된 리소스를 요청했을 때 사용하는 응답 코드
	(로그인 하지 않은 유저가 로그인 했을 때, 요청 가능한 리소스를 요청했을 때)
403	유저 인증상태와 관계 없이 응답하고 싶지 않은 리소스를 클라이언트가 요청했을 때 사용하는 응답 코드
	(403 보다는 400이나 404를 사용할 것을 권고. 403 자체가 리소스가 존재한다는 뜻이기 때문에)
405	클라이언트가 요청한 리소스에서는 사용 불가능한 Method를 이용했을 경우 사용하는 응답 코드
상태코드	
301	클라이언트가 요청한 리소스에 대한 URI가 변경 되었을 때 사용하는 응답 코드
	(응답 시 Location header에 변경된 URI를 적어 줘야 합니다.)
500	서버에 문제가 있을 경우 사용하는 응답 코드

Telegram API

- HTTPS request 로 접근 가능
- Bot API
 - 봇을 이용해서, 사용자와 채팅 가능
 - 봇 : 전화번호 등이 필요없는 특별한 계정
 - 텔레그램 서버와 통신을 위해 복잡한 작업을 할 필요없음.
- Telegram API
 - 맞춤형 Telegram client 를 별도로 제작할 수 있음.

Telegram Bot API

The screenshot shows the Telegram Bot API documentation page in a web browser. The browser's address bar shows the URL `core.telegram.org/bots/api`. The page has a yellow header with navigation links in Korean: [홈](#), [자주묻는질문](#), [앱](#), [API](#), [프로토콜](#), and [스키마](#). A Twitter icon and the text [트위터](#) are on the right. Below the header, a breadcrumb trail shows [Telegram Bots](#) > [Telegram Bot API](#). The main content area is titled

Telegram Bot API

 and contains the text: "The Bot API is an HTTP-based interface created for developers keen on building bots for Telegram. To learn how to create and set up a bot, please consult our [Introduction to Bots](#) and [Bot FAQ](#)." Below this is a section titled

Recent changes

 with the text: "Subscribe to [@BotNews](#) to be the first to know about the latest updates and join the discussion in [@BotTalk](#)". The date **April 21, 2023** is followed by the sub-header **Bot API 6.7**. A bulleted list of updates follows:

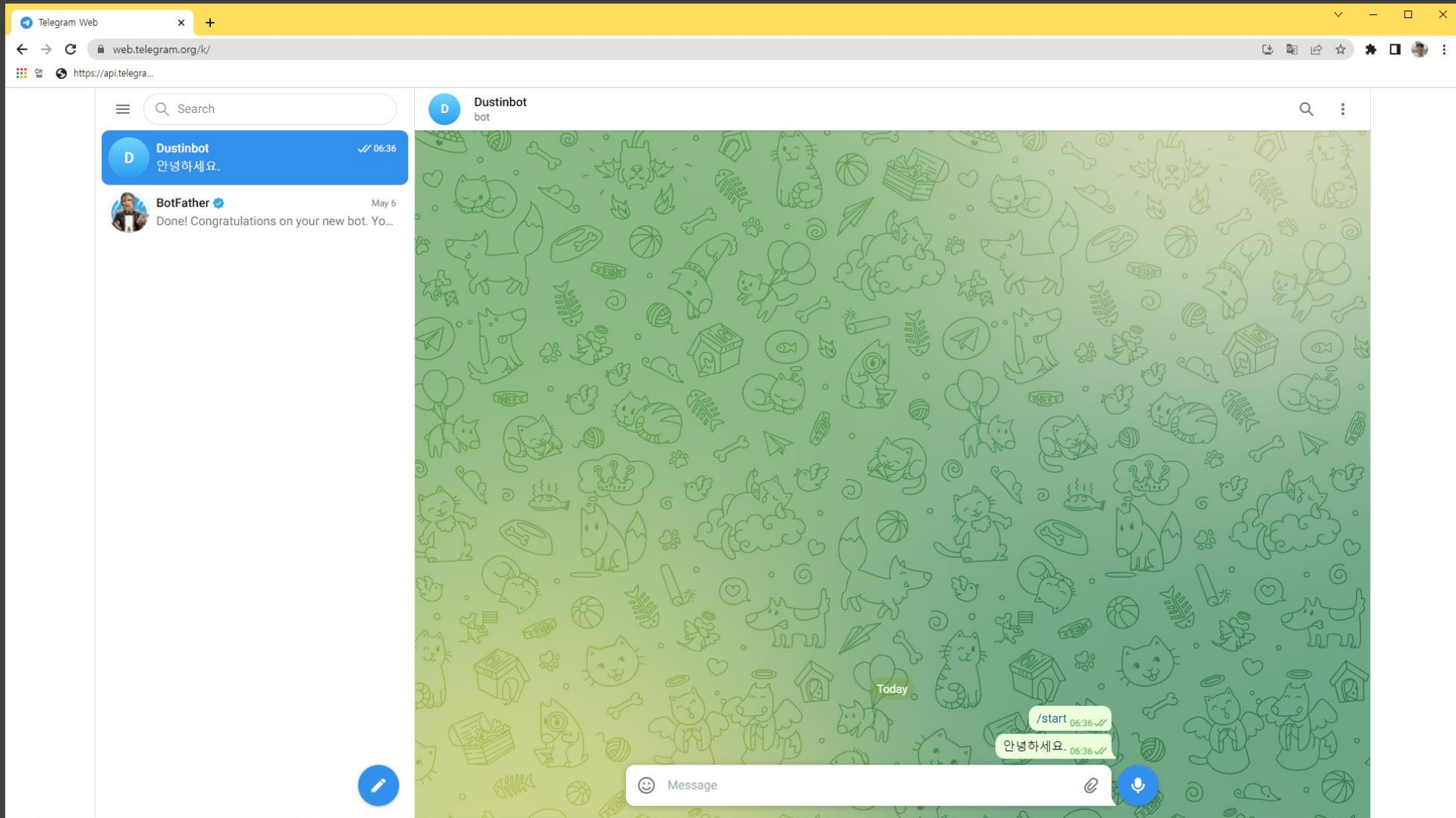
- Added support for launching [Web Apps](#) from inline query results by replacing the parameters `switch_pm_text` and `switch_pm_parameter` of the method [answerInlineQuery](#) with the parameter `button` of type [InlineQueryResultsButton](#).
- Added the field `web_app_name` to the class [WriteAccessAllowed](#).
- Added the field `switch_inline_query_chosen_chat` of the type [SwitchInlineQueryChosenChat](#) to the class [InlineKeyboardButton](#), which allows bots to switch to inline mode in a chosen chat of the given type.
- Added the field `via_chat_folder_invite_link` to the class [ChatMemberUpdated](#).
- Added the ability to set different bot names for different user languages using the method [setMyName](#).
- Added the ability to get the current bot name in the given language as the class [BotName](#) using the method [getMyName](#).
- Added the ability to change bot settings from the bot's profile in official Telegram apps, including the ability to set animated profile photos.

 On the right side of the page, there is a sidebar titled **Recent changes** with a list of links: [April 21, 2023](#), [March 9, 2023](#), [February 3, 2023](#), [Authorizing your bot](#), [Making requests](#), [Using a Local Bot API S...](#), [Getting updates](#), [Available types](#), [Available methods](#), [Updating messages](#), [Stickers](#), [Inline mode](#), [Payments](#), [Telegram Passport](#), and [Games](#).

Telegram API Wrapper

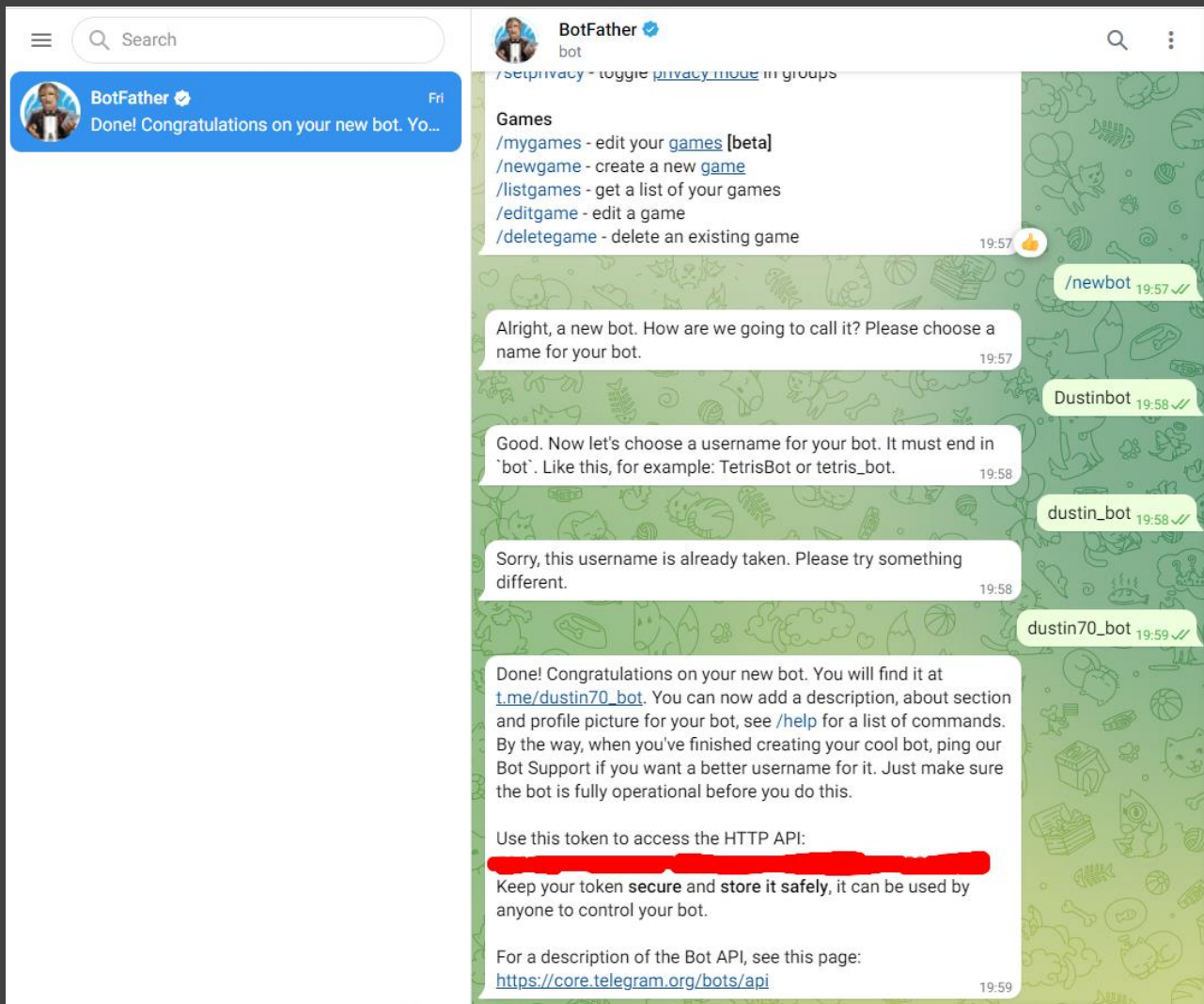
- `pyTelegramBotAPI`
- `python-telegram-bot`

Telegram Web - <https://web.telegram.org/k/>



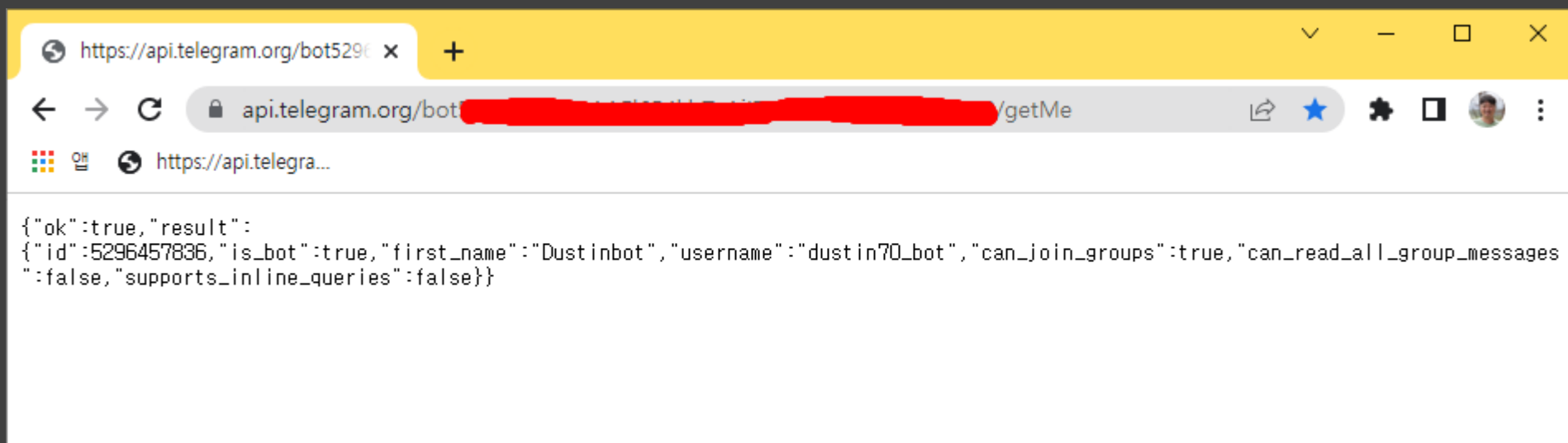
봇 계정 생성

- @BotFather와 대화를 통해서 봇 계정 생성



봇 계정 확인

- <https://api.telegram.org/botxxxxxx/getMe>



getMe

```
import requests
import json

telegram_url = 'https://api.telegram.org/botXXXXXXXX'

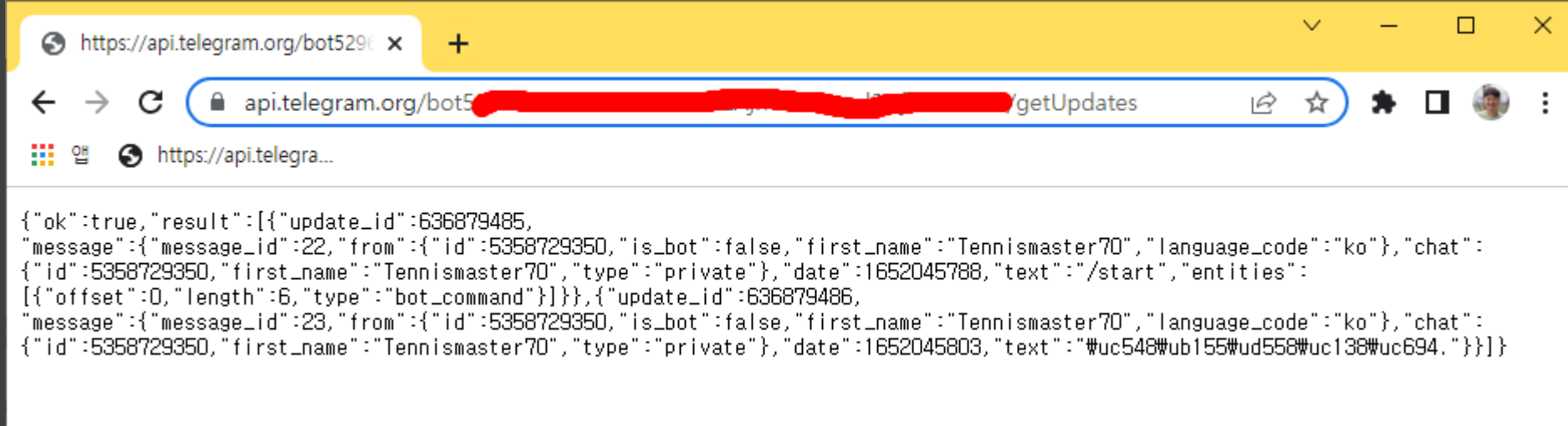
# get information about me, or mybot
r = requests.get(f'{telegram_url}/getMe')
print(r.ok)
print(r.headers)
print(r.headers['content-type'])
print(json.dumps(r.json(), indent=4, ensure_ascii=False))
```

봇을 통해 메시지를 처리하려면?

- 반드시 bot 과 대화를 최소 한번은 START 해야 한다.

메시지 확인 (봇이 받은)

- <https://api.telegram.org/botxxxxxx/getUpdates>



업데이트 확인

```
def get_all_updates():  
    # https://core.telegram.org/bots/api#getupdates  
    r = requests.get(f'{telegram_url}/getUpdates')  
    if r.ok:  
        updates = r.json()['result']  
        for u in updates:  
            # print(u)  
            # print(json.dumps(u, indent=4, ensure_ascii=False))  
            print(u['message']['text'])  
    else:  
        print(f'FAIL : {r.json()}')
```

```
get_all_updates()
```

최신 업데이트만 확인

```
with open('botstatus.json') as f:  
    botstatus = json.load(f)
```

```
{  
    "last_update_id": -1  
}
```

```
def get_new_updates():  
    r = requests.get(f'{telegram_url}/getUpdates',  
                    params={'offset' : botstatus['last_update_id'] + 1})  
    if r.ok:  
        updates = r.json()['result']  
        if updates:  
            for u in updates:  
                print(u['message']['text'])  
            botstatus['last_update_id'] = updates[-1]['update_id']  
            with open('botstatus.json', 'w') as f:  
                json.dump(botstatus, f)  
        else:  
            print(f'FAIL : {r.json()}')
```

```
get_new_updates()
```

json 해석

```
{
  "ok": true,
  "result": [
    {
      "update_id": 296381954,
      "message": {
        "message_id": 1,
        "from": {
          "id": 5358729350,
          "is_bot": false,
          "first_name": "\ub300\ud604",
          "last_name": "\uc774",
          "username": "Tennismaster70",
          "language_code": "ko"
        },
        "chat": {
          "id": 5358729350,
          "first_name": "\ub300\ud604",
          "last_name": "\uc774",
          "username": "Tennismaster70",
          "type": "private"
        },
        "date": 1682769798,
        "text": "/start",
        "entities": [
          {
            "offset": 0,
            "length": 6,
            "type": "bot_command"
          }
        ]
      }
    },
    {
      "update_id": 296381955,
      "message": {
```

텍스트 메시지 수신

```
def show_update(u):
    if 'message' in u: # message data
        msg = u['message']
        chat_id = msg['chat']['id']
        sender_username = msg['from'].get('username', 'NOUSERNAME')
        sender_fullname = msg['from'].get('first_name', '')+' '+msg['from'].get('last_name', '')
        if 'text' in msg:
            text = msg['text']
            print(f'{chat_id} {sender_fullname}({sender_username}) : {text}')
        else:
            print(json.dumps(msg, indent=2, ensure_ascii=False))
```

텍스트 메시지 송신

```
def send_message(chat_id, text):
    r = requests.get(f'{telegram_url}/sendMessage', params={'chat_id' : chat_id, 'text': text})
    if r.ok:
        print(json.dumps(r.json()['result'], indent=4, ensure_ascii=False))
    else:
        print(f'FAIL : {r.json()}')

# send message to master
send_message(6964349173, '안녕하세요. 봇입니다.')
```