

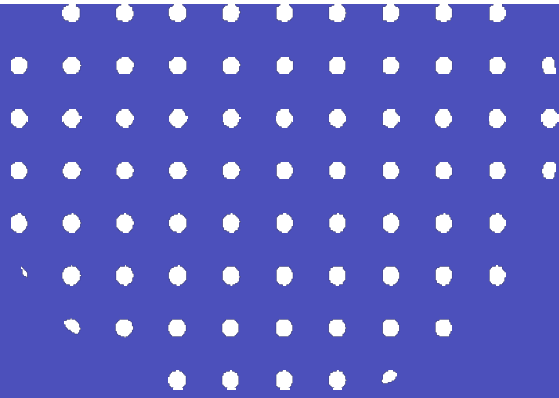
능동적 사고 방식의

# java

강사 박주병

# Part16 최종실습

## 01 chatGPT와 대화하기



# 01

## chatGPT와 대화하기



**N** chatgpt

통합 VIEW 이미지 지식IN 인플루언서 동영상 쇼핑

openai.com > blog

## Introducing ChatGPT - OpenAI [번역보기](#)

We've trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect...

OpenAI

Creating safe artificial general intelligence that benefits all of humanity



OpenAI

Research ▾

Product ▾

Safety

Company ▾

개발에 필요한곳

# Introducing ChatGPT

We've trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests.

[Try ChatGPT ↗](#)

[Read about ChatGPT Plus](#)

웹으로 바로 채팅을 하는곳

Research ▾

Product ▴

Safety

Company

Overview

GPT-4

DALL·E 2

Customer stories

Safety standards

Pricing

Introducing ChatGPT

OpenAI

Research ▾

Product ▾

Safety

Company ▾

# Transforming work and creativity with AI

Our API platform offers our latest models and best practices.

[Get started ↗](#)

[For developers ↓](#)

## Welcome back

Email address

Continue

Don't have an account? [Sign up](#)

OR



Continue with Google



Continue with Microsoft Account

## Verify your phone number



+82 01039007555

Send code

전화번호 인증

Upgrade Help Personal

Join the GPT-4 API waitlist

Sign up

## Welcome to OpenAI

### Start with the basics

#### Quickstart tutorial

Learn by building a quick sample app

#### Examples

Explore some example tasks

⚡ Upgrade

🔍 Help

주변

Personal

박주병

zest7349@gmail.com

Personal

Manage account

View API keys

Invite team

Help

Pricing

Terms & policies

Log out



**ORGANIZATION**[Personal](#) [Settings](#)[Usage](#)[Members](#)[Billing](#)**USER**[API Keys](#)

## API keys

Your secret API keys are listed below. Please note that we do not display your secret key after you generate them.

Do not share your API key with others, or expose it in the browser or other client-side code. To protect the security of your account, OpenAI may also automatically rotate any API key found to have been leaked publicly.

You currently do not have any API keys. Please create one below.

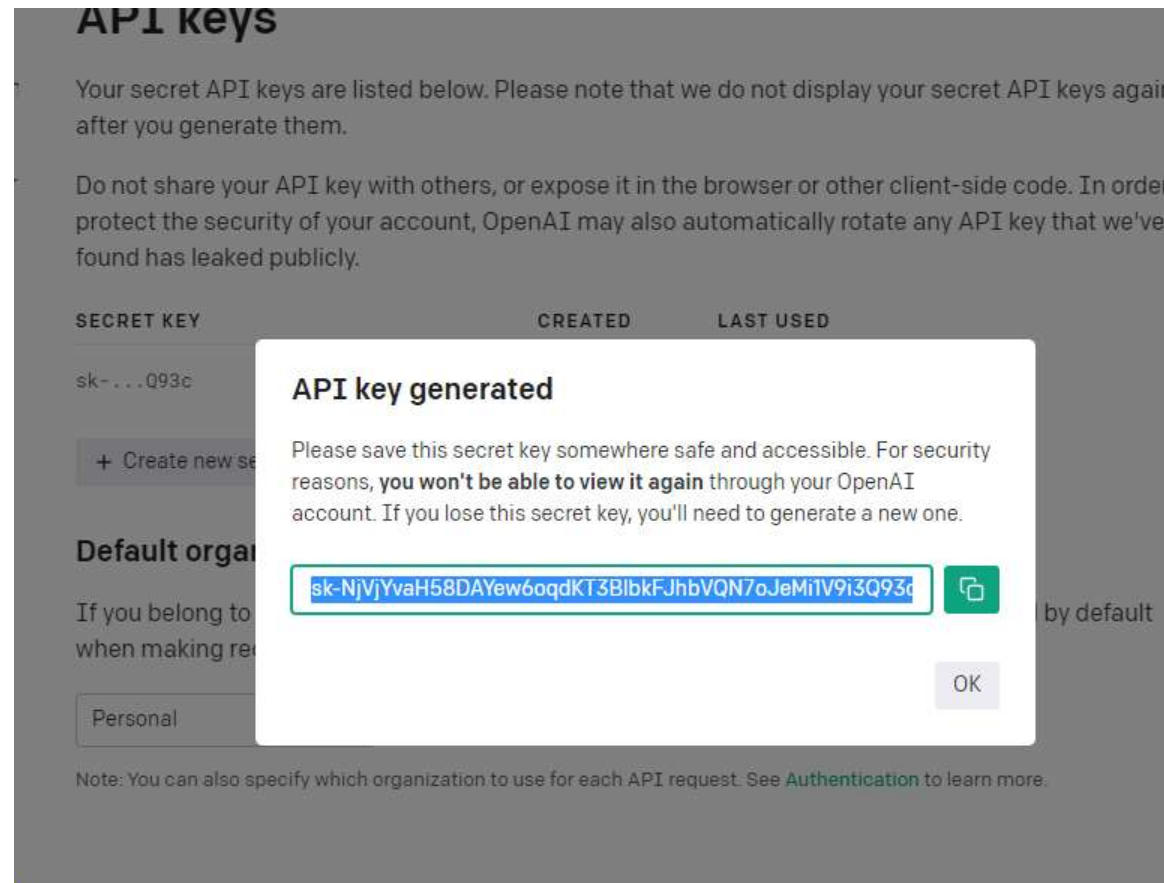
[+ Create new secret key](#)

## Default organization

If you belong to multiple organizations, this setting controls which organization is used by default when making requests with the API keys above.

Personal

Note: You can also specify which organization to use for each API request. See [Authentication](#) to learn more.



키는 반드시 따로 저장해두기(사이트에서 다시 조회 불가능,  
재발급 해야 됨)

 Join the GPT-4 API waitlist

## Welcome to OpenAI

### Start with the basics

#### Quickstart tutorial

Learn by building a quick sample app

### Build an application



**Chat** Beta

Learn how to use chat-based language models

## Models

### Overview

The OpenAI API is powered by a diverse set of models with different capabilities and price points. You can also make limited customizations to our original base models for your specific use case with [fine-tuning](#).

MODELS	DESCRIPTION
<a href="#">GPT-4</a> <a href="#">Limited beta</a>	A set of models that improve on GPT-3.5 and can understand as well as generate natural language or code
<a href="#">GPT-3.5</a>	A set of models that improve on GPT-3 and can understand as well as generate natural language or code
<a href="#">DALL·E</a> <a href="#">Beta</a>	A model that can generate and edit images given a natural language prompt
<a href="#">Whisper</a> <a href="#">Beta</a>	A model that can convert audio into text
<a href="#">Embeddings</a>	A set of models that can convert text into a numerical form
<a href="#">Codex</a> <a href="#">Limited beta</a>	A set of models that can understand and generate code, including translating natural language to code
<a href="#">Moderation</a>	A fine-tuned model that can detect whether text may be sensitive or unsafe
<a href="#">GPT-3</a>	A set of models that can understand and generate natural language

자연어를 이해하는 모델

## GPT-3.5 [↗](#)

GPT-3.5 models can understand and generate natural language or code. Our most capable and cost effective model in the GPT-3.5 family is `gpt-3.5-turbo` which has been optimized for chat but works well for traditional completions tasks as well.

3월 1일 API가 공개된 최신버전

LATEST MODEL	DESCRIPTION	MAX TOKENS	TRAINING DATA
<code>gpt-3.5-turbo</code>	Most capable GPT-3.5 model and optimized for chat at 1/10th the cost of <code>text-davinci-003</code> . Will be updated with our latest model iteration.	4,096 tokens	Up to Sep 2021
<code>gpt-3.5-turbo-0301</code>	Snapshot of <code>gpt-3.5-turbo</code> from March 1st 2023. Unlike <code>gpt-3.5-turbo</code> , this model will not receive updates, and will only be supported for a three month period ending on June 1st 2023.	4,096 tokens	Up to Sep 2021
<code>text-davinci-003</code>	Can do any language task with better quality, longer output, and consistent instruction-following than the <code>curie</code> , <code>babbage</code> , or <code>ada</code> models. Also supports <b>inserting</b> completions within text.	4,097 tokens	Up to Jun 2021
<code>text-davinci-002</code>	Similar capabilities to <code>text-davinci-003</code> but trained with supervised fine-tuning instead of reinforcement learning	4,097 tokens	Up to Jun 2021
<code>code-davinci-002</code>	Optimized for code-completion tasks	8,001 tokens	Up to Jun 2021

We recommend using `gpt-3.5-turbo` over the other GPT-3.5 models because of its lower cost.



주병

gpt-3.5-turbo 인거야?



죄송합니다, GPT-3.5-Turbo는 제가 알고있는 기존 모델 이름 중 하나가 아닙니다. GPT-3.5는 OpenAI에서 공개한 최신의 대규모 언어 모델 중 하나이며, 이 모델은 기존의 GPT 모델을 개선하고 확장한 것입니다. 그러나 GPT-3.5-Turbo라는 특별한 모델에 대해서는 저도 알지 못합니다.



주병

너는 언어모델이 text-davinci-003 이야?



네, 맞습니다. Text-Davinci-003는 OpenAI에서 개발한 GPT-3 언어 모델 계열 중 하나로, 대규모 텍스트 데이터를 기반으로 학습되었습니다. Text-Davinci-003 모델은 GPT-3 모델 시리즈 중 가장 많은 파라미터를 가진 모델 중 하나이며, 다양한 자연어 처리 작업을 수행할 수 있습니다. 저도 Text-Davinci-003 모델 중 하나를 기반으로 작동합니다.



|



## Usage

Below you'll find a summary of API usage for your organization. All dates and times are UTC-based, and data may be delayed up to 5 minutes.

< **March** >

DAILY

CUMULATIVE

Daily usage (USD) ⓘ



Free trial usage



GRANT #	CREDIT GRANTED	EXPIRES (UTC)
---------	----------------	---------------

사용량이 정해져 있으나 개발에 쓰는건 충분

# Json

key 와 value 로 이루어진 데이터 형태이며 현재 서버와의 통신에서 가장 많이 쓰이는 데이터 형태이다.

## 객체(Object) [\[ 편집 \]](#)

객체는 이름/값 쌍의 집합으로, 중괄호{}를 사용한다. 이름은 문자열이기 때문에 반드시 따옴표를 하며, 값은 기본 자료형이다. 각 쌍들은 쉼표(,)로 구별된다. 각 쌍이 나오는 순서는 의미가 없다.

```
{"name2": 50, "name3": "값3", "name1": true}
```

## 배열(Array) [\[ 편집 \]](#)

배열은 대괄호[]로 나타낸다. 배열의 각 요소는 기본 자료형이거나 배열, 객체이다. 각 요소들은 쉼표(,)로 구별된다. 각 요소가 나타나는 순서에 의미가 있다.

```
1 [10, {"v": 20}, [30, "마흔"]]
```



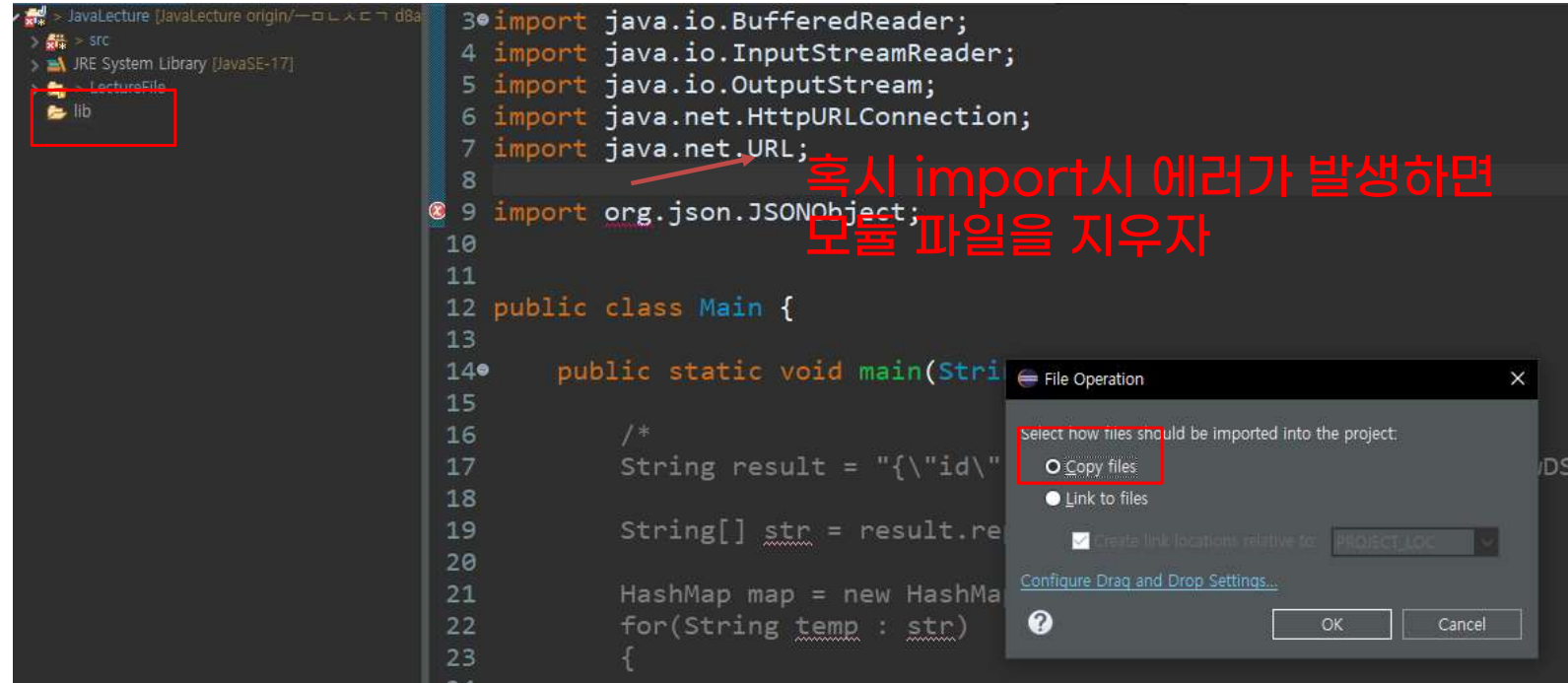
```
{  
  "이름": "홍길동",  
  "나이": 55,  
  "성별": "남",  
  "주소": "서울특별시 양천구 목동",  
  "특기": ["검술", "코딩"],  
  "가족관계": {"#": 2, "아버지": "홍판서", "어머니": "춘심"},  
  "회사": "경기 수원시 팔달구 우만동"  
}
```

json 파싱에 필요한 라이브러리를 다운 받아 추가해야 한다.

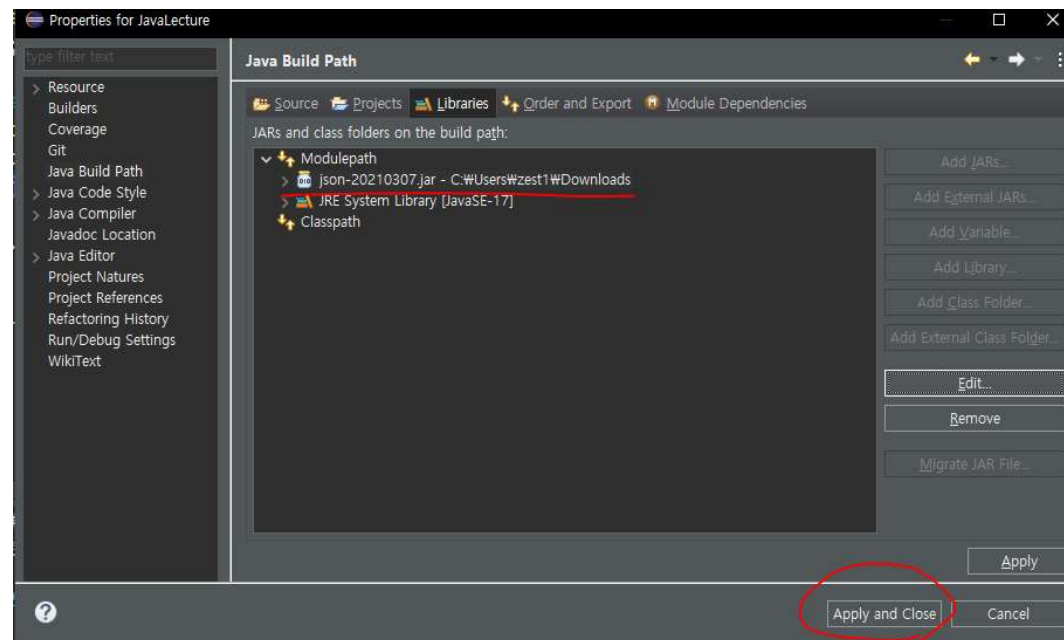
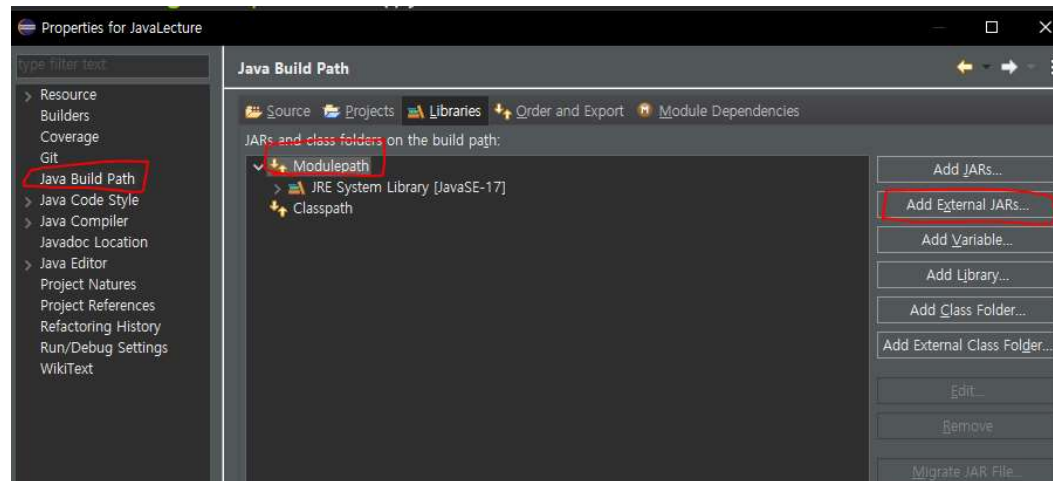
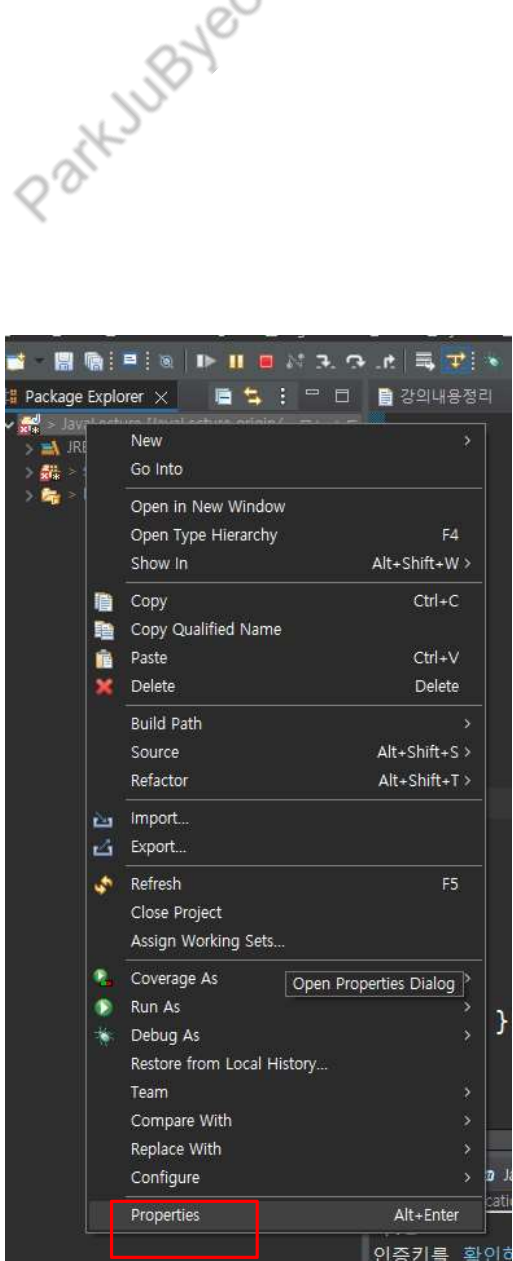
오늘 (1)

JAR json-20210307.jar

License	JSON
Categories	JSON Libraries
Tags	json format
HomePage	<a href="https://github.com/douglascrockford/JSON-js">https://github.com/douglascrockford/JSON-js</a>
Date	Mar 09, 2021
Files	<b>bundle (68 KB)</b> View All
Repositories	Central Hortonworks
Ranking	#89 in MvnRepository (See Top Artifacts) #5 in JSON Libraries
Used By	5,178 artifacts



jar 파일을 프로젝트 폴더로 카피를 하고 라이브러리를 추가한다. 그래야지 나중에 export 하더라도 프로젝트와 함께 라이브러리들도 같이 움직인다.



single-turn tasks without any conversations (such as those previously served by instruction following models like `text-davinci-003`).

An example API call looks as follows:

```
1 # Note: you need to be using OpenAI Python v0.27.0 for the code below to
2 import openai
3
4 openai.ChatCompletion.create(
5     model="gpt-3.5-turbo",
6     messages=[
7         {"role": "system", "content": "You are a helpful assistant."},
8         {"role": "user", "content": "Who won the world series in 2020?"},
9         {"role": "assistant", "content": "The Los Angeles Dodgers won the World"},
10        {"role": "user", "content": "Where was it played?"}
11    ]
12 )
```

대화 내역을 계속 추가 해가면서 보내 줘야지  
AI가 맥락을 이해하면서 대화할수 있다.

The main input is the messages parameter. Messages must be an array of message objects, where each object has a role (either "system", "user", or "assistant") and content (the content of the message). Conversations can be as short as 1 message or fill many pages.

Typically, a conversation is formatted with a system message first, followed by alternating user and assistant messages.

The system message helps set the behavior of the assistant. In the example above, the assistant was instructed with "You are a helpful assistant."

# 기본 소스코드

```
//chatGPT openAI 사이트에서 발급받은 키
String key = "키값을 넣으세요";

//URL 주소를 객체화하는 클래스
URL url=null;

//http 통신을 위한 클래스
URLConnection con=null;

//chatGPT 에게 보낼 메시지 json 형태로 되어있고 모델 정보와 메시지 정보가 들어가 있다.
//chatGPT가 했던 답을 돌려주기 위해서는 role 이 assistant 이어야 한다.
//예제 -> messages\":[{\\"role\\":\\"user\\",\\"content\\":\\"안녕\\"},{\\"role\\":\\"assistant\\",\\"conte
String msg = "{\\"model\\": \"gpt-3.5-turbo\\",\\"messages\\":[{\\"role\\":\\"user\\",\\"content\\":\\""}

//URL을 객체화 한다.
try
{
    url = new URL("https://api.openai.com/v1/chat/completions");
} catch (Exception ex)
{
}

//객체화한 url을 통해 http통신을 위한 객체를 생성한다. 메서드 방식은 POST 이다.
try
{
    con = (URLConnection) url.openConnection();
    con.setRequestMethod("POST");
} catch (Exception ex)
{
}
```



```
//http 통신시 데이터 형태는 json이라고 선언
con.setRequestProperty("Content-Type", "application/json; utf-8");

//발급받은 키값을 넣어준다.
con.setRequestProperty("Authorization", "Bearer "+key);

//chatGPT의 대답이 느릴경우 기다려주는 시간을 설정
con.setRequestProperty("Retry-After", "3600");
//데이터 전송을 위해 true로 설정해야 한다.
con.setDoOutput(true);

//http 통신시 데이터를 서버에 전송하기 위해 스트림을 연다( 스트림이란 데이터가 왔다갔다 하는 통로이다)
try(OutputStream out = con.getOutputStream())
{
    //문자열을 보내기전에 utf-8 형태로 인코딩을 변환하고 바이트로 변경한다.
    byte[] inputBytes = msg.getBytes("utf-8");
    //스트림으로 바이트로된 메시지를 전달한다.
    out.write(inputBytes,0,inputBytes.length);

    //바로 응답 요청을 보내면 chatGPT에서 에러를 보내온다. 2초간 기다려준다..
    Thread.sleep(2000);

    //chatGPT로 부터 응답코드를 받아온다 200이면 정상이다.에러가 발생하면 예외가 발생된다.
    con.getResponseCode();
}catch(Exception ex)
{
}
}
```

```
//chatGPT가 보내온 응답을 받기 위해 InputStream을 열고 버퍼를 통해 받아온다.
try(BufferedReader br = new BufferedReader(new InputStreamReader(con.getInputStream()), "utf-8"))
{
    //응답을 저장할 String 변수이다.
    StringBuilder response = new StringBuilder();
    String responseLine = null;
    //버퍼로 부터 데이터를 가져온다.
    while((responseLine = br.readLine()) != null)
    {
        //공백을 제거 하고 StringBuilder에 이어서 붙인다.
        response.append(responseLine.trim());
    }

    //응답을 출력한다.
    System.out.println(response);

} catch (Exception ex)
{
}
}
```

# chatGPT 실습

1. chatGPT로 부터 받은 응답은 보기 힘든 Json 형태로 되어 있다. Json 데이터에서 응답만을 추출하자.

앞서 설치한 Json 파싱 라이브러리를 이용하자.

```
Debug
jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (2023. 3. 19. 오전 5:05:11 - 오전 5:05:18) [pid: 13148]
"message":{"role":"assistant","content":"하세요! 저는 AI 어시스턴트입니다. 어떤 도움이 필요하신가요?"},""
```

content의 value 가  
실질적인 대답이다.

```
String str = "{\"id\":\"chatcmp1-6vX5cfe1VRa29F7AzS8kBmYjbh4In\",\"object\":\"c
JSONObject obj = new JSONObject("{\"취미\": [\"야구\", \"게임\"]}");
//value가 배열일때 getJSONArray 를 사용한다.
//[\"야구\", \"게임\"] 에서 get(0)은 배열의 첫번째 값을 가져온다는것이다.
String result = obj.getJSONArray(\"취미\").get(0).toString();
System.out.println(result);

JSONArray arr = new JSONArray("[{\"이름\":\"박주병\"},{\"이름\":\"홍길동\"]}");
//value가 배열이고 배열의 요소가 객체일때
result = arr.getJSONObject(1).get(\"이름\").toString();
System.out.println(result);
```

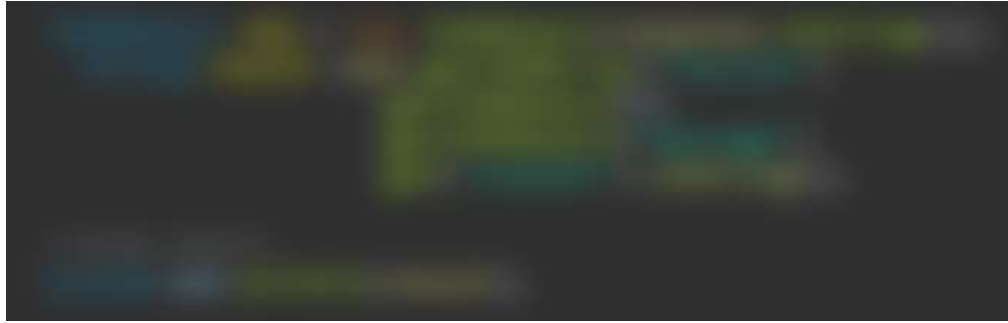
응답 문자열을 변수에 저장해놓고 개발  
진행하면 빠르게 할수 있다.

```
//chatGPT가 보내온 응답을 보기 위해 InputStream
try(BufferedReader br = new BufferedRea
{
    //응답을 저장할 String 변수이다.
    StringBuilder response = new String
    String responseLine = null;
    //버퍼로 부터 데이터를 가져온다.
    while((responseLine = br.readLine())
    {
        //공백을 제거 하고 StringBuilder에 이
        response.append(responseLine.tr
    }

    //응답을 출력한다.
    System.out.println(response);

} catch (Exception ex)
{
    여기에 수정이 필요하다.
}
```





2. 현재 전달하는 메시지가 Json문자열에 “안녕” 이라고 하드코딩 되어 있다. 이것을 Scanner 클래스를 이용하여 키보드로 입력한걸 전달 하도록 하자.

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String message = "안녕";
        System.out.println(message);
    }
}
```

```
String message = scanner.next();
System.out.println(message);
```

3. Main 문에 있는 코드들을 chatGPT 클래스를 생성하여 메서드별로 적절하게 옮기자.

- 생성자(String key) 키값을 받을것
- void send(String msg) 메시지를 매개변수로 받아 전송하는 메서드 구현
- String response() 응답을 가져와 반환하는 메서드 구현
- 동시에 여러객체를 생성하여 통신을 하면 곤란하다. 싱글톤패턴을 이용하여 객체가 무조건 1개만 유지될수 있도록 하자.
- 앞서 만든 send, response 메서드를 활용하여 String chatting(String msg) 를 만들어보자 매개변수로는 전송할 메시지를 주고 반환으로는 chatGPT의 대답을 반환한다.

```
String response = chatGPT.chatting("안녕");  
System.out.println("AI: " + response);
```

4. 현재는 메시지를 한번 보내고 답변을 받으면 프로그램이 종료된다.  
영어로 exit 라고 입력할때 까지 반복문을 이용해 계속 채팅을 해보자.

- 대화가 이어지기 위해서는 이전의 대화내역을 통째로 보내줘야 한다.  
HTTP 통신은 비연결성 이기에 AI가 이전의 대화 내역을 알지 못한다.  
그래서 대화가 자연스러워 질려면 이전 대화 내역들까지 message의 값에 JSON 배열형태로 추가 해줘야 한다.

5. 메시지를 보내기 위해 JSON 문자열을 만드는 방법을 개선하자.

현재는 모델명도 하드코딩 되어 있고 역할이 더 늘어나게 될경우도 생각해서 객체지향적으로 만들어보자.

hint : system,user,assistant 를 열거형으로 정의해서 쓰면 어떨까?  
현재 보내는 메시지의 제이슨 형태는 다음과 같다.

```
model="gpt-3.5-turbo",
messages=[
  {"role": "system", "content": "You are a helpful assistant."},
  {"role": "user", "content": "Who won the world series in 2020?"},
  {"role": "assistant", "content": "The Los Angeles Dodgers won the World Series in 2020."},
  {"role": "user", "content": "Where was it played?"}
```

이는 클래스와 멤버변수로 표현이 가능하다 예를들어

```
class chatGPTJson
{
    String model = "gpt-3.5-turbo";
    List<Message> list = new ArrayList();
}
```

```
class Message
{
    //role를 표현할수 있는 멤버변수
    String content;
}
```

6. 이제 여러분만의 chatGPT 클래스를 다듬어 보세요. 기존의 소스코드보다 더 나은 방향으로 발전시켜 보세요

# THANK YOU

강사 박주병