

개발형 코딩 테스트

- 정해진 목적에 따라서 동작하는 완성된 프로그램을 개발하는 것을 요구하는 코딩 테스트 유형입니다.



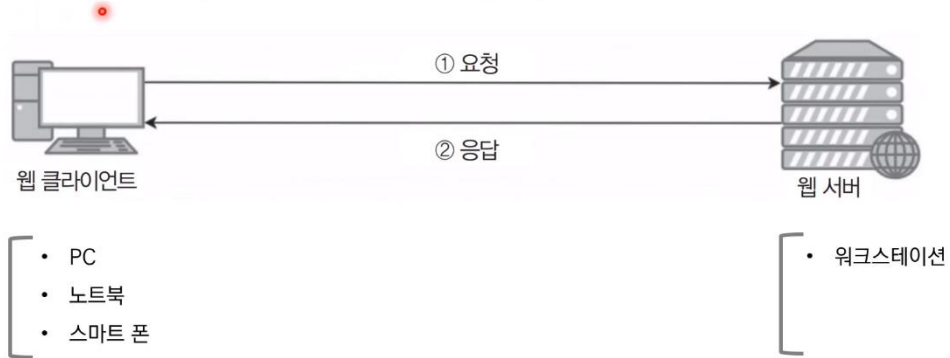
- 일부 기업은 해커톤을 통해 채용을 진행합니다.
 - 해커톤(Hackathon)이란 단기간에 아이디어를 제품화하는 프로젝트 이벤트입니다.
 - 대개 1~2일 정도 진행되며 다수의 해커톤이 대회 형식을 빌려 해커톤이 끝나면 만든 프로그램을 시연하고 발표한 다음 채점을 진행합니다.

개발형 코딩 테스트

- 개발형 코딩 테스트는 분야에 따라 상세 요구사항이 다를 수 있습니다.
 - 예시 1) 모바일 클라이언트 개발: 안드로이드, iOS 앱 개발
 - 예시 2) 웹 서버 개발: 스프링(Spring), 장고(Django) 등의 서버 개발 프레임워크 활용
- 하지만 분야에 상관없이 꼭 알아야 하는 개념과 도구에 대해서 학습할 필요가 있습니다.
 - 서버, 클라이언트, JSON, REST API, ...

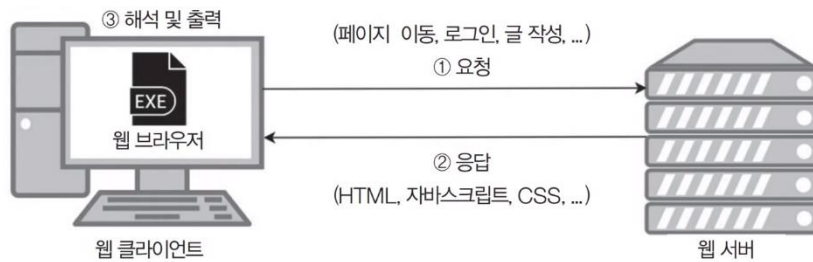
서버와 클라이언트

- 클라이언트가 요청(Request)을 보내면 서버가 응답(Response)합니다.



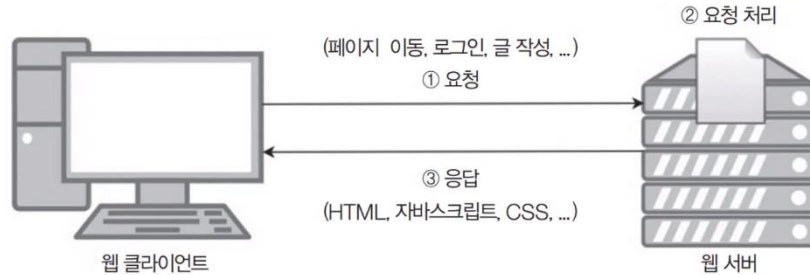
클라이언트(Client) = 고객

- 서버로 요청(Request)을 보내고 응답(Response)이 도착할 때까지 기다립니다.
- 서버로부터 응답을 받은 뒤에는 서버의 응답을 화면에 출력합니다.
 - 예시 1)** 웹 브라우저: 서버로부터 받은 HTML, CSS 코드를 화면에 적절한 형태로 출력합니다.
 - 예시 2)** 게임 앱: 서버로부터 받은 경험치, 친구 귓속말 정보 등을 화면에 적절한 형태로 출력합니다.



서버(Server) = 서비스 제공자

- 클라이언트로부터 받은 요청(Request)을 처리해 응답(Response)을 전송합니다.
 - 예시)** 웹 서버: 로그인 요청을 받아 아이디와 비밀번호가 정확한지 검사하고 그 결과를 응답합니다.



HTTP 개요

- HTTP(HyperText Transfer Protocol)는 웹상에서 데이터를 주고받기 위한 프로토콜을 의미합니다.
 - 보통은 웹 문서(HTML 파일)를 주고받는 데 사용됩니다.
 - 모바일 앱 및 게임 개발 등에서 특정 형식의 데이터를 주고받는 용도로도 사용됩니다.
- 클라이언트는 요청의 목적에 따라서 적절한 HTTP 메서드를 이용해 통신을 진행합니다.
 - 대표적인 **HTTP 메서드**는 다음과 같습니다.

HTTP 메서드	설명	사용 예시
GET	특정한 데이터의 조회를 요청한다.	특정 페이지 접속, 정보 검색
POST	특정한 데이터의 생성을 요청한다.	회원가입, 글쓰기
PUT	특정한 데이터의 수정을 요청한다.	회원 정보 수정
DELETE	특정한 데이터의 삭제를 요청한다.	회원 정보 삭제

파이썬 웹 요청 예제: GET 방식

```
import requests

target = "http://google.com"
response = requests.get(url=target)
print(response.text)
```

실행 결과

```
<!doctype html><html itemscope=" "...
```

개발형 코딩 테스트

나동빈

개발형 코딩 테스트 준비하기

- 다음은 2020 카카오 2차 코딩 테스트 안내문에 쓰여 있던 문장입니다.

“오프라인 코딩 테스트에서는 JSON format의 데이터를 응답하는 REST API를 활용해야 하니, REST API 호출과 JSON format 데이터를 파싱해 활용할 수 있는 parser 코드를 미리 준비해 오시기 바랍니다.”

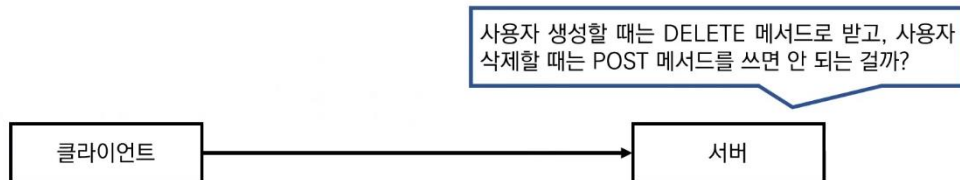
- 개발형 코딩 테스트의 핵심 키워드: REST API, JSON

개발형 코딩 테스트

나동빈

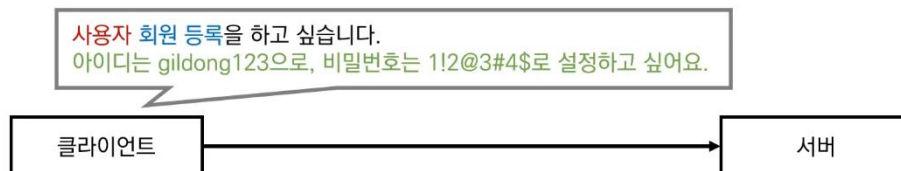
REST의 등장 배경

- HTTP는 GET, POST, PUT, DELETE 등의 다양한 HTTP 메서드를 지원합니다.
- 실제로는 서버가 각 메서드의 기본 설명을 따르지 않아도 프로그램을 개발할 수 있습니다.
- 하지만 저마다 다른 방식으로 개발하면 문제가 될 수 있어 기준이 되는 아키텍처가 필요합니다.



REST 개요

- REST (Representational State Transfer)는 각 자원(Resource)에 대하여 자원의 상태에 대한 정보를 주고받는 개발 방식을 의미합니다.
- REST의 구성 요소
 - **자원(Resource)**: URI를 이용
 - **행위(Verb)**: HTTP 메서드를 이용
 - **표현(Representations)**: 페이로드를 이용



REST 개요

- REST의 구성 요소
 - **자원(Resource)**: URI를 이용
 - **행위(Verb)**: HTTP 메서드를 이용
 - **표현(Representations)**: 페이로드를 이용



REST API란?

- **API (Application Programming Interface)**: 프로그램이 상호작용하기 위한 인터페이스를 의미합니다.
- **REST API**: REST 아키텍처를 따르는 API를 의미합니다.
- **REST API 호출**: REST 방식을 따르고 있는 서버에 특정한 요청을 전송하는 것을 의미합니다.

JSON

- **JSON (JavaScript Object Notation)**: 데이터를 주고받는 데 사용하는 경량의 데이터 형식
- JSON 형식을 따르는 데이터 예시는 다음과 같습니다.

```
{
  "id": "gildong123",
  "password": "1!2@3#4$",
  "age": 30,
  "hobby": ["football", "programming"]
}
```

- JSON 데이터는 **키와 값의 쌍**으로 이루어진 데이터 객체를 저장합니다.

JSON 객체 사용 예제

```
import json

# 사전 자료형(dict) 데이터 선언
user = {
    "id": "gildong",
    "password": "1!2@3#4$",
    "age": 30,
    "hobby": ["football", "programming"]
}

# 파이썬 변수를 JSON 객체로 변환
json_data = json.dumps(user, indent=4)
print(json_data)
```

실행 결과

```
{
  "id": "gildong",
  "password": "1!2@3#4$",
  "age": 30,
  "hobby": [
    "football",
    "programming"
  ]
}
```

JSON 객체 파일 저장 예제

```
import json

# 사전 자료형(dict) 데이터 선언
user = {
    "id": "gildong",
    "password": "1!2@3#4$",
    "age": 30,
    "hobby": ["football", "programming"]
}

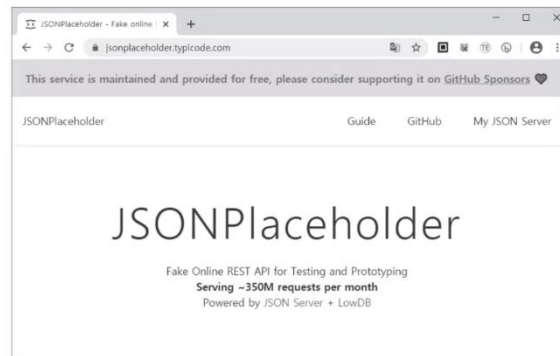
# JSON 데이터로 변환하여 파일로 저장
with open("user.json", "w", encoding="utf-8") as file:
    json_data = json.dump(user, file, indent=4)
```

개발형 코딩 테스트

나동빈

REST API 연습용 서비스

- 목킹(Mocking)이란 어떠한 기능이 있는 것처럼 흉내내어 구현한 것을 의미합니다.
- 가상의 REST API 제공 서비스: <https://jsonplaceholder.typicode.com/>



개발형 코딩 테스트

나동빈

REST API 호출 실습해보기

- API 호출 경로: <https://jsonplaceholder.typicode.com/users/1>
- HTTP 메서드: GET

```
{
  "id": 1,
  "name": "Leanne Graham",
  "username": "Bret",
  "email": "Sincere@april.biz",
  (생략)
}
```

개발형 코딩 테스트

나동빈

```
Replit - Python Online Compiler x https://jsonplaceholder.typicode.com/users/1
{
  "id": 1,
  "name": "Leanne Graham",
  "username": "Bret",
  "email": "Sincere@april.biz",
  "address": {
    "street": "Kulas Light",
    "suite": "Apt. 556",
    "city": "Gwenborough",
    "zipcode": "92998-3874",
    "geo": {
      "lat": "-37.3159",
      "lng": "81.1496"
    }
  },
  "phone": "1-770-736-8031 x56442",
  "website": "hildegard.org",
  "company": {
    "name": "Romaguera-Crona",
    "catchPhrase": "Multi-layered client-server neural-net",
    "bs": "harness real-time e-markets"
  }
}
```

REST API 호출 실습해보기

- API 호출 경로: <https://jsonplaceholder.typicode.com/users>
- HTTP 메서드: GET

```
[
  {
    "id": 1,
    "name": "Leanne Graham",
    "username": "Bret",
    "email": "Sincere@april.biz",
    (생략)
  },
  {
    "id": 2,
    "name": "Ervin Howell",
    "username": "Antonette",
    "email": "Shanna@melissa.tv",
    (생략)
  },
  (생략)
]
```

개발형 코딩 테스트

나동빈

Replit - Python Online Compiler x <https://jsonplaceholder.typicode.com/users> x +

← → ↻ jsonplaceholder.typicode.com/users

```
[
  {
    "id": 1,
    "name": "Leanne Graham",
    "username": "Bret",
    "email": "Sincere@april.biz",
    "address": {
      "street": "Kulas Light",
      "suite": "Apt. 556",
      "city": "Gwenborough",
      "zipcode": "92998-3874",
      "geo": {
        "lat": "-37.3159",
        "lng": "81.1496"
      }
    },
    "phone": "1-770-736-8031 x56442",
    "website": "hildegard.org",
    "company": {
      "name": "Romaguera-Crona",
      "catchPhrase": "Multi-layered client-server neural-net",
      "bs": "harness real-time e-markets"
    }
  },
  {
    "id": 2
```

REST API를 호출하여 회원 정보를 처리하는 예제

```
import requests

# REST API 경로에 접속하여 응답(Response) 데이터 받아오기
target = "https://jsonplaceholder.typicode.com/users"
response = requests.get(url=target)

# 응답(Response) 데이터가 JSON 형식이므로 바로 파이썬 객체로 변환
data = response.json()

# 모든 사용자(user) 정보를 확인하며 이름 정보만 삽입
name_list = []
for user in data:
    name_list.append(user['name'])

print(name_list)
```