api convention

---

1. 엔드포인트 이름 규칙 (명사, 하이픈, 소문자)

- **명사**로 적는다
- 이름이 길어지면 **하이픈**으로 연결한다
- **소문자**로 통일한다

좋은 예시

```
DB 상에서 book이라는 테이블을 사용한다고 가정해보자.
- 특정 책 하나를 가져오는 요청은 여러 책 중에서 하나를 가져오는 것 이므로, GET /books:id
- 책 전체를 가져오는 요청은 여러 책들의 리스트를 가져오는 것 이므로, GET /books
- 책 한권을 등록하는 요청은 여러 책 안에 한 권의 책을 임의의 ID를 부여하면서 추가하는 것 이므로, POST /books
- 특정 책의 내용을 변경하는 요청은 여러 책 안에 있는 한 권의 책을 변경하는 것 이므로, PUT /books:id
```

2. 데이터 송수신 포멧으로는 JSON을 사용

3. HTTP 상태코드

200	클라이언트의 요청이 정상적으로 수행되었음
201	클라이언트가 생성과 관련된 요청을 하였고, 이 요청이 정상적으로 완료됨(POST)
301	클라이언트가 요청한 데이터의 URI가 변경되었을 경우
400	**클라이언트의 요청이 부적절한 경우**
401	클라이언트가 인증과정(로그인 등)을 거치지 않은 상태에서 보안된 데이터에 접근하려고 하는 경우
403	클라이언트가 인증과 관련 없이, 존재는 하지만 응답할 수 없는 데이터를 요청한 경우 (일반적으로 사용 X)
404	클라이언트가 인증과 관련 없이, 응답할 수 없는 데이터를 요청한 경우 (일반적으로 사용 O)
405	클라이언트가 요청한 데이터에 해당 메소드가 존재하지 않거나 사용 불가능 상태인 경우
500	서버에 문제가 있을 경우

4. response 규칙

- 성공했을 경우에는 message를 사용하지 않는다
- 실패했을 경우 status code 와 message를 사용한다

좋은 예시

```
// 데이터가 여러개일때
[
  {
     bookId: 1,
     name: "Harry Potter"
  },
  {
     bookId: 2,
     name: "Avatar"
  }
]
```

```
// 중첩구조로 보내야할때
{
   "data": [
     {
       "bookId": 1,
       "name": "Harry Potter"
     },
     {
       "bookId": 2,
       "name": "Avatar"
     }
   ],
   "totalDocs": 200,
   "nextPageId" : 3
}
```

```
// 좋은예시
// 새로운 책을 추가할때, 성공했을 경우 권장하는 응답 메시지
// POST /books
{
     "bookId": 3,
     "name": "Toy Story"
}
```

실패한 경우

```
{
  "code": "book/not_found",  // 또는 일련의 숫자 또는 코드형태
  "message": "ID가 4인 책을 찾을 수 없습니다."
}
```

5. 표준 UTC를 사용하자

API는 시간 또는 공간에 상관없이 어디에서나 호출될 수 있다. 그렇기 때문에 동일한 날짜 표준 방식을 통해 일관성 있는 출력을 보여줄 필요가 있다. ISO8601은 날짜/시간 데이터의 국제 표준 방식으로, 날짜는 Z 또는 UTC 형식이어야 한다.

```
{
    "createdAt": "2022-11-08T15:00:00Z"
}
```

6. path, query

- path parameter 는 **정제되지 않은 데이터**를 호출
- query parameter 는 좀 더 복잡한 **조건**을 줘서 내가 원하는 **정제된 결과물을 호출**

사용예시

path parameter

GET http://10.58.4.1:8000/products

RESPONSE 200 OK

```
{
    "results" : [
        {
            "id"      : 1,
            "name" : "무농약 깐 생강",
            "price" : "3000원",
        },
        {
            "id"      : 2,
            "name" : "무럭무럭 녹각 영지 키트",
            "price" : "5000원",
        },
        {
            "id"      : 3,
            "name" : "사과",
            "price" : "9000원",
        }
    ]
}
```

GET http://10.58.4.1:8000/products/1

RESPONSE 200 OK

```
{
    "id"      : 1,
    "name" : "무농약 깐 생강",
    "price"  : "3000원",
}
```

GET http://10.58.4.1:8000/products/1/reviews

RESPONSE 200 OK

```
[
    {
        "id"          : 59
        "title"       : "최고의 생강",
        "content"  : "제가 이번주에 ..."
    },
    {
        "id"          : 101
        "title"       : "무농약 깐 생강 재 구매",
        "content"  : "완전 유기농인 느낌이!!"
    }
]
```

>wecode

001

query string

GET /products?price=3000원

RESPONSE

```
{
    "results" : [
        {
            "id"      : 1,
            "name" : "무농약 깐 생강",
            "price" : "3000원",
        },
        {
            "id"      : 3,
            "name" : "사과",
            "price" : "3000원",
        }
    ]
}
```

GET /products?price=3000원&name=사과

RESPONSE

```
{
    "results" : [
        {
            "id"      : 3,
            "name" : "사과",
            "price" : "3000원",
        }
    ]
}
```