

[3강] Postman 실습, 공적 마스크 API 살펴보기

🕒 Created	@Aug 13, 2020 6:04 PM
🏷️ Tags	api postman
📝 메모	아니 JSON 주석 안써지네

Postman을 다루어 보도록 하겠습니다. 그리고 Swagger Hub를 읽으며 실제 사용할 데이터를 살펴보는 과정도 함께 진행 해 보도록 하겠습니다. (공적마스크 API 제공이 중단됨에 따라 본 영상 수강 시에는 타 API를 사용하여 따라해보시는 것을 권장 드립니다.)

[JSONPlaceholder](#)

[URL의 구성](#)

[Postman 실습](#)

[Open API](#)

[공적마스크 API](#)

JSONPlaceholder

해당 사이트에 대해 알아보고

- Fake online REST API
- REST API를 테스트, 프로토타이핑 작업

플젝 진행하다가 front-end 작업해야하는데 back-end 작업이 완료되지 않았을 때 JSON 파일을 이용해 프로토타입을 만드는것

⇒ 오늘은 rest api 요청만 할 예정 (GET, POST, PUT, PATCH, DELETE)

1. 크롬켜서 사이트 접속 ([링크](#))

Resources, Routes에 주목

▼ Resources

Resources

JSONPlaceholder comes with a set of 6 common resources:

/posts	100 posts
/comments	500 comments
/albums	100 albums
/photos	5000 photos
/todos	200 todos
/users	10 users

Note: resources have relations. For example: **posts** have many **comments**, **albums** have many **photos**, ... see below for routes examples.

1. [/posts](#)

- 리스트 형식으로 들어옴 (하이퍼링크 누르면 이동)
- userID, id, title, body 네가지 필드를 가지고 있음을 확인할 수 있다.

2. [/comments](#)

- posts와 비슷하게 userID 대신에 email 필드가 있고 post와의 관계가 필요하므로 postID 필드가 추가됨

3. 대충 이런식으로 뜯어보면 각 속성에 따른 필드가 보임

▼ Routes

Routes

All HTTP methods are supported.

GET	/posts
GET	/posts/1
GET	/posts/1/comments
GET	/comments?postId=1
GET	/posts?userId=1
POST	/posts
PUT	/posts/1
PATCH	/posts/1
DELETE	/posts/1

Note: you can view detailed examples [here](#).

리소스에 접근하려 할 땐 리소스가 가리키는 locator == URL을 알아야함.
(→ URL의 구성 참고)

URL의 구성

아래 네가지 필드로 구성됨

프로토콜 http, https, file 등

호스트 주소 www.naver.com

파일 경로 /home, /index.html

Query parameter ?id=1 &postID=1 (검색, 필터링, 데이터교환시 사용)

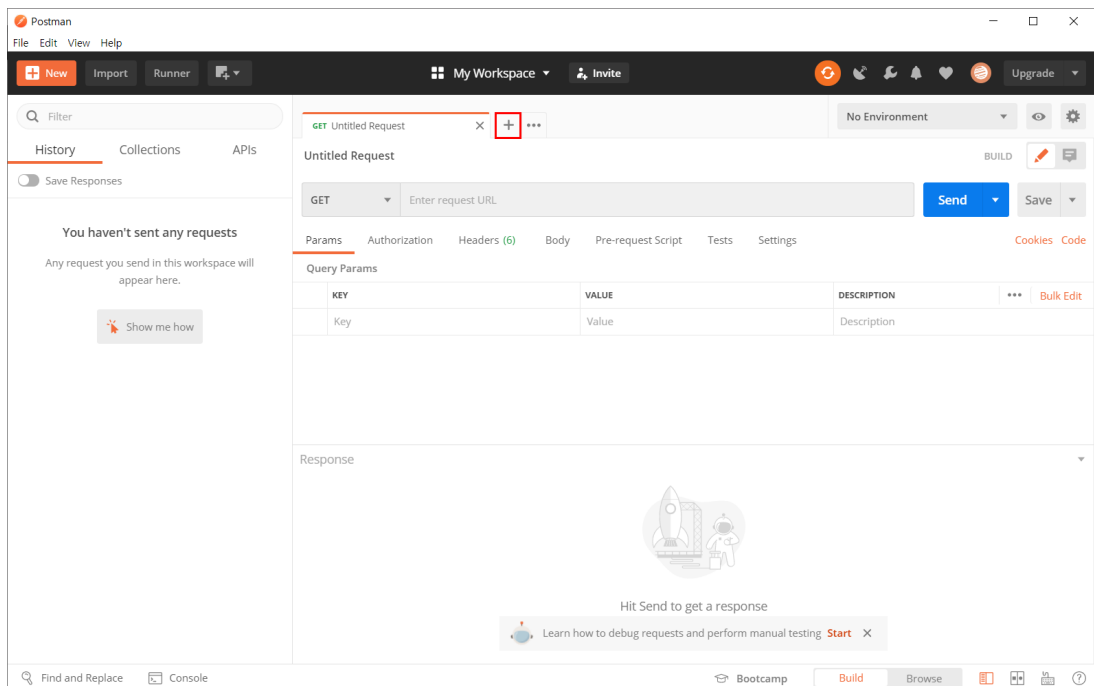
물음표로 시작, ID가 1이니 뭐니...

Postman 실습

해당 사이트에서 제공하는 여섯가지 리소스를 여러가지 HTTP 요청 메소드를 사용해서 조작

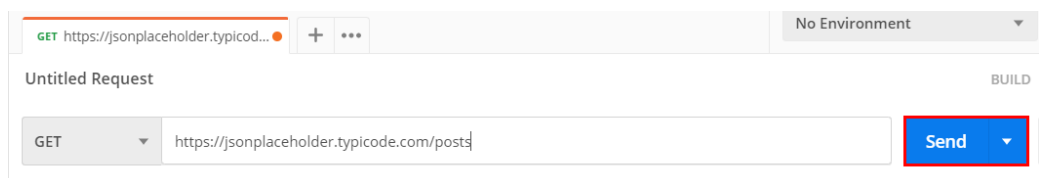
- method 확인

▼ + 버튼을 눌러 Untitled Request 생성



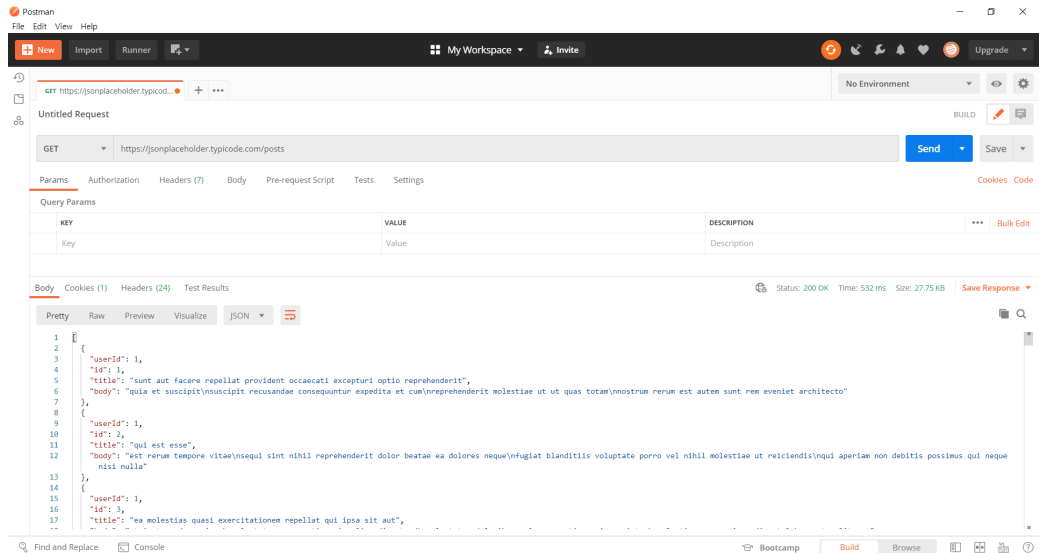
▼ [GET] JSONPlaceholder에서 제공하는 posts 리소스를 요청

1. URL 작성하고 Send를 클릭

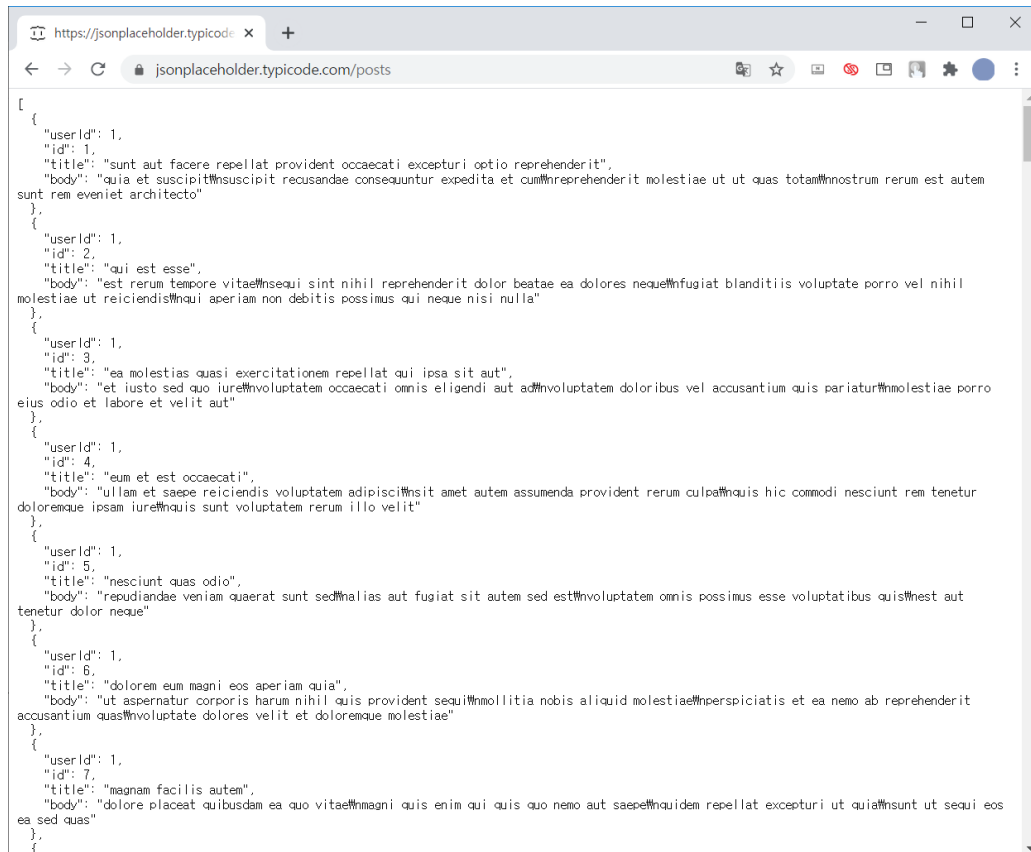


⇒ postman에서 검색한 결과와 웹페이지에서 posts 확인한 결과가 같음을 알 수 있음

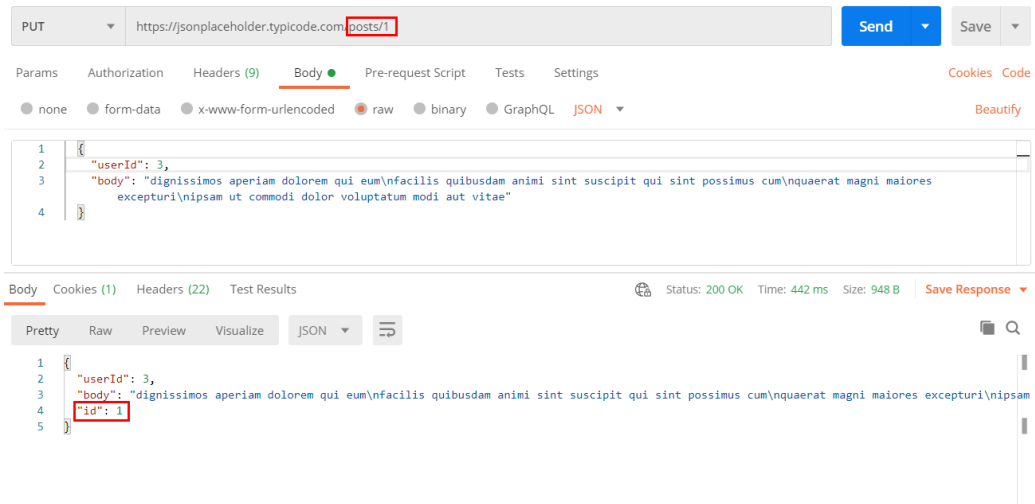
2. postman 에서 검색한 결과



3. jsonplaceholder 웹페이지에서 posts 확인한 결과

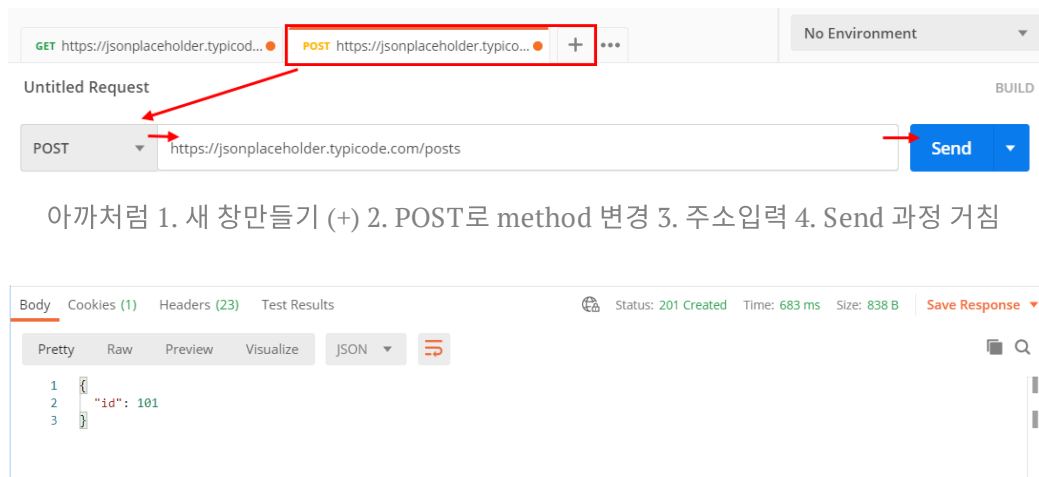


- `/posts/1` 이렇게되면 id가 1인 페이지에 접근할 수 있음.



▼ [POST] get 사용할때와 유사하나 다른점: body의 내용을 실어 보내면 서버가 인식하고 생성과정을 거치게 됨

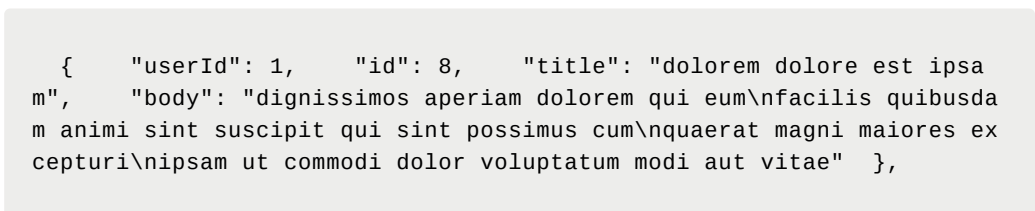
▼ 아무것도 안넣고 post



아까처럼 1. 새 창만들기 (+) 2. POST로 method 변경 3. 주소입력 4. Send 과정 거침

아무것도 안넣고 보내면 자동생성되는 id만 생성됨

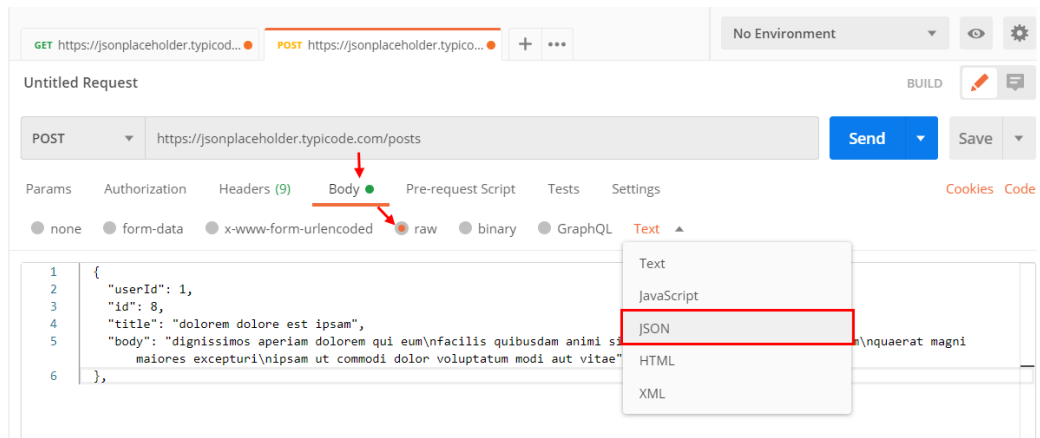
▼ get에서 받은 내용중 id 8번인 글을 가져와서 post



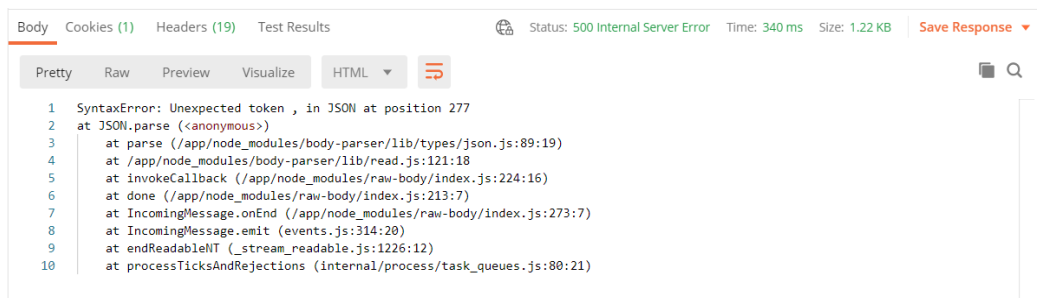
1. body에
2. raw로 가서

3. Text를 JSON으로 바꾸고 내용 추가 후에

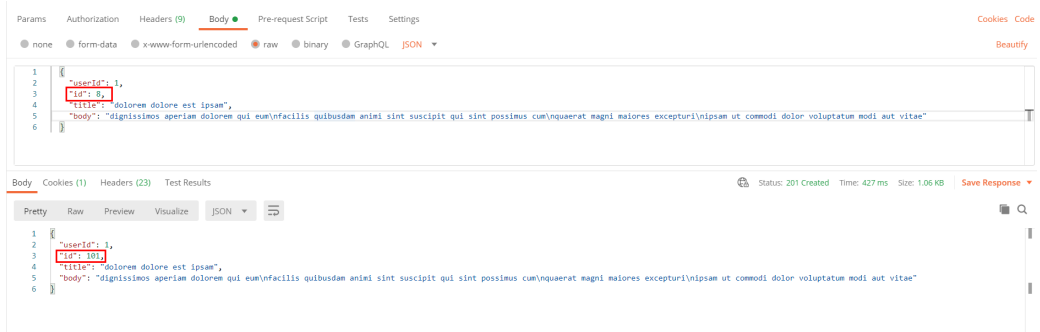
4. Send



과정 (근데 캡처본 끝에 , 콤마 들어가서 에러났음. 마지막 콤마 지워야한다)



} 옆에 , 붙여서 Syntax Error 뜸



진짜 결과. 네모칸친부분 (ID) 빼고 다 똑같이 출력된다

? 근데 아이디도 같이 보냈는데 왜 아이디는 다른가

: 아이디는 고유해야하기때문에 서버에서 임의로 지정함

▼ 다른 필드를 넣어봄 ("comments")

```
1 {
2   "userId": 1,
3   "id": 8,
4   "comments": [],
5   "title": "dolorem dolore est ipsam",
6   "body": "dignissimos aperiam dolorem qui eum\nfacilis quibusdam animi sint suscipit qui sint possimus cum\nquaerat magni maiores excepturi\nipsam ut commodi dolor voluptatum modi aut vitae"
7 }
```

Body Cookies (1) Headers (23) Test Results Status: 201 Created Time: 424 ms Size: 1.08 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "userId": 1,
3   "id": 101,
4   "comments": [],
5   "title": "dolorem dolore est ipsam",
6   "body": "dignissimos aperiam dolorem qui eum\nfacilis quibusdam animi sint suscipit qui sint possimus cum\nquaerat magni maiores excepturi\nipsam ut commodi dolor voluptatum modi aut vitae"
7 }
```

여기는 쓰는대로 출력되는 상태라 에러가 안나는데 정말 사용하는 API 서버의 경우 posts에서 사용되지 않는 필드를 넣을 경우 에러가 나 문제가 발생할 수 있음.

▼ [PUT] 통으로 수정

PUT https://jsonplaceholder.typicode.com/posts/1 Send Save

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "userId": 3,
3   "body": "dignissimos aperiam dolorem qui eum\nfacilis quibusdam animi sint suscipit qui sint possimus cum\nquaerat magni maiores excepturi\nipsam ut commodi dolor voluptatum modi aut vitae"
4 }
```

Body Cookies (1) Headers (22) Test Results Status: 200 OK Time: 462 ms Size: 948 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "userId": 3,
3   "body": "dignissimos aperiam dolorem qui eum\nfacilis quibusdam animi sint suscipit qui sint possimus cum\nquaerat magni maiores excepturi\nipsam ut commodi dolor voluptatum modi aut vitae"
4   "id": 1
5 }
```

input 값에 따라 title이 아예 삭제됨

▼ [PATCH] 일부만 수정 (PUT과 다름)

1. 이 상황에서 "title"만 지우고 patch 하면

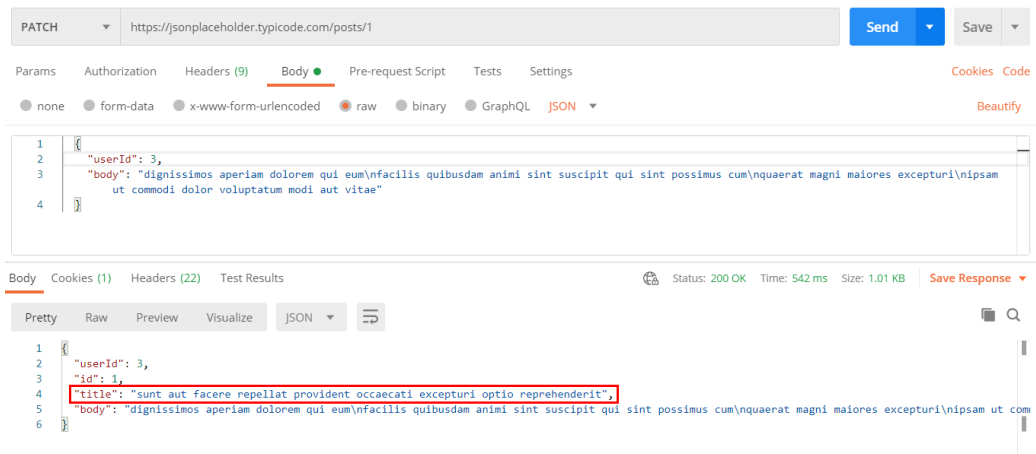
```
1 {
2   "userId": 3,
3   "title": "dolorem dolore est ipsam",
4   "body": "dignissimos aperiam dolorem qui eum\nfacilis quibusdam animi sint suscipit qui sint possimus cum\nquaerat magni maiores excepturi\nipsam ut commodi dolor voluptatum modi aut vitae"
5 }
```

Body Cookies (1) Headers (22) Test Results Status: 200 OK Time: 476 ms Size: 988 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "userId": 3,
3   "title": "dolorem dolore est ipsam",
4   "body": "dignissimos aperiam dolorem qui eum\nfacilis quibusdam animi sint suscipit qui sint possimus cum\nquaerat magni maiores excepturi\nipsam ut commodi dolor voluptatum modi aut vitae"
5   "id": 1
6 }
```

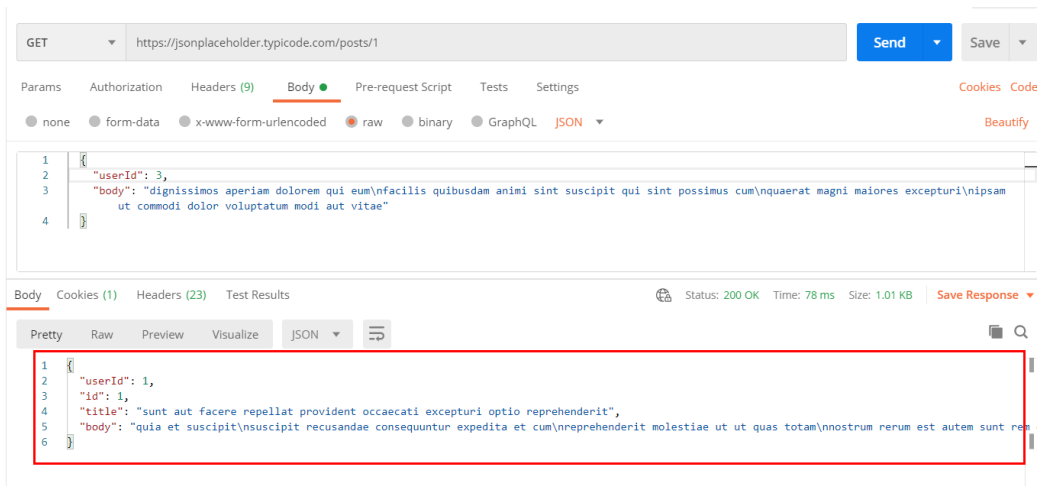

2. 원래 문서에서 입력값부분만 수정된다



이렇게

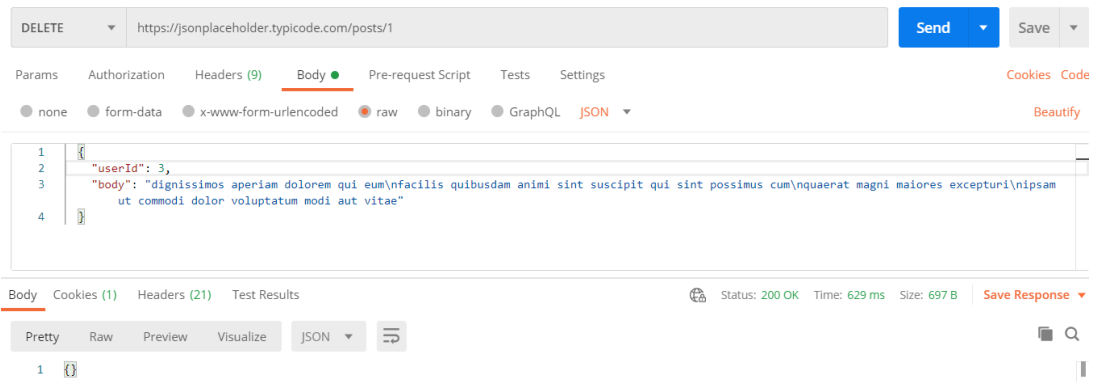
PATCH는 일부분만 수정하는 method이기 때문에 input한 userID와 body만 원래 문서였던 1번문서에서 수정하는 것.

3. 기존의 1번 문서와 비교



원래 1번 문서는 이렇게 생겼다

▼ [DELETE] 삭제

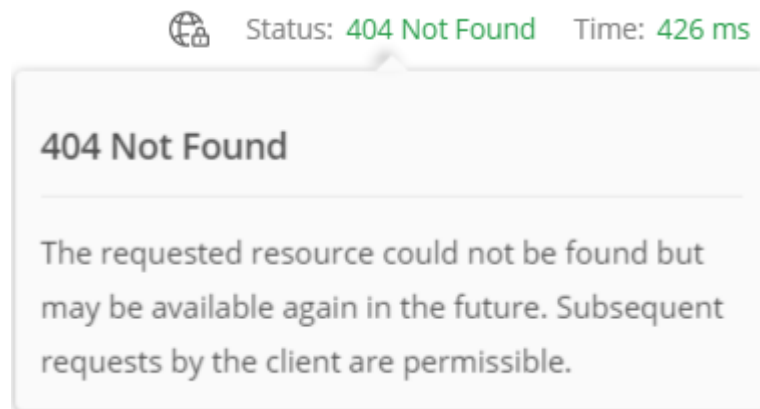


빈 object가 출력됨

- **!?** 변경이 한번만 되고 지속이 안되는데

이 경우에는 JSONPlaceholder 서버가 제공하는 fake api 서버가 진짜 api 서버가 아니라 api서버처럼 작동하는 api라서 바꾼다고 전부 바뀌는게 아니라 일시적으로 보여주는것.

- 에러 확인하는 방법 (400번대면 bad parameter 에러임) 현재는 문제 없음



Open API

| open API가 무엇인지

공적마스크 API

| openAPI의 예인 공적마스크 API 확인 (본격적으로 시작하기 전에)