

## Table of contents

1	GitLab API on GitLab.com	2
1.1	Access Token	2
1.2	API base url	2
1.3	JQ	2
1.4	호출 테스트	2
1.5	Project Id or Url-encoded project path	2
1.5.1	Project Id	2
1.5.2	Url-encoded Project Path	2

# 1 GitLab API on GitLab.com

## 1.1 Access Token

여기에서 api scope를 체크해서 개인 access token을 생성한다. 생성된 토큰은 다시 볼 수 없으므로 생성 직후 어딘가에 잘 보관해야 한다.

## 1.2 API base url

API 문서를 보다 보면, 거의 모든 curl 예제가 다음처럼 `https://gitlab.example.com/api/v4/`을 가지고 작성되어 있다. 당연히 그대로 해서는 gitlab.com에 대해서 사용이 안된다. 그 문서는 self-hosted gitlab과 gitlab.com 모두에 대해서 범용으로 작성되었기 때문에, gitlab.com을 위해서는 그 부분을 `https://gitlab.com/api/v4/`처럼 example.을 제외하고 해야 한다.

```
curl --header "PRIVATE-TOKEN: <your_access_token>"  
↪ https://gitlab.example.com/api/v4/projects/5/pages/domains/www.domain.example
```

## 1.3 JQ

GitLab API의 결과는 그냥 읽으려면 너무 어렵다. 그래서 jq를 이용해서 포매팅해야 한다. `sudo apt install -y jq`해서 설치해두자. 그 후 curl을 가지고 호출할 때 `|jq '.'`을 덧붙이면 포매팅된 json 결과를 볼 수 있다.

## 1.4 호출 테스트

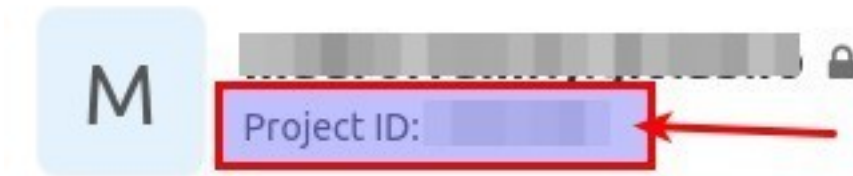
```
curl --header "PRIVATE-TOKEN: 생성된_access_token" "https://gitlab.com/api/v4/users/사용자아이디/projects" | jq '.'  
실행해서 결과를 얻을 수 있다면 일단 성공이다.
```

## 1.5 Project Id or Url-encoded project path

API 문서를 보다 보면, `GET /projects/:id/pages/domains`처럼 The ID or URL-encoded path of the project인 :id를 자주 볼 수 있다.

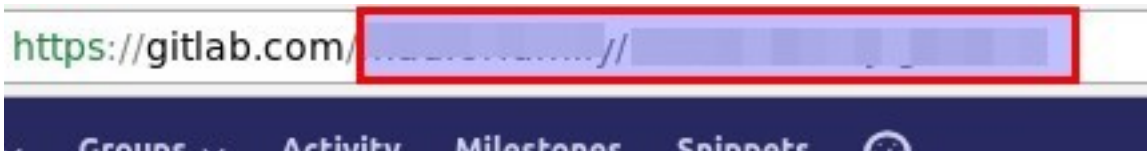
### 1.5.1 Project Id

Project Id는 프로젝트 페이지에서 다음처럼 찾아볼 수 있는 unique한 숫자이다.



### 1.5.2 Url-encoded Project Path

Project path는 GitLab.com에서 프로젝트의 메인 페이지에서 다음처럼 찾을 수 있다. 사용자 프로젝트라면 사용자명/프로젝트명일 것이고, 그룹 프로젝트라면 그룹명/프로젝트명일 것이다.



그런데, 그 Project Path를 Url-encode해야 API 호출할 때 사용할 수 있다. `https://www.urlencoder.org/`로 가서 그 Project Path를 가지고 encode한 결과를 사용해야 한다.