

JWT, Paging, Transaction

7주차

01

피드백

02

JWT

02

JWT, Paging, Transaction

■ 학습 목표

1. 인증과 권한 부여(인가)에 대한 지식 습득
 2. 로그인 유지방법에 대해서 이해 및 적용
 3. DB가 항상 정확하고 일관된 상태를 유지하는 방법
-

03

인증과 권한 부여(인가)

■ 인증이란



■ 인종이란

(이름과 사진 같은 정보로) 네이버 지원자인지 입증하는 과정

홍길동



(식별 가능한 정보로) 서비스에 등록된 유저의 신원을 입증하는 과정

- 권한 부여(인가)란

코테도 보고

화장실도 가고

탕비실도 가고

개발자 사무실도 들어가보고

- 권한 부여(인가)란

ONLY

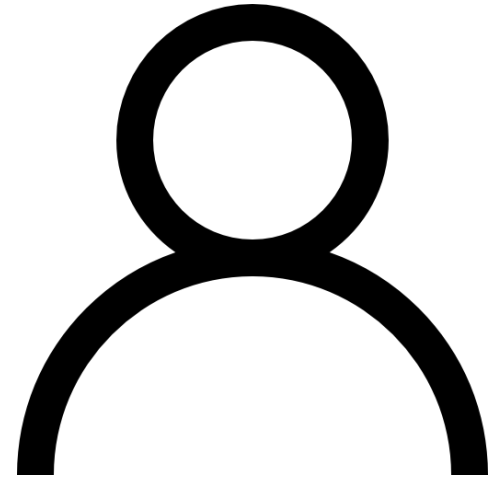


TEST

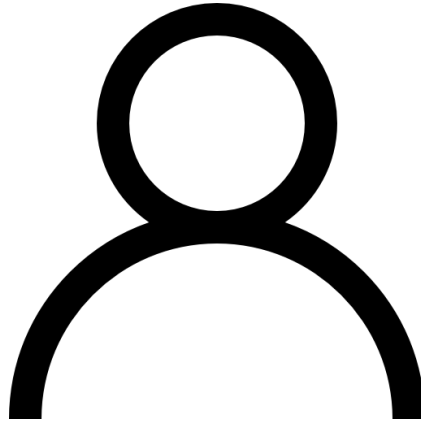
- 권한 부여(인가)란

인증된 사용자에게 대한 자원 접근 권한 확인

■ 웹



■ 웹



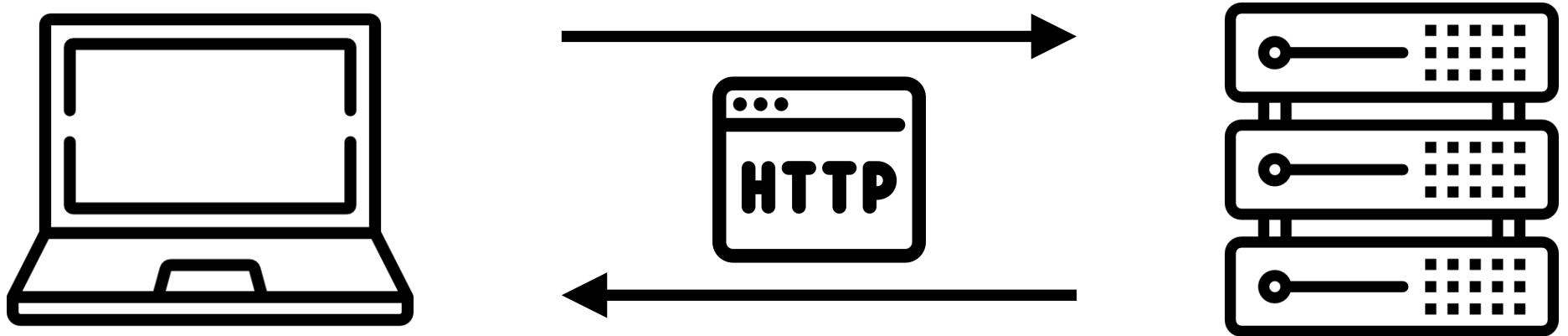
- 인증과 권한 부여(인가)

자원을 적절한/유효한 사용자에게
전달/공개 하기 위한 방법

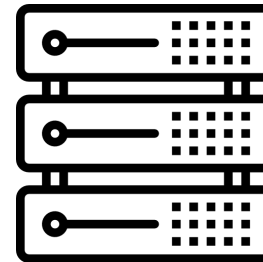
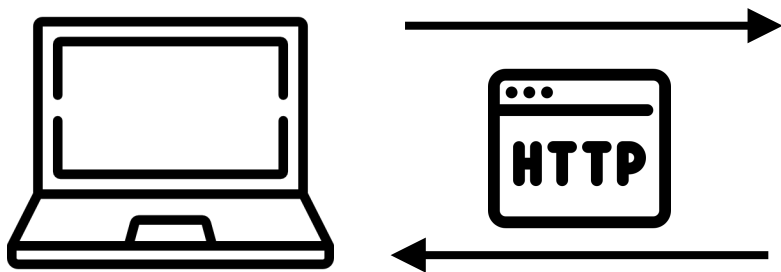
■ 방법

1. 인증하기 - Request Header
 2. 인증 유지하기 - Browser
 3. 안전하게 인증하기 - Server
 4. 효율적으로 인증하기 - Token
 5. 다른 채널을 통해 인증하기 - OAuth
-

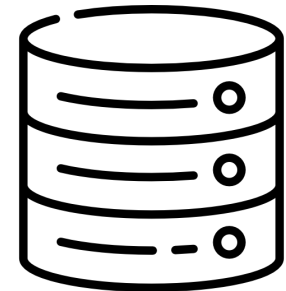
■ 방법 - 사전 지식



■ 1. Request Header 활용하기

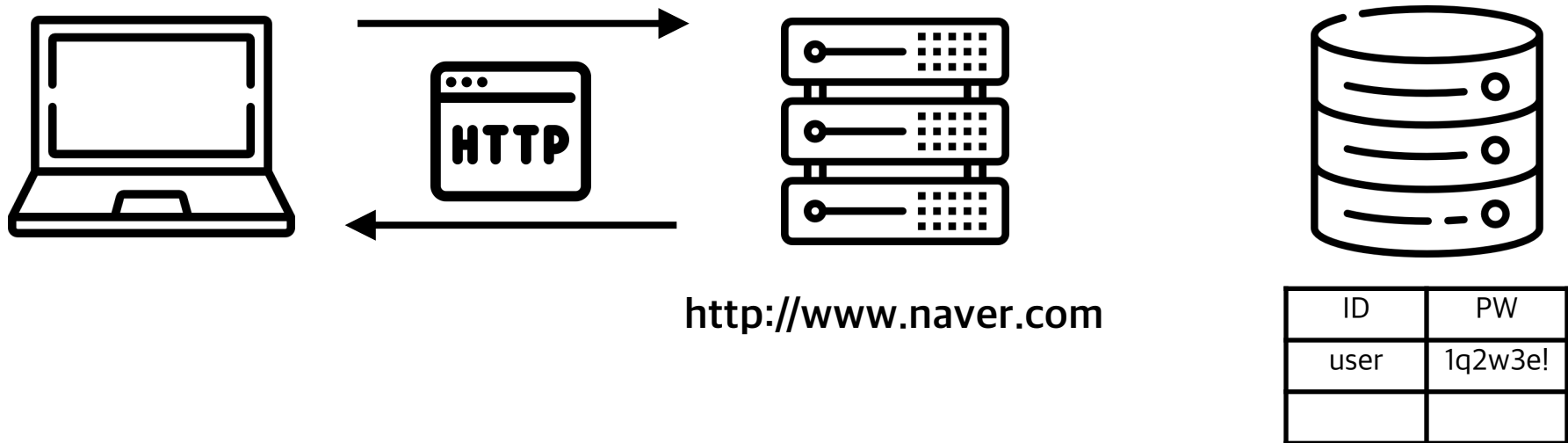


`http://www.naver.com`



ID	PW
user	1q2w3e!

■ 1. Request Header 활용하기



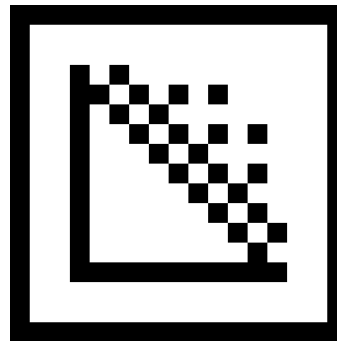
`http://www.naver.com`

`http://user:1q2w3e!@www.naver.com`

- 1. Request Header 활용하기

http://user:1q2w3e!@www.naver.com

user:1q2w3e!



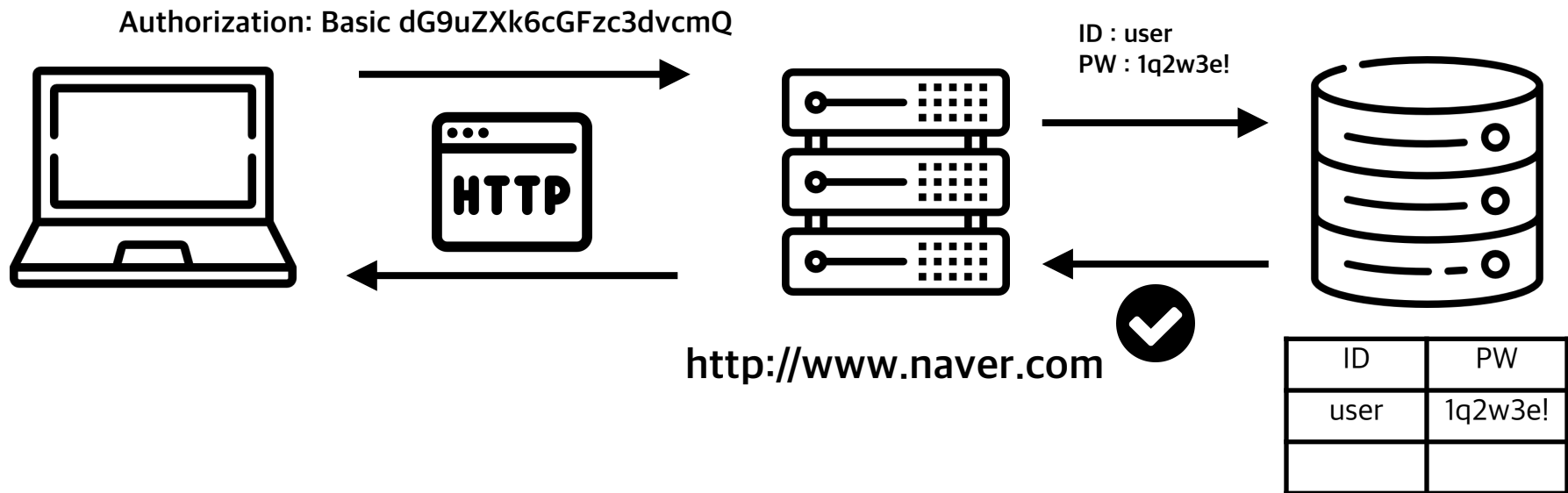
dG9uZXk6cGFzc3dvcmQ

Base 64

요청 헤더

Authorization: Basic dG9uZXk6cGFzc3dvcmQ

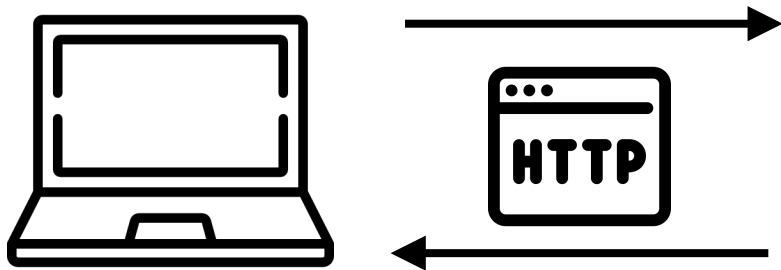
■ 1. Request Header 활용하기



- 1. Request Header만 활용할 시 문제점

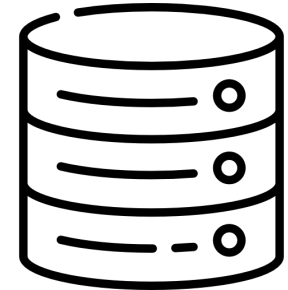
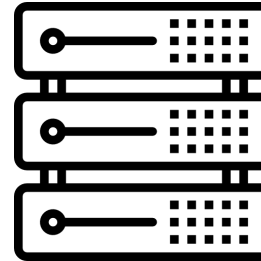
매번 인증해야 한다.

■ 2 : Browser 활용하기



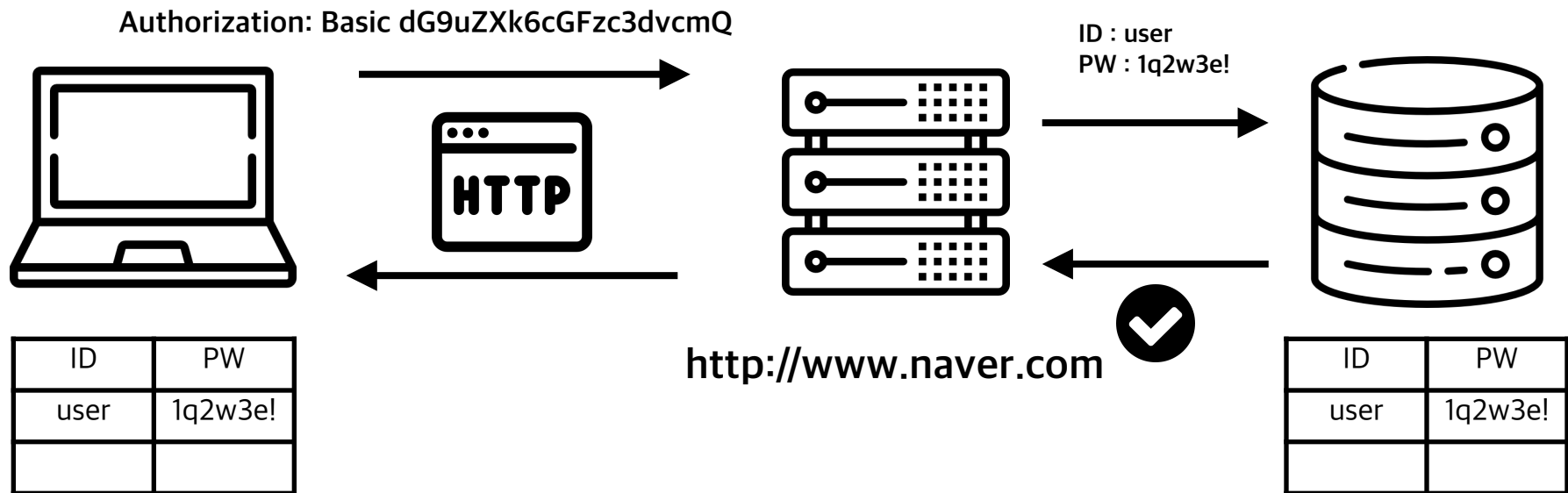
ID	PW
user	1q2w3e!

<http://www.naver.com>

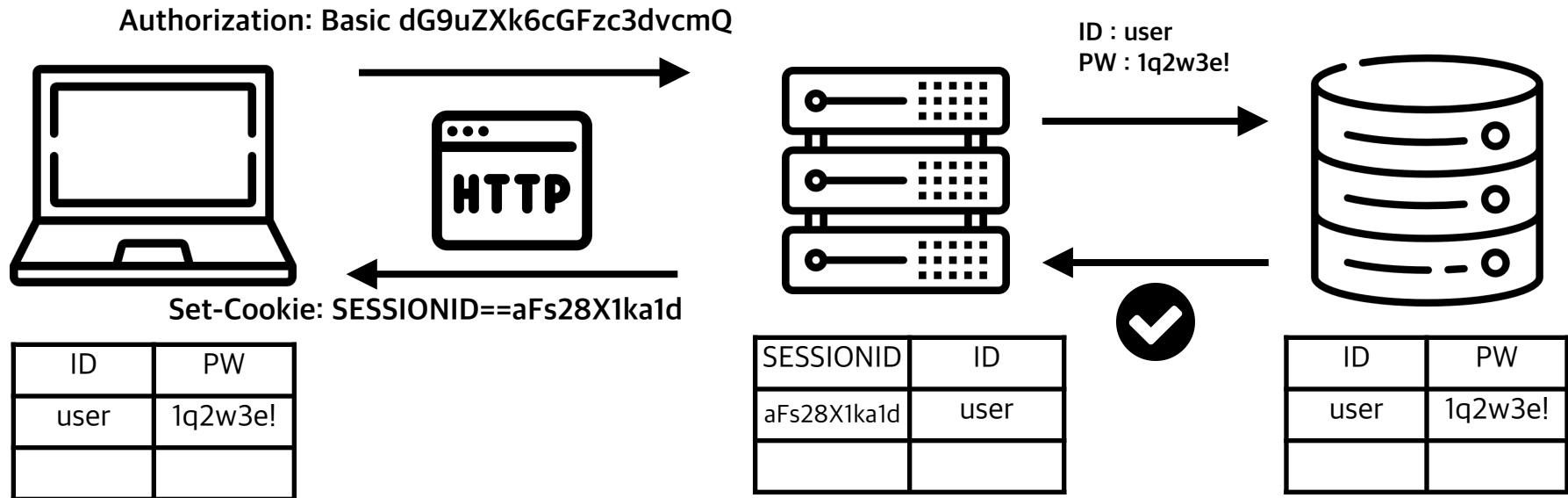


ID	PW
user	1q2w3e!

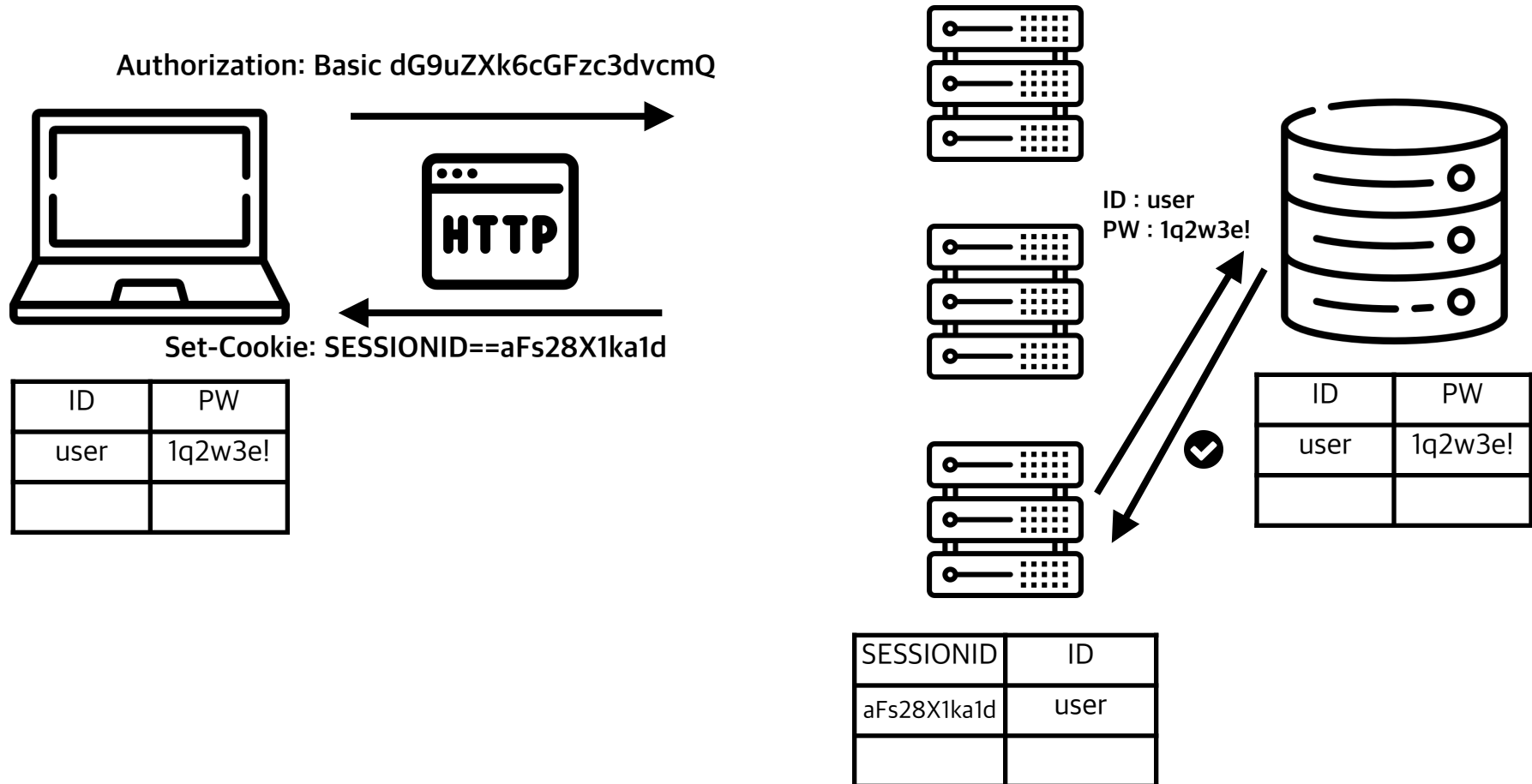
■ 2 : Browser 활용하기



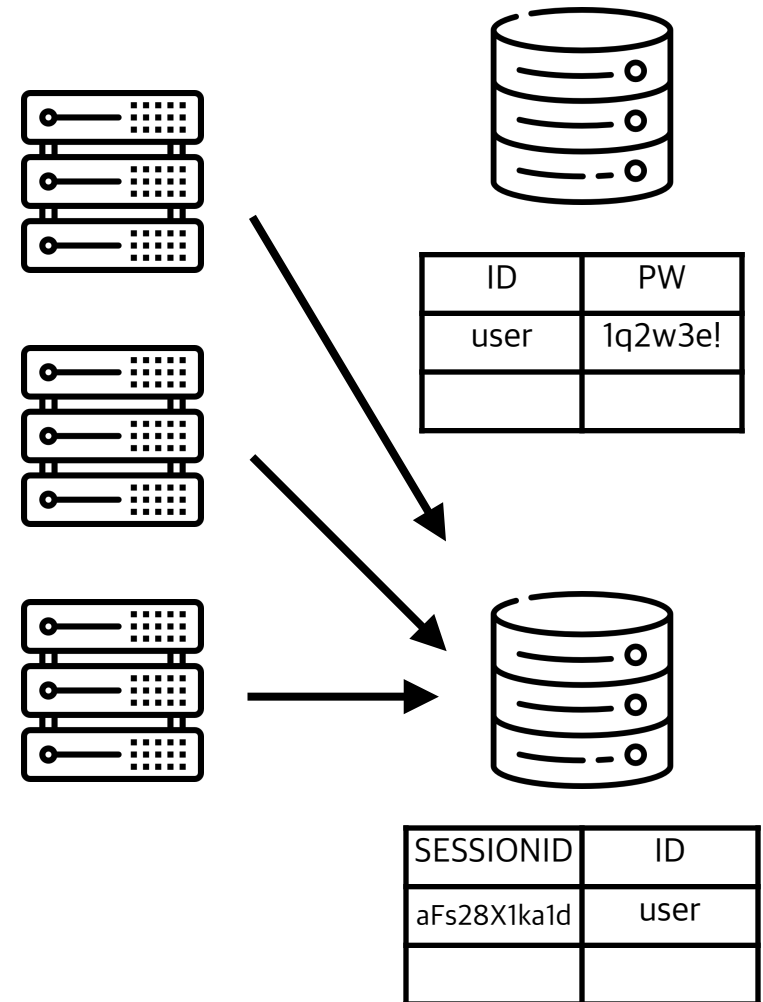
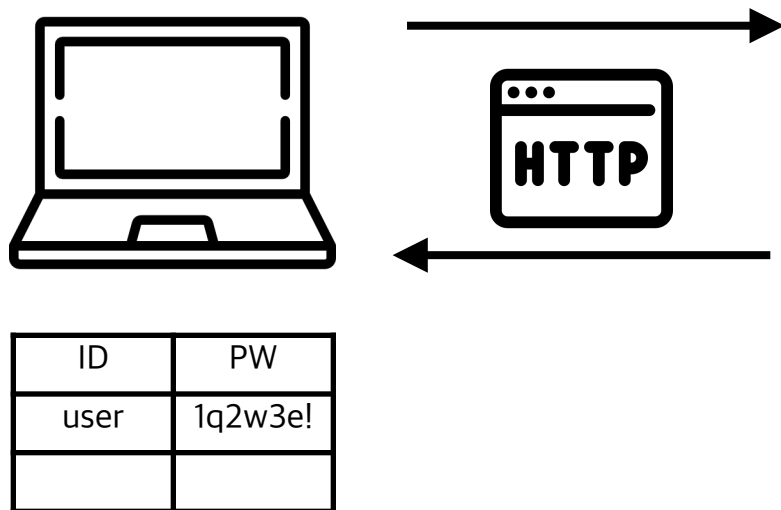
■ 3 : Session 활용하기



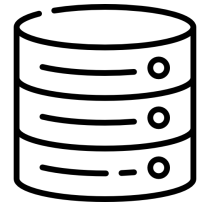
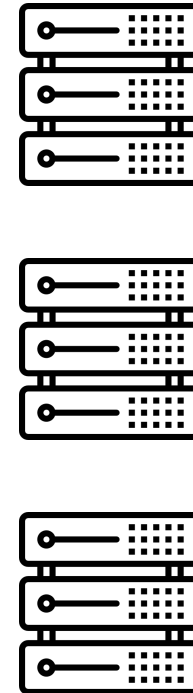
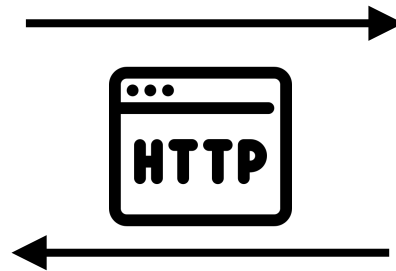
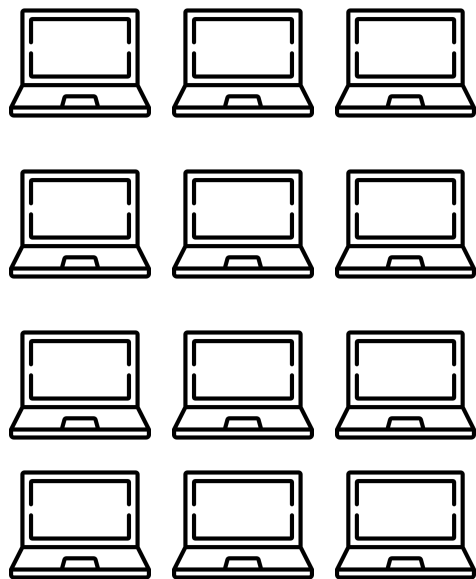
■ 3 : Session 활용하기 - 문제점



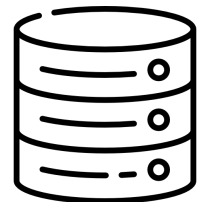
■ 3 : Session 활용하기 - 세션 DB



■ 3 : Session 활용하기 - 세션 DB

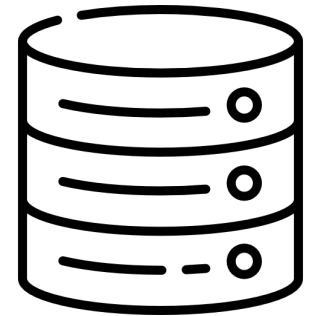
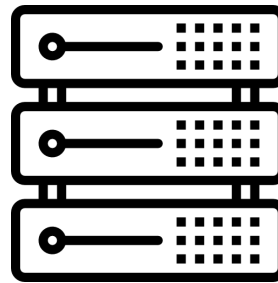
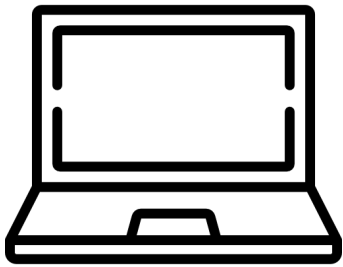


ID	PW
user	1q2w3e!

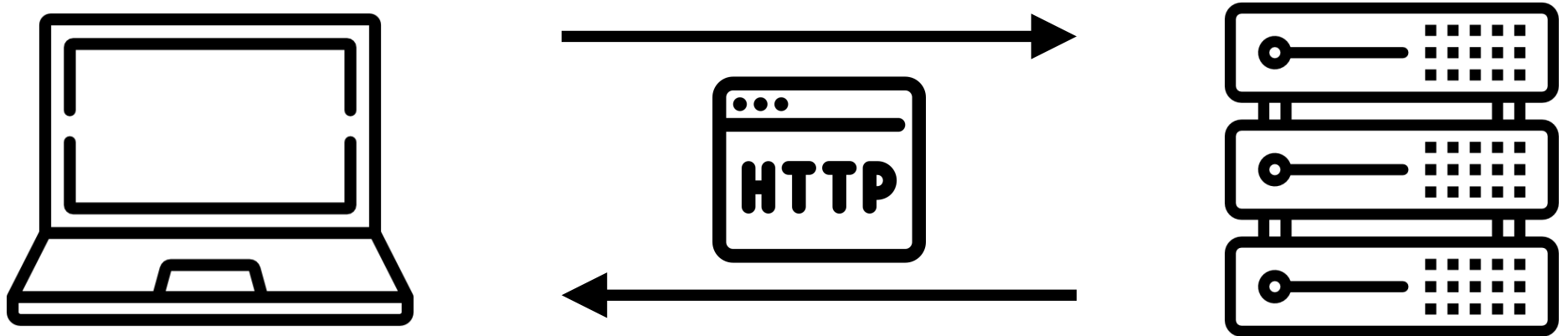


SESSIONID	ID
aFs28X1ka1d	user

■ Statueful



- Stateless



- Stateless



- 토큰

TOKEN

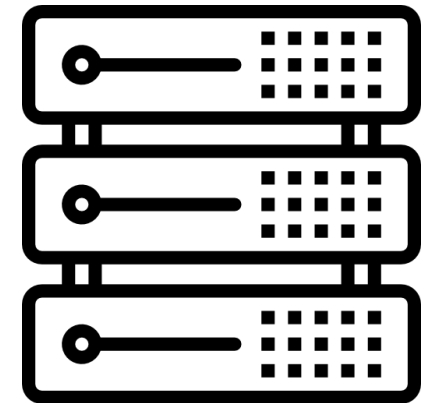
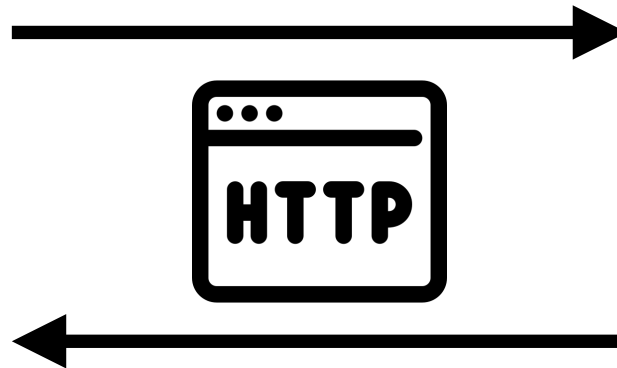
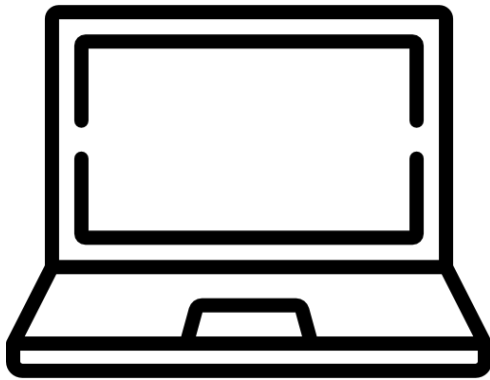
- 토큰

JSON WEB TOKEN



■ 4 : JWT 활용하기

Authorization: Basic dG9uZXk6cGFzc3dvcmQ



Key	Value
access	eyJhbG~



eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiJlbnRlc3QxMTExQGdtYWlsLmNvbSIsIm1hdCI6MTYyNzc5Njg1MywiZXhwIjozODU1MzMyODUzLCJzdWliOiJ1c2VySW5mbYJ9.ysSiHK598YR4BKbH_D6C9CbVG4uVXR5bCaHGmLTp6jk

■ 4 : JWT 활용하기

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiJlbnRlc3QxMTExQGdtYWlsLmNvbSImlhdCI6MTYyNzc5Njg1MywiZXhwIjo5ODUzLCJzdWIiOiJ1c2VySW5mb3J9.ysSiHK598YR4BKbH_D6C9CbVG4uVXR5bCaHGmLTp6jk



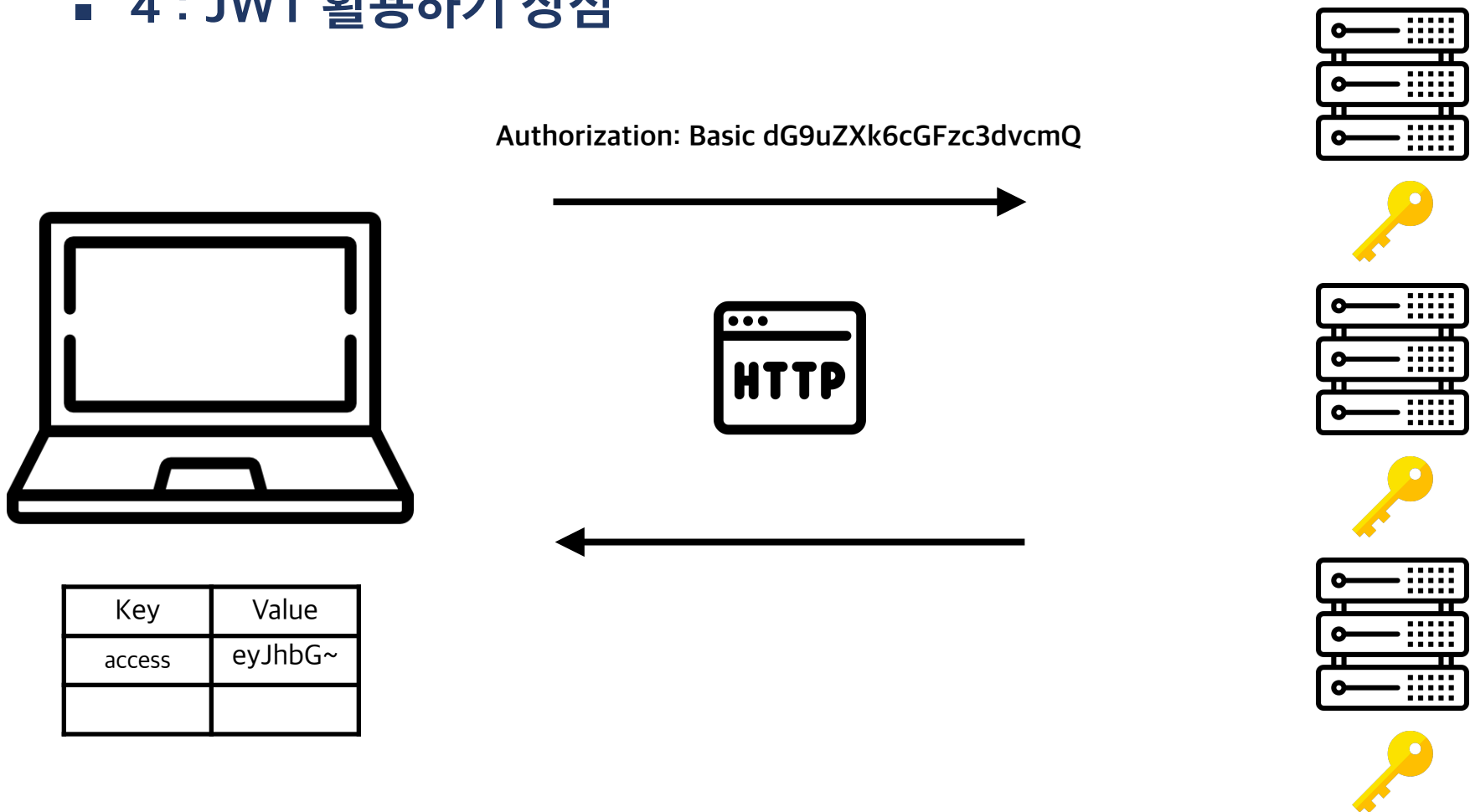
사용자 정보

이름

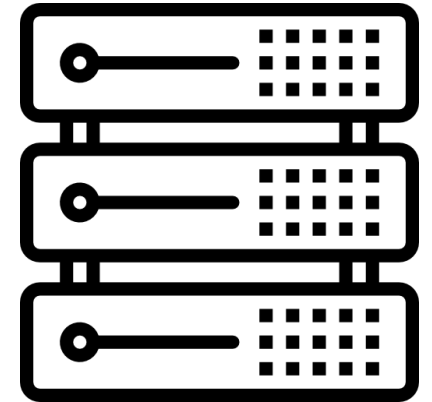
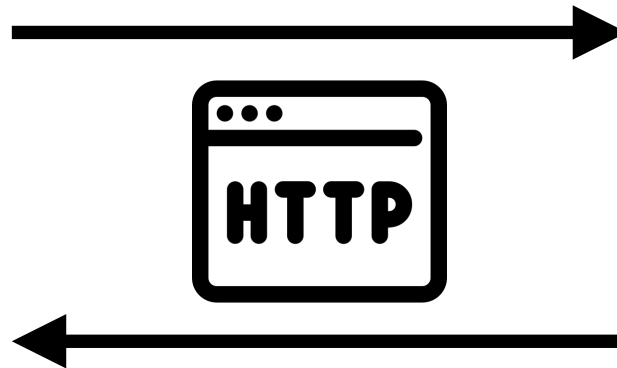
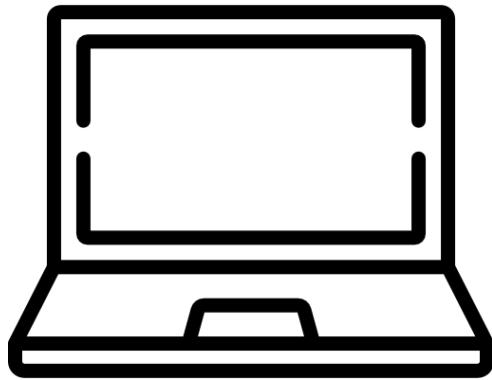
만료시기

권한

■ 4 : JWT 활용하기 장점



■ 4 : 토큰 활용하기 단점

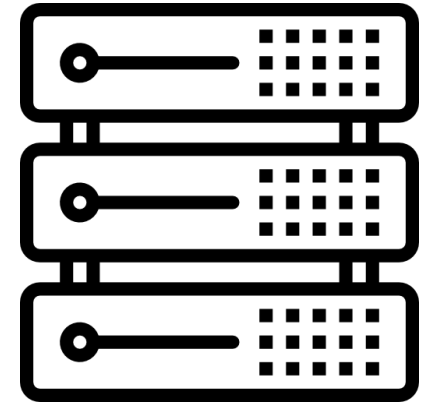
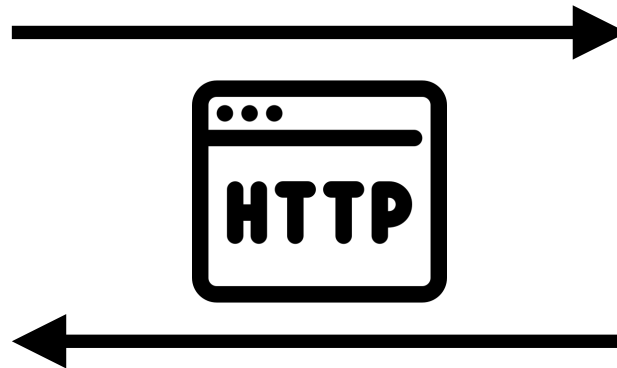
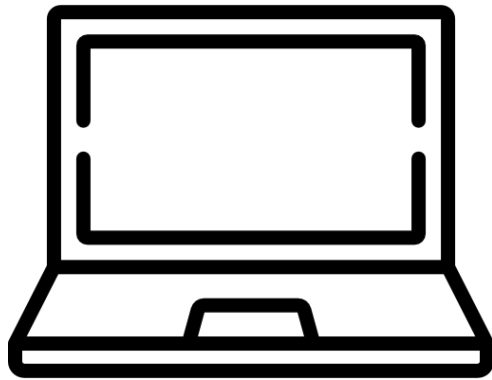


Key	Value

해커



■ 4 : 토큰 활용하기 단점



Key	Value
access	eyJhbG~

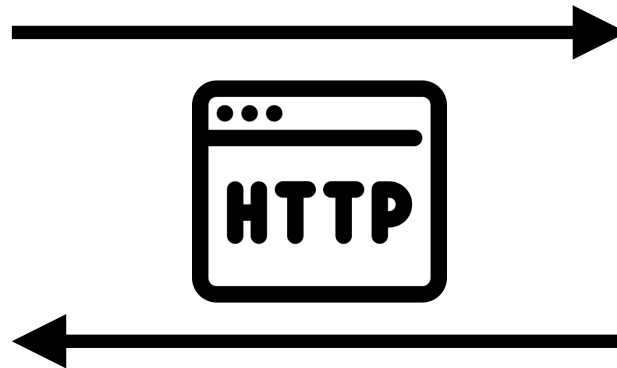
만료기한 30분

■ 4 : 토큰 활용하기 - Refresh Token



Key	Value

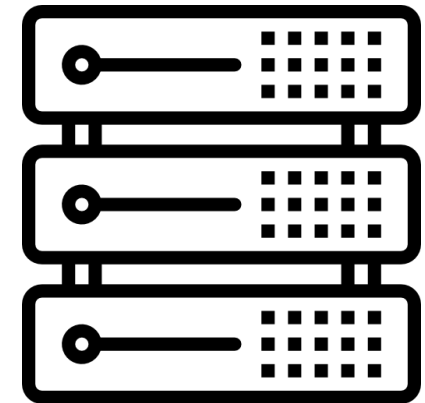
Authorization: Basic dG9uZXk6cGFzc3dvcmQ



Access Token : eyJhbG~

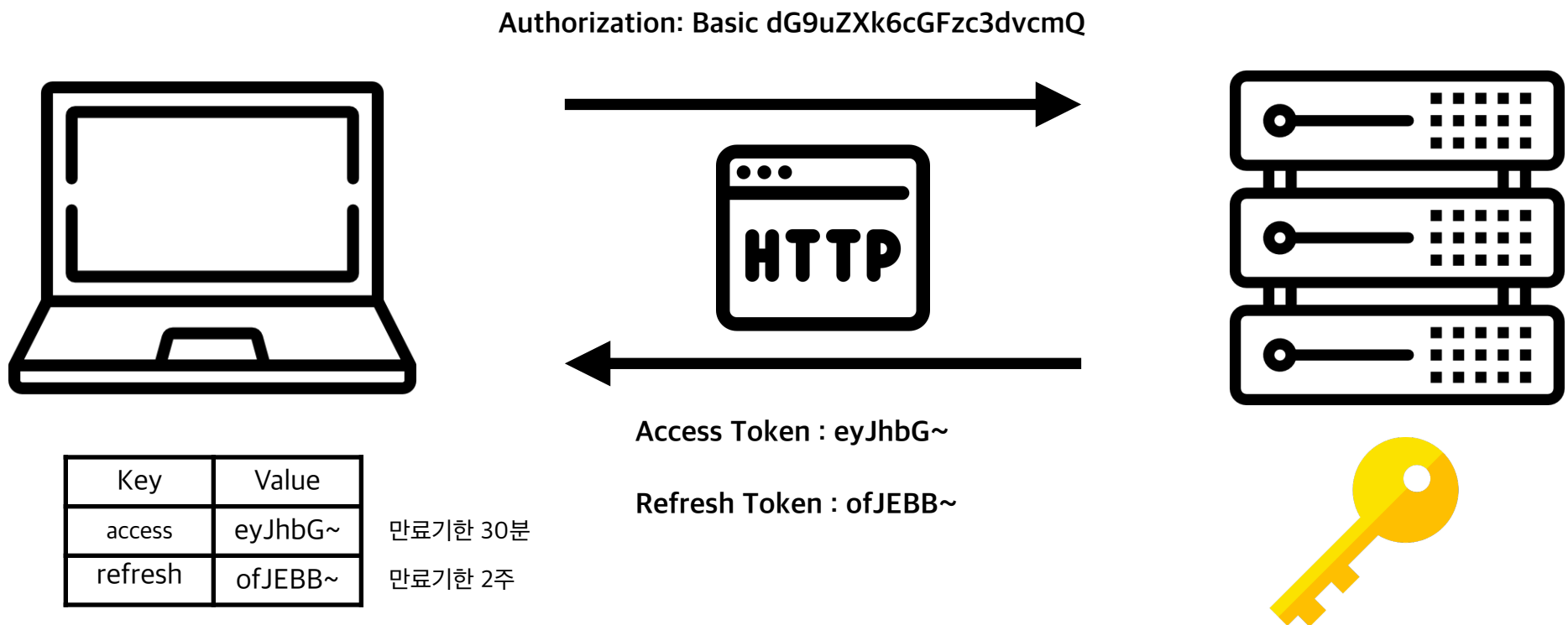
Refresh Token : ofJEBB~

Refresh Token	Expire Time
ofJEBB~	2021.10.10



Access Token : eyJhbG~

■ 4 : 토큰 활용하기 - Refresh Token

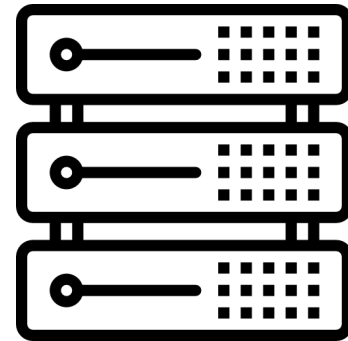
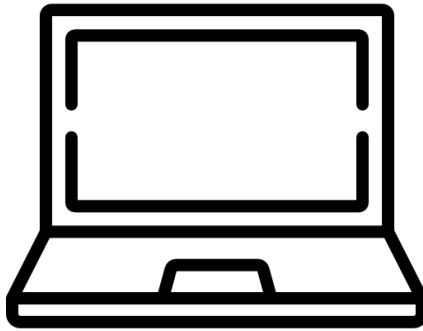
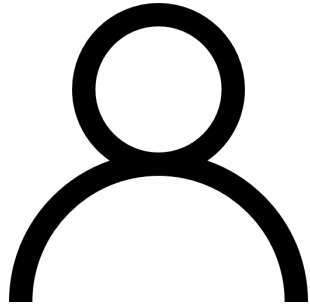


■ 4 : 토큰 활용하기 - 핵심

토큰으로 상태를관리하기 때문에 따로 세션을 둘 필요가 없다.

토큰 관리를 해야한다. 결국 토큰도 탈취당할 수 있다.

■ 복습



JWT

03

OAuth

- OAuth

What is OAuth?

- OAuth

다른 웹사이트 상의 자신들의 정보에 대해
접근 권한을 부여할 수 있는 공통적인 수단
개방형 표준

■ OAuth

☐ 로그인 상태 유지

☒ 보안접속

로그인

N

네이버 로그인



카카오 로그인

f

페이스북 로그인

아이디찾기

비밀번호찾기

회원가입

- OAuth

사용하는 인증절차

OAuth

☐ 로그인 상태 유지 ☒ 보안접속

로그인

N

네이버 로그인

카카오톡

카카오 로그인

f

페이스북 로그인

아이디찾기

비밀번호찾기

회원가입

■ OAuth 인증절차

유저 브라우저

서비스(Costumer)

OAuth 제공자

■ OAuth 인증절차

유저 브라우저

서비스(Costumer)

OAuth 제공자

- OAuth

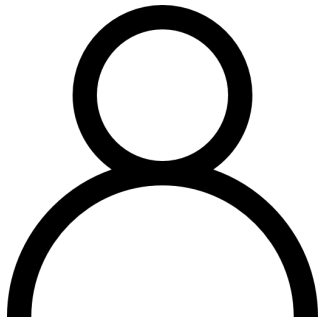
OAuth 2.0

Authorization Code Grant

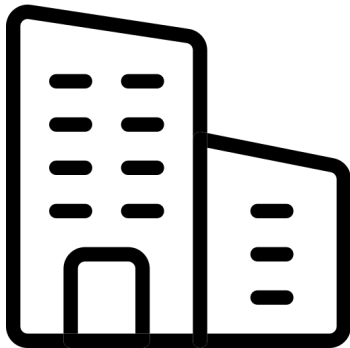
클라이언트가 다른 사용자 대신 특정 리소스에

접근을 요청할 때 사용

■ OAuth 장점



서비스에 ID/PW를 알려주지 않아도 됨
원할 때 access token의 권한 취소가 가능



유저의 access token만 가지고 있으면 됨
사용자의 ID/PW를 몰라도 허가 받은 API 접근 가능

04

Kakao Social Login

05

Paging

■ Paging

데이터베이스에서 읽어와 화면을 출력할 때

모든 데이터를 가져오는 것보다

출력될 페이지의 데이터만 나눠서 가져오는 것

■ Paging



■ Paging



- Paging

마지막 인덱스를 기억하고,
그 다음 인덱스를 페이징 크기에 맞게 주는 방식

06

Transaction

- Transaction

트랜잭션이란?

- Transaction

사용자가 게시물을 작성하고,
파일 업로드를 한 후 저장을 할 때
트랜잭션을 어떻게 처리해 주는 것이 좋을까요?

■ Transaction



A



B

■ Transaction

1) 처리 시작

- 데이터베이스 커넥션 생성

- **트랜잭션 시작**

2) 사용자의 로그인 여부 확인

3) 사용자의 글쓰기 내용의 오류 여부 확인

4) 첨부로 업로드된 파일 확인 및 저장

5) 사용자의 입력 내용을 DBMS에 저장

6) 첨부 파일 정보를 DBMS에 저장

7) 저장된 내용 또는 기타 정보를 DBMS에 조회

8) 게시물 등록에 대한 알림 메일 발송

9) 알림 메일 발송 이력을 DBMS에 저장

- **트랜잭션 종료(COMMIT)**

- 데이터베이스 커넥션 반납

10) 처리 완료

■ Transaction

1) 처리 시작

2) 사용자의 로그인 여부 확인

3) 사용자의 글쓰기 내용의 오류 여부 확인

4) 첨부로 업로드된 파일 확인 및 저장

- 데이터베이스 커넥션 생성(또는 커넥션 풀에서 가져오기)

- **트랜잭션 시작**

5) 사용자의 입력 내용을 DBMS에 저장

6) 첨부 파일 정보를 DBMS에 저장

- **트랜잭션 종료(COMMIT)**

7) 저장된 내용 또는 기타 정보를 DBMS에 조회

8) 게시물 등록에 대한 알림 메일 발송

- **트랜잭션 시작**

9) 알림 메일 발송 이력을 DBMS에 저장

- **트랜잭션 종료(COMMIT)**

- 데이터베이스 커넥션 종료 또는 커넥션 풀에 반납

10) 처리 완료

07

과제

■ 과제 (Standard Mission)

- JWT 활용하여 API 디벨롭하기
- Paging 처리를 하여 조회 일부 API 디벨롭하기
- Transaction 활용하여 API를 디벨롭하기

■ Reference

<https://www.flaticon.com/>
<https://choheeis.github.io/>
<https://www.youtube.com/watch?v=y0xMXIOAfss>
<https://www.youtube.com/watch?v=JZgD8aPkHSc>
<https://www.dailyimpact.co.kr/news/articleView.html?idxno=49345>
<https://zzang9ha.tistory.com/374?category=954131>