

Node.js를 이용한 로그인 세션 관리

작성자	조유연
작성일자	2025.03.20

<구현한 핵심 기능>

사용자가 로그인하면 server.js에서 session을 생성하고, 생성한 session을 DB에 저장하여 중복 로그인 시 리소스 접근을 제한합니다.

<데이터베이스 구조>

1. session_login_db_sessions

sessions 테이블을 생성하는 쿼리입니다. 유저의 세션 정보가 저장됩니다.

Information	
Table: sessions	
Columns:	
<u>sessionId</u>	varchar(128) PK
userId	int
createdAt	timestamp
isSimultaneousLogin	tinyint(1)
expiresAt	timestamp

- sessionId 컬럼에는 로그인한 유저의 세션 정보
- userId: 로그인한 유저의 username
- createdAt: 로그인 시각
- isSimultaneousLogin: 동시 로그인 여부를 체크. 0이면 첫 로그인, 1이면 2개 이상의 브라우저에서 동시 로그인한 것을 의미함
- expiresAt: 세션 정보의 만료시간(default: createdAt에 저장된 시간으로부터 1시간 이후)

2. session_login_db_login

users 테이블을 생성하는 쿼리입니다. 회원가입된 유저의 정보를 담고 있습니다.

Information	
Table: users	
Columns:	
id	int AI PK
username	varchar(50)
password	varchar(255)
created_at	timestamp

- id: 유저를 구분하는 식별자
- username: 유저의 사용자이름(ID)
- password: 유저의 비밀번호
- created_at: 회원가입한 날짜와 시각

<구현 페이지>

1. 로그인(<http://localhost:3000/login>)

[기본페이지]

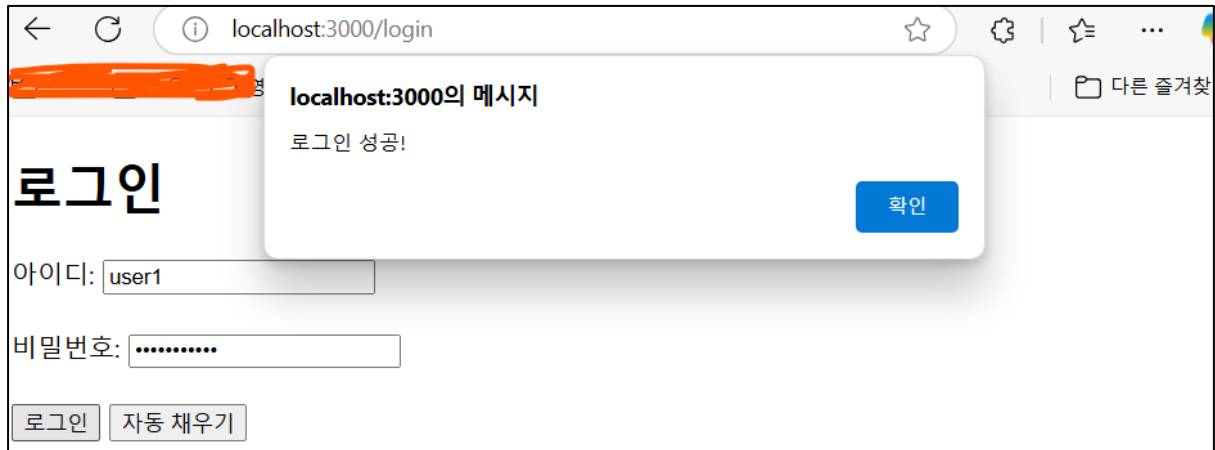
로그인

아이디:

비밀번호:

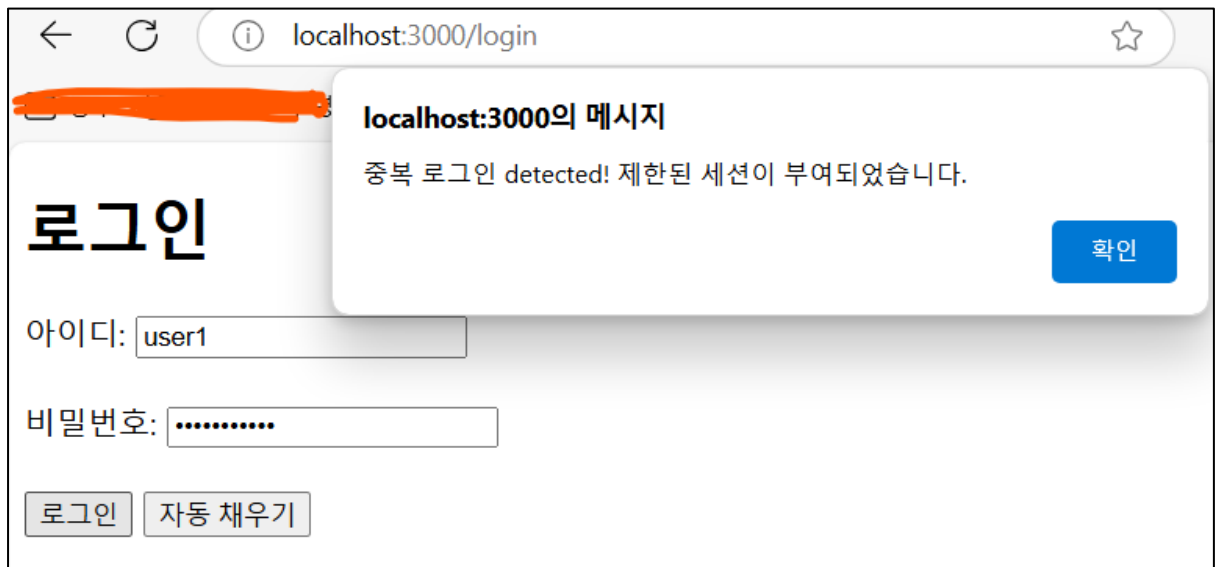
- 유저는 자신의 아이디와 비밀번호를 입력하고 '로그인' 버튼을 눌러 로그인할 수 있습니다.
- '자동 채우기' 버튼을 누르면 테스트용 데이터가 자동으로 삽입됩니다.

[중복된 세션이 없는 첫 로그인인 경우]



- '로그인 성공!'이라는 팝업 메시지와 함께 dashboard.html을 반환받습니다.

[중복 로그인일 경우]



- 이미 해당 유저의 세션이 DB에 존재할 경우, 위와 같은 경고창이 뜹니다.

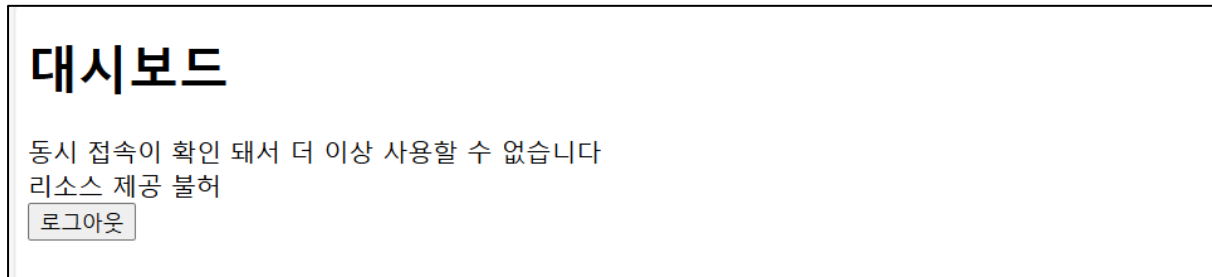
2. 대시보드(http:localhost:3000/dashboard)

[첫 로그인일 경우]



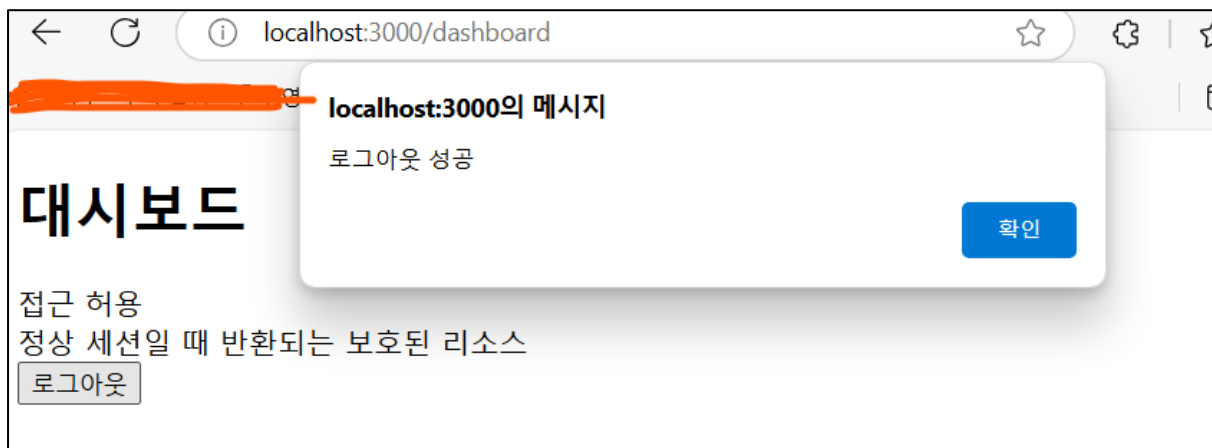
- 중복된 세션이 없는 첫 로그인에 성공했을 경우 리다이렉트되는 페이지입니다.

[중복된 세션이 존재하는 경우]



- 여러 브라우저에서 로그인할 경우 리소스 사용이 불가하다는 페이지를 반환합니다.

3. 로그아웃



- 유저는 대시보드 페이지에서 '로그아웃' 버튼을 눌러 정상적으로 로그아웃할 수 있습니다.

<테스트(실행) 방법>

1. DB설정

server.js에서 MySQL 접속정보를 로컬 환경에 맞게 변경할 수 있습니다.

```
12 // MySQL 연결 설정
13 const db = mysql.createPool({
14   host: "localhost",
15   user: "root",
16   password: "yuyeon_990312_session_login_5xK39Sd4cdEBksAFL",
17   database: "session_login_db",
18 });
```

2. users, sessions 테이블 생성

server.js를 통해 연결한 MySQL Database에 sessions 테이블과 users 테이블을 생성합니다.

송부 폴더에 포함된 session_login_db_sessions와 session_login_db_users를 실행하면 쉽게 users와 sessions 테이블을 생성할 수 있습니다.

node_modules	2025-03-20 오후 12:18	파일 폴더	
public	2025-03-20 오후 12:18	파일 폴더	
package	2025-03-15 오후 9:02	JSON 원본 파일	1KB
package-lock	2025-03-15 오후 9:02	JSON 원본 파일	35KB
server	2025-03-20 오후 2:41	JavaScript 원본 파...	5KB
session_login_db_sessions	2025-03-15 오후 9:57	SQL Text File	3KB
session_login_db_users	2025-03-15 오후 9:57	SQL Text File	3KB

- session_login_db_sessions에 포함된 쿼리는 sessions 테이블을 생성합니다.
- session_login_db_users에 포함된 쿼리는 users 테이블을 생성하고 테스트 데이터를 테이블에 주입합니다.
- session_login_db_users의 쿼리에 username: user1, password: 2b\$10\$oQzs7F3gEQOyJu49FHq82e.nfPupAxfDY/tzLzYPgvgGqoTwQ.Lk. (password123의 암호화된 값)을 users 테이블에 insert하는 쿼리가 포함되어 있습니다. 구현 기능을 확인하기 위한 테스트값입니다.

```
INSERT INTO `users` VALUES (1,'user1','2b$10$oQzs7F3gEQOyJu49FHq82e.nfPupAxfDY/tzLzYPgvgGqoTwQ.Lk.','2025-03-15 08:14:27'),(3,'user2','2b$10$ItyKUCMtQfA8pLcgW.qkIeIkjKz0tzlkyR7xwtiptZpQWZ4RNZ/R2','2025-03-15 08:35:10');
```

위 쿼리는 테스트데이터를 생성합니다.

3. 서버 실행

터미널에서 `node server.js` 을 입력합니다.

서버 실행 중: `http://localhost:3000`

위와 같은 출력문구가 뜨면 서버 실행에 성공했습니다.

브라우저에 출력된 url을 입력하여 웹 페이지에 접속합니다.

- <http://localhost:3000/login> 접속 시 [login.html](http://localhost:3000/login)을 GET요청을 통해 가져옵니다.
- 로그인 성공 시 <http://localhost:3000/dashboard> 로 리다이렉트됩니다. 동시에 데이터베이스의 'sessions' 테이블에 유저의 세션 정보가 저장됩니다.
- 다른 브라우저에서 로그인을 시도하면 입력받은 userId를 통해 데이터베이스

의 sessions테이블에서 해당 유저의 세션 정보가 있는지 확인합니다. 만약 이미 세션 정보가 존재한다면, '중복 session detected' 이라는 경고창과 함께 제한된 리소스를 제시합니다.

- 세션 중복없이 첫 로그인일 경우 '로그인 성공'이라는 팝업창과 함께 제한없이 리소스를 이용할 수 있습니다.