

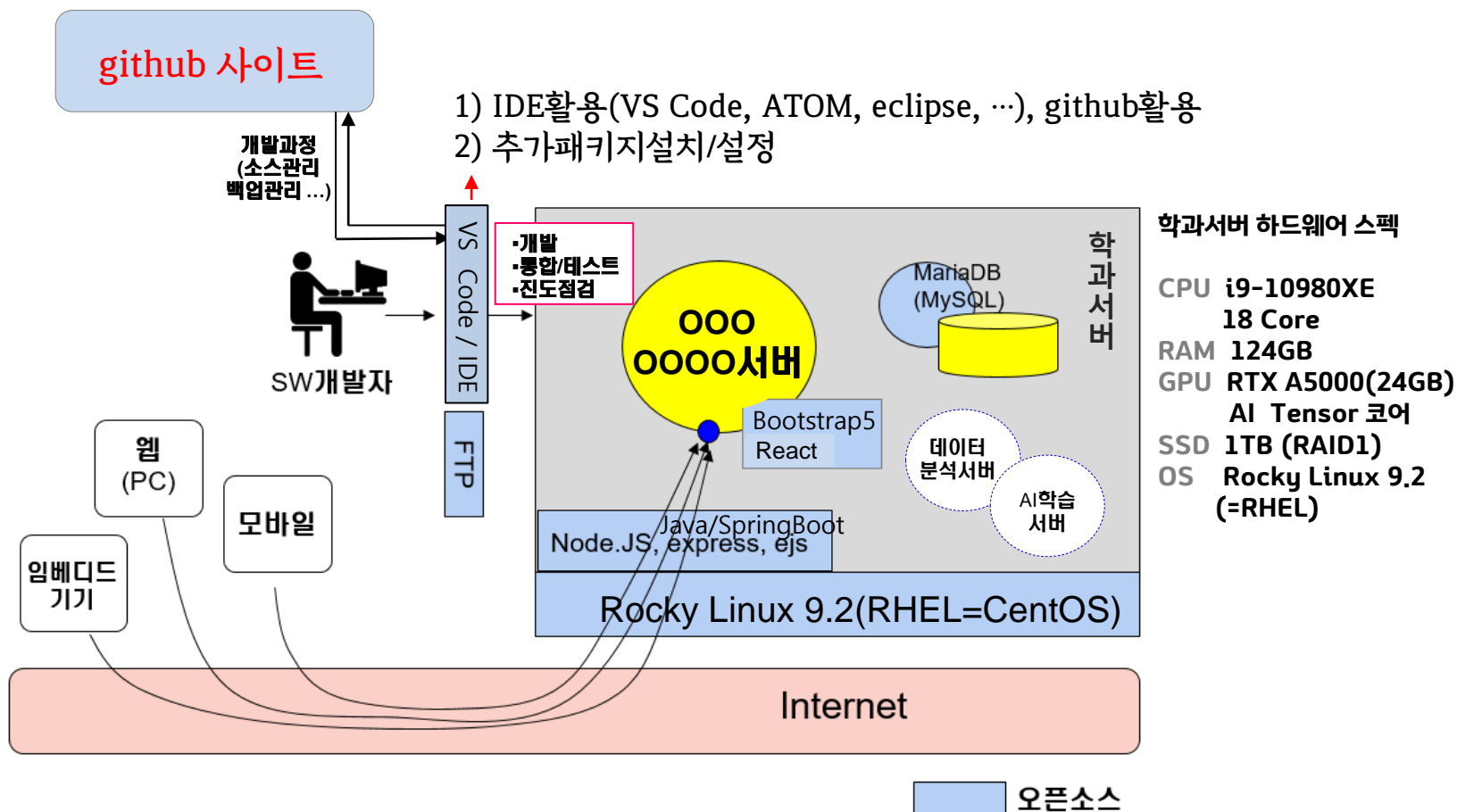
학과서버 운영가이드

## 개발환경 구축/설정 안내

컴퓨터공학과

# 개발환경 구축/설정

## ■ 학과프로젝트서버

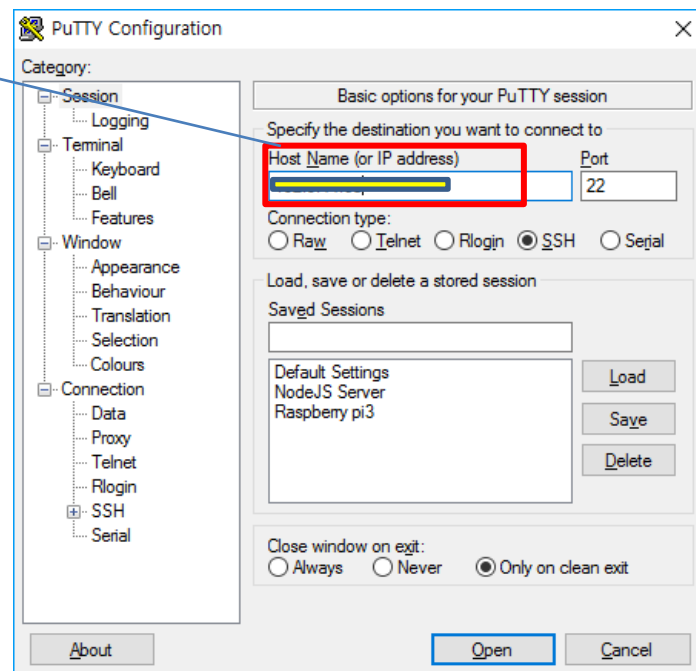


# 개발환경 구축/설정

- 서버: **ceprj.gachon.ac.kr**

팀번호	계정ID (팀장관리)	팀전용포트	초기비밀번호
이秉민	1 t23201	60001	Ce계정ID..  예, Cet23201..  로그인 후, 반드시 복잡한 비밀번호로 변경할 것!
	2 t23202	60002	
	3 t23203	60003	
	4 t23204	60004	
	5 t23205	60005	
강석환	1 t23206	60006	
	2 t23207	60007	
	3 t23208	60008	
	4 t23209	60009	
강동현	1 t23210	60010	
	2 t23211	60011	
	3 t23212	60012	
	4 t23213	60013	

team ↑ 2023-2학기 팀번호

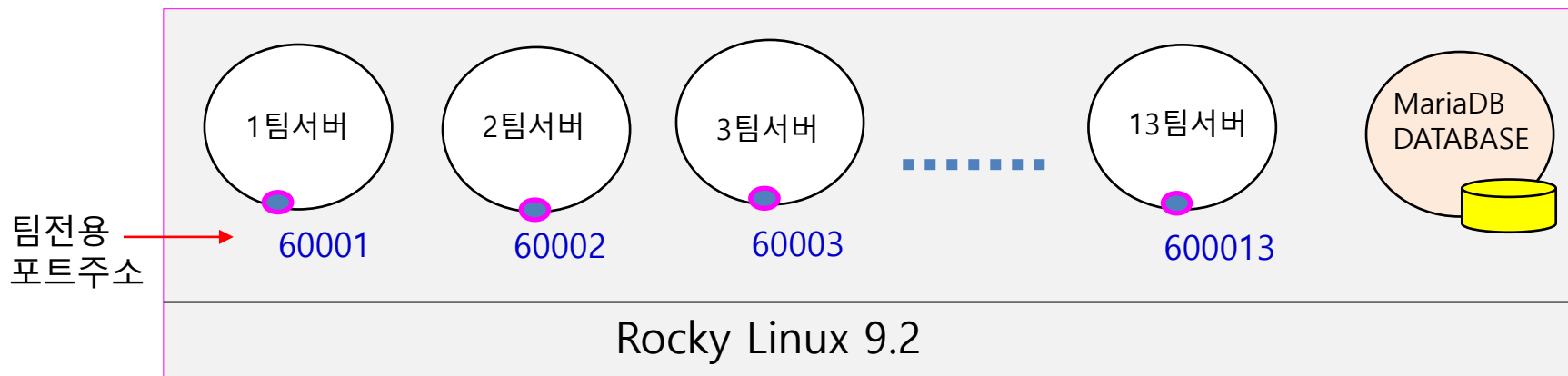


- 방화벽 개방포트 : **ftp(21), ssh(22), http(80)**

- 각 팀별 비밀번호 관리 중요(해외에서의 해킹가능성이 매우 높음)
- 비밀번호 변경할 때, 영대소문자,특수문자,숫자 조합 필수!, 10자리이상!

# 개발환경 구축/설정

## 서버 & 팀전용 포트주소



인터넷접속

- 파일을 업로드 할 때 ..... ftp 를 이용할 것!
- vi, VS Code, IDE로 파일작성/편집 할 때 ... ssh (putty) 이용할 것!
- 웹브라우저로 접속 할 때 ..... 웹브라우저 이용!

(예, 1팀이 접속할 때, <http://ceprj.gachon.ac.kr:60001> )

(예, 5팀이 접속할때, <http://ceprj.gachon.ac.kr:60005> )

- 포트주소가 더 필요한 팀은 요청할 것 !
- 단, DB포트는 외부에서 직접 접속금지!  
(즉, 반드시 웹서버를 통해서만 접속)

# 개발환경 구축/설정(node.js)

## ■ 서버환경(Node.JS) 설치 (v18.18.0)

- Linux 환경에서 설치

```
$ su -  
# wget https://nodejs.org/dist/v18.18.0/node-v18.18.0-linux-x64.tar.xz  
# tar xvfJ node-v18.18.0-linux-x64.tar.xz  
# cd node-v18.18.0-linux-x64  
# cp -R * /usr  
# exit  
$ node -v  
v18.18.0  
$ npm -v  
9.8.1  
$
```

node.js 버전

이미 서버에 설치를 해 놓았으므로 참고만 할 것!!

# 개발환경 구축/설정 (node.js)

## ■ (node.js/express 를 사용할 각팀은) 외부모듈 확장설치 할 것!

- 학과서버(Linux) 로그인한 후, 지정 디렉터리에 설치!

```
$ mkdir svr
$ cd svr
$ pwd
/home/t23201/svr
```

각 팀별마다  
예) 1팀의 경우

```
$ npm install ejs
$ npm install express
$ npm install cookie-parser
$ npm install body-parser
$ npm install express-session
$ npm install mysql
$ npm install multer
$ npm install http-errors
....
```

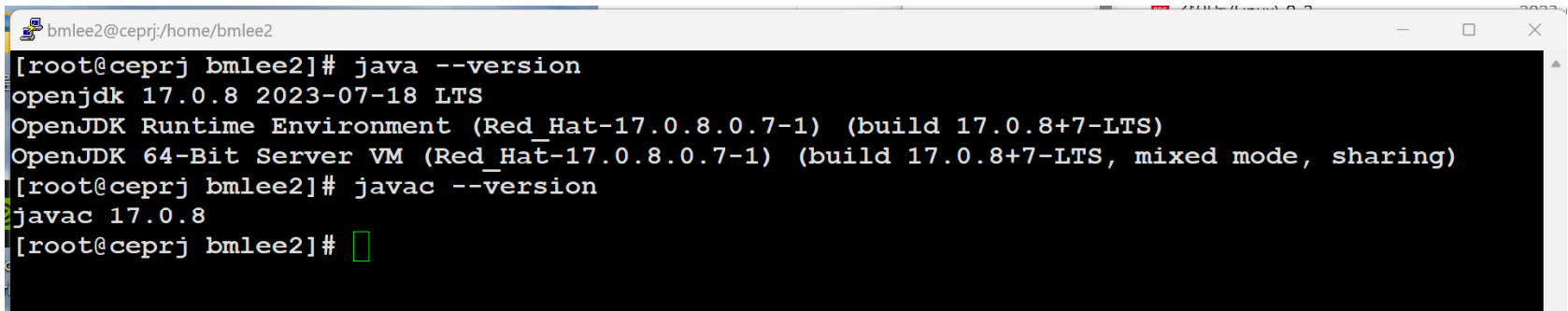
Local설치 디렉터리 위치

```
/home/t23101/svr — node_modules
├── express
├── ejs
├── express-session
├── body-parser
├── cookie-parser
├── mysql
└── ...
```

# 개발환경 구축/설정 (Java/Spring)

## ■ SpringBoot 프레임워크 설치

- Linux 환경에서 설치

A terminal window with a black background and white text. The window title is 'bmlee2@ceprj:/home/bmlee2'. The user is root at ceprj. The terminal shows the output of 'java --version' and 'javac --version' commands. The output for java shows 'openjdk 17.0.8 2023-07-18 LTS' and 'OpenJDK Runtime Environment (Red\_Hat-17.0.8.0.7-1) (build 17.0.8+7-LTS)'. The output for javac shows 'javac 17.0.8'. The prompt is '[root@ceprj bmlee2]#'.

```
[root@ceprj bmlee2]# java --version
openjdk 17.0.8 2023-07-18 LTS
OpenJDK Runtime Environment (Red_Hat-17.0.8.0.7-1) (build 17.0.8+7-LTS)
OpenJDK 64-Bit Server VM (Red_Hat-17.0.8.0.7-1) (build 17.0.8+7-LTS, mixed mode, sharing)
[root@ceprj bmlee2]# javac --version
javac 17.0.8
[root@ceprj bmlee2]#
```

현재 설치된 SpringBoot 프레임워크를 있으나, (만약에) 추가로 더 설치가 필요하다면,  
사용할 팀들 중에서 대표1명을 정해, 조율후 설치 할 것! (5일동안)

# 개발환경 구축/설정 (Java/Spring)

## ■ AI학습 프레임워크 설치

- Linux 환경, GPU 에서 설치 (단, 절대로 nvidia 드라이버 업데이트 금지!!!)
- 현재 설치된 GPU

```
[root@ceprj ~]# lspci | grep -e VGA
04:00.0 VGA compatible controller: ASPEED Technology, Inc. ASPEED Graphics Family (rev 41)
65:00.0 VGA compatible controller: NVIDIA Corporation GA102GL [RTX A5000] (rev a1)
[root@ceprj ~]#
```

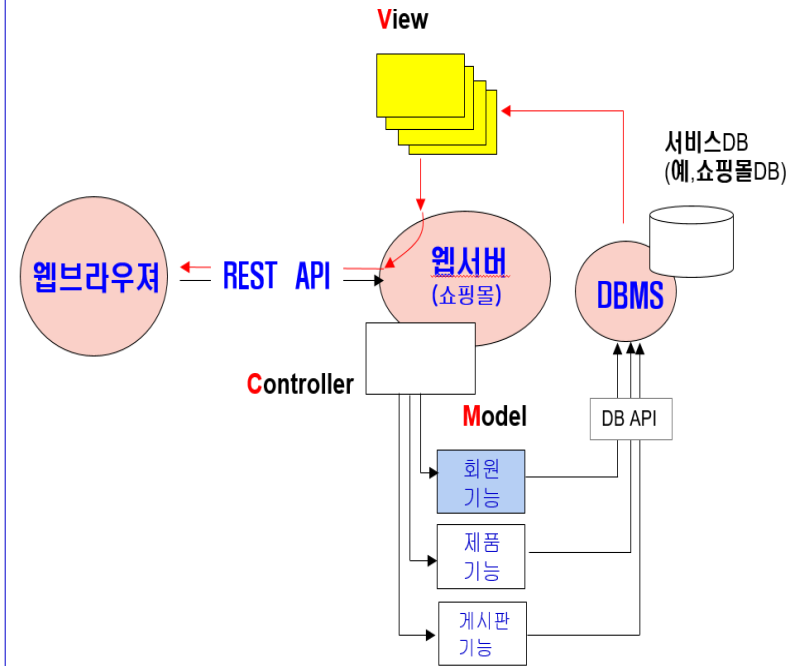
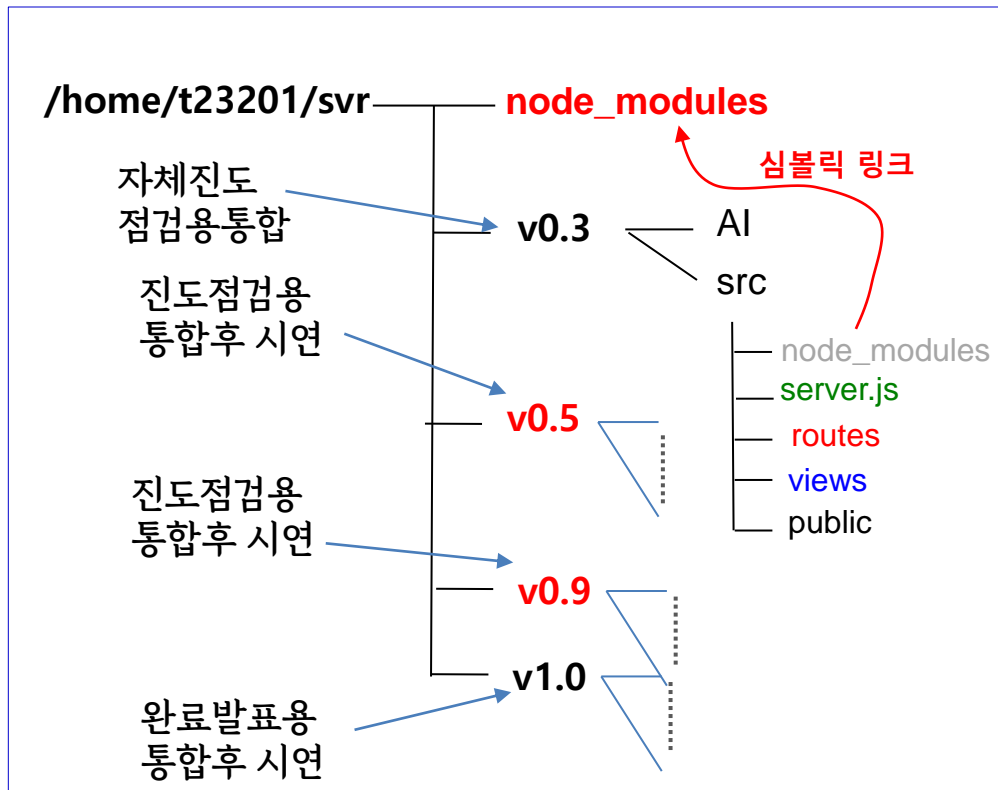
- 현재 설치된 nvidia 드라이버

```
[root@ceprj ~]# dnf info nvidia-driver
Last metadata expiration check: 2:08:07 ago on Tue 10 Oct 2023 06:20:02 PM KST.
Installed Packages
Name           : nvidia-driver
Epoch         : 3
Version        : 535.86.10
Release        : 1.el9
Architecture   : x86_64
Size           : 207 M
Source         : nvidia-driver-535.86.10-1.el9.src.rpm
Repository     : @System
From repo      : cuda-rhel9-x86_64
Summary        : NVIDIA's proprietary display driver for NVIDIA graphic cards
URL            : http://www.nvidia.com/object/unix.html
License        : NVIDIA License
Description    : This package provides the most recent NVIDIA display driver which allows for
                  : hardware accelerated rendering with recent NVIDIA chipsets.
                  :
                  : For the full product support list, please consult the release notes for driver
                  : version 535.86.10.
```



# 개발환경 구축/설정

- 아래의 디렉터리 구조를 반드시 따른다
- 진도점검은 서버에 통합하여 테스트후, 확인 받는다
- 졸업인증 2차심사시에 학과서버에 통합된 것으로 시연해야 함!



**MVC 구조!**

# DB 개발환경설정

## ▪ DB사용 요구조건(학과서버에 설치됨)

- MariaDB v10.11.4 (이미 설치함!)

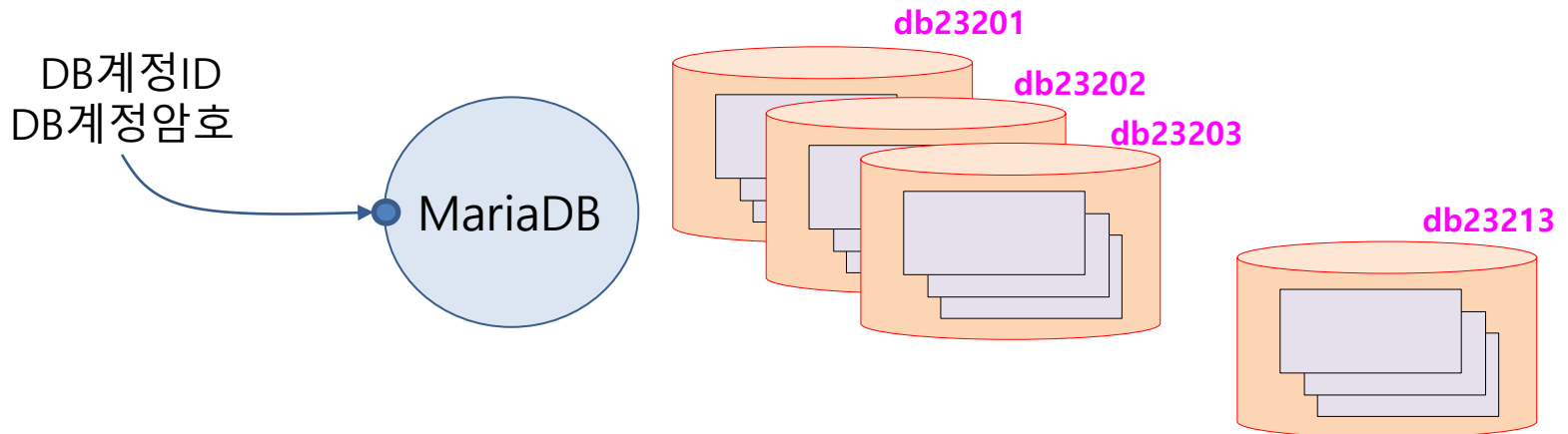
- DB계정ID/암호: **dbid232** / **dbpass232**

- DB명은 팀마다 아래의 지정된 룰로 사용한다.

(1팀: **db23201**, 2팀: **db23202**, 3팀: **db23203**, ... , 13팀: **db232013**)

- 테이블이름은 각 팀마다 설계서에 정의한대로 사용한다.

모든 팀이 이 DB계정과 암호를  
똑같이 사용하여야 함  
단, 팀별로 db명은 다름!



- DB는 로컬서버(웹서버프로그램 or putty)에서만 접속이 가능하다.
- DB는 원격으로 접속이 불가능하다. (**원격으로 DB관리tool 사용금지**)

# DB개발환경 설치

## ■ DB 사용방법

```
t23100@ceprj:~/src
[t23100@ceprj src]$ mysql -u dbid231 -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 10
Server version: 10.11.2-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| db23100  |
| information_schema |
| mysql    |
| performance_schema |
| sys      |
| test     |
+-----+
6 rows in set (0.000 sec)

MariaDB [(none)]> use db23100
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [db23100]> show tables;
+-----+
| Tables_in_db23100 |
+-----+
| member            |
+-----+
1 row in set (0.000 sec)

MariaDB [db23100]> select * from member;
+-----+-----+-----+
| name  | uid  | pass |
+-----+-----+-----+
| 이 병 문 | bmlee | 1234  |
| 방 탄  | BTS  | butter |
| 마 마 무 | mama | mumu  |
| sdf    | sdf  | sdf   |
+-----+-----+-----+
4 rows in set (0.000 sec)

MariaDB [db23100]> quit
Bye
[t23100@ceprj src]$
```

DB계정ID

DB계정 암호

예, 0팀 DB명

예, member 테이블

# DB개발환경 설치

## ■ DB 사용방법(사례)

The screenshot displays a development environment with a Node.js server and two web browser windows.

**Terminal Window (Top):** Shows the command `node server.js` being executed. The output indicates the web server is running and provides the following URLs:

- 회원 조회 : `http://192.9.202.17:60020/members`
- 회원 가입 : `http://192.9.202.17:60020/insert`

**Web Browser Window (Left):** Displays the "회원조회" (Member Search) page. It features a table with the following data:

이름	사용자id	비밀번호
이병문	bmlee	1234
방탄	BTS	butter
마마무	mama	mumu
sdf	sdf	sdf

**Web Browser Window (Right):** Displays the "회원가입" (Member Registration) page. It contains a form with the following fields:

- 이름: 가천대
- 사용자id: gachon
- 비밀번호: ....
- 가입: [button]

# DB개발환경 설치

## ■ DB 사용방법(사례)

```
const fs = require('fs');
const ejs = require('ejs');
const mysql = require('mysql');
const express = require('express');
const bodyParser = require('body-parser');
```

```
// MySQL DB 연결
const client = mysql.createConnection({
  host: 'localhost',      // DB서버 IP주소
  port: 3306,             // DB서버 Port주소
  user: 'dbid232',        // DB접속 아이디
  password: 'dbpass232',  // DB암호
  database: 'db23200'     //사용할 DB명
});
```

```
// 서버를 생성합니다.
const app = express();
app.use(bodyParser.urlencoded({ extended: false }));

// 서버를 실행합니다.
app.listen(60020, function () {
  console.log('웹서버 실행중...');
  console.log('회원조회: http://192.9.202.17:60020/members');
  console.log('회원가입: http://192.9.202.17:60020/insert');
});
```

```
// 라우트를 수행합니다.
app.get('/insert', (request, response) => {

  fs.readFile('views/insert.html', 'utf8', (error, data) => { //회원가입화면
    response.send(data); // 회원가입 화면전송
  });
});
```

server.js

자기팀 지정DB명을  
사용해야 함!!!  
(예, 4팀 db23204)

# DB개발환경 설치

## ■ DB 사용방법(사례)

server.js

```
app.get('/members', (request, response) => {
  fs.readFile('views/list.ejs', 'utf8', (error, data) => { // List화면
    // 데이터베이스 쿼리를 실행합니다.
    client.query('SELECT * FROM member', (error, results) => {
      // 응답합니다.
      response.send(ejs.render(data, {
        data: results // 회원조회 결과화면
      }));
    });
  });
});

app.post('/insert', function (request, response) {
  let body = request.body;

  console.log(body.name);
  console.log(body.uid);
  console.log(body.pass);

  client.query('INSERT INTO member (name, uid, pass) VALUES (?, ?, ?) ',
    [body.name, body.uid, body.pass], ( ) => {
    console.log("Insertion into DB was completed !");
    response.writeHead(200, { 'Content-Type': 'text/html' });
    response.end('<meta charset="utf-8"> 가입완료! <a href="http://192.9.202.17:60020/members">회원조회</a>');
  });
});
```

# DB개발환경 설치

## ■ DB 사용방법 (사례)

views/insert.html

```
<!DOCTYPE html>
<html>
<head>
  <title>회원가입</title>
</head>
<body>
  <h1>회원가입</h1>
  <hr />
  <form method="post">
    <fieldset>
      <legend>회원가입</legend>
      <table>
        <tr>
          <td><label>이름</label></td>
          <td><input type="text" name="name" /></td>
        </tr>
        <tr>
          <td><label>사용자id</label></td>
          <td><input type="text" name="uid" /></td>
        </tr>
        <tr>
          <td><label>비밀번호</label></td>
          <td><input type="password" name="pass" /></td>
        </tr>
      </table>
      <input type="submit" value = "가입" />
    </fieldset>
  </form>
</body>
</html>
```

# DB개발환경 설치

## ■ DB 사용방법 (사례)

views/list.ejs

```
<!DOCTYPE html>
<html>
<head>
  <title>List Page</title>
</head>
<body>
  <h1>회원조회</h1>
  <hr />
  <a href = "http://192.9.202.17:60020/insert/">+가입</a> <br>
  <table width="100%" border="1">
    <tr>
      <th>이름</th>
      <th>사용자id</th>
      <th>비밀번호</th>
    </tr>
    <% data.forEach(function (item, index) { %>
      <tr>
        <td><%= item.name %></td>
        <td><%= item.uid %></td>
        <td><%= item.pass %></td>
      </tr>
    <% }); %>
  </table>
</body>
</html>
```



# DB개발환경 설치

## ▪ SpringBoot DB연동 설정파일 (참고용)

```
then-then-0.0.1-SNAPSHOT-plain.jar spring.datasource.url=jdbc:mysql://localhost:3306/db23103
spring.datasource.username=dbid231
spring.datasource.password=dbpass231
spring.datasource.driver-class-name=org.mysql.jdbc.Driver
spring.jpa.hibernate.ddl-auto=validate
spring.jpa.properties.hibernate.format_sql=true
spring.jpa.show-sql=true
spring.web.resources.static-locations=classpath:/static/
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MariaDBDialect
@token secret key
jwt.token.secret="gachonUniversityBigdateProjectThenThenWebService"
server.port=60003
```

참조할 샘플: application.yml