
웹 프로그래밍 및 데이터베이스 2 (Node.js)

서종원

Node.js 개요

- Node.js는 **Chrome V8 JavaScript 엔진**으로 빌드된 **JavaScript 런타임 환경**

- JavaScript 엔진: JavaScript 코드를 컴퓨터가 이해하고 실행할 수 있는 기계어 코드로 변환해주는 프로그램
- V8은 Google에서 만든 JavaScript 엔진
- 런타임: 실행환경, 프로그래밍 언어가 구동되는 환경

- Node.js의 특징

이벤트 기반

- 이벤트: 클릭 키보드 입력, 파일 읽기 완료 등 시스템에서 발생하는 **특정 사건을 의미함**
- 이벤트 기반 프로그래밍: 프로그램의 흐름이 정해진 순서대로만 가는 것이 아니라 **특정 이벤트가 발생 했을 때 지정해둔 함수가 수행되는 방식**

논 블로킹 I/O (Non-blocking I/O)

- 블로킹 (동기방식): 동시에 일어나는 방식이 아닌 **순서에 맞춰 진행된다**는 의미
 - 웨이터가 1번 테이블에서 주문을 받고 주방에 주문을 전달한 뒤 음식이 다 만들어져 나올 때까지 주방 앞에서 대기
 - 2번 테이블, 3번 테이블 손님이 웨이터를 불러도 웨이터는 움직이지 않음.
 - 즉, 하나의 작업이 끝날 때까지 다른 모든 작업이 차단 (Block)됨
- 논 블로킹 (비동기방식): **순서에 맞추지 않고 진행한다**
 - 웨이터는 1번 테이블 주문을 주방에 전달한 후, 기다리지 않고 2번, 3번 테이블로 가서 주문을 받음
 - 주방에서 1번 음식이 나왔다는 이벤트가 발생하면 그때가서 음식을 가져다 줌
 - 하나의 작업 (Input/Output, DB 조회, 네트워크 요청)이 완료되길 **기다리지 않고 바로 다른 작업을 처리하는 방법**

단일 스레드

- 스레드 (Thread): 프로그램 내에서 작업을 수행하는 일꾼
- 단일 스레드: 한명의 일꾼이 모든 요청을 받고 처리하는 형태 단일 스레드는 멀티 스레드에 의해 발생하는 교착 (Deadlock)을 방지할 수 있음
 - 메모리 사용량도 적음, GPU 작업에 취약함

Node.js 개요

```
stu30@seo-B650M-PG-Lightning:~$ mkdir node_p  
stu30@seo-B650M-PG-Lightning:~$ cd node_p/  
stu30@seo-B650M-PG-Lightning:~/node_p$ ls
```

```
stu30@seo-B650M-PG-Lightning:~/node_p$ node -v  
v22.20.0  
stu30@seo-B650M-PG-Lightning:~/node_p$ npm -v  
10.9.3
```

Node.js 개요

```
stu30@seo-B650M-PG-Lightning:~/node_p$ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.
```

See ``npm help init`` for definitive documentation on these fields and exactly what they do.

Use ``npm install <pkg>`` afterwards to install a package and save it as a dependency in the package.json file.

Press ^C at any time to quit.

package name: (node_p)

version: (1.0.0)

description:

entry point: (index.js)

test command:

git repository:

keywords:

author:

license: (ISC)

About to write to /data/stu30/node_p/package.json:

```
{
  "name": "node_p",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}
```

Is this OK? (yes)

npm notice

npm notice New major version of npm available! 10.9.3 -> 11.6.1

npm notice Changelog: <https://github.com/npm/cli/releases/tag/v11.6.1>

npm notice To update run: `npm install -g npm@11.6.1`

npm notice

```
stu30@seo-B650M-PG-Lightning:~/node_p$ ls
package.json
```

GNU nano 7.2

```
{
  "name": "node_p",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}
```

Node.js 개요

function

- 일반 함수

async function

- 함수 앞에 async 붙으면 기다릴 줄 아는 능력을 부여함
- 항상 Promise를 반환함: async 함수는 호출되는 즉시 Promise 객체를 먼저 돌려줌
그리고 실제 작업은 백그라운드에서 계속 진행됨
- await 사용 가능: async 함수 안에서만 await 라는 키워드를 사용할 수 있음
백그라운드에서 진행 중인 Promise이 끝날 때까지 해당 함수의 실행만 잠시 멈추고 기다리는 역할을 함

const

- 상수(constant)를 의미함
- Const로 변수를 만들면 한번 값을 할당한 뒤에는 절대로 다른 값으로 바꿀 수 없음

let

- 값이 바뀔 수 있는 변수이며, 재할당이 가능함

var

- 옛날에 변수를 선언하던 방식 (const, let 이 나오기 이전..)
- if, for 문 안에서 var 로 선언한 변수가 계속 살아있는 버그..

Node.js 개요

```
GNU nano 7.2 first example.js
// --- 카페 바리스타의 하루 (동기/비동기 최종 비교 버전) ---

// 1. 일반 동기(Synchronous) 함수 : 아이스티 만들기
// - 이 함수는 'return'을 하기 전 까지 메인 스레드를 완전히 점유합니다.
function makeIcedTeaSync() {
  console.log("    [동기 작업 시작] ==> 바리스타가 아이스티를 직접 만듭니다. (1초 소요)");
  const startTime = Date.now();
  while (Date.now() - startTime < 1000) {} // 1초 동안 다른 일을 못하게 막음
  console.log("    [동기 작업 완료] <== 아이스티 완성!");
  return "아이스티";
}

// 2. 비동기(Asynchronous) 함수 : 커피 음료 만들기
// - 'async' 키워드가 붙어있어 항상 Promise(약속)를 반환합니다.
// - 이 함수는 즉시 Promise를 반환하고, 실제 작업은 백그라운드에서 계속됩니다.
async function makeCoffeeAsync(name, duration) {
  console.log(`    - [비동기 요청] '${name}' 주문 접수. 머신 작동! (예상 ${duration / 1000}초)`);
  return new Promise(resolve => {
    setTimeout(() => {
      console.log(`    △ [비동기 완료] '${name}' 완성!`);
      resolve(name);
    }, duration);
  });
}

// --- 실제 카페 운영 시나리오 ---
async function startCafe() {
  console.log("☕ 카페 오픈!");

  // --- 비동기 작업들을 먼저 '동시에' 요청합니다. ---
  // 'await'이 없으므로, 작업 지시만 하고 바로 다음 코드로 넘어갑니다.
  const lattePromise = makeCoffeeAsync("카페라떼", 3000);
  const espressoPromise = makeCoffeeAsync("에스프레소", 1000);

  console.log("\n바리스타 : '커피 머신들은 돌려놔오니, 다른 일을 해볼까...'");

  // --- 비동기 작업이 진행되는 동안, '동기' 작업을 수행합니다. ---
  // makeIcedTeaSync()는 일반 함수이므로, 1초 동안 프로그램의 흐름을 '막습니다(blocking)'.
  const icedTea = makeIcedTeaSync();
  console.log(`바리스타 : '손님, 주문하신 ${icedTea} 먼저 나왔습니다!'`);

  // --- 동기 작업이 끝난 후, 비동기 작업들이 완료되기를 기다립니다. ---
  console.log("\n바리스타 : '이제 아까 시켜둔 커피들이 다 됐나 확인해봐야지.'");
  const completedCoffees = await Promise.all([lattePromise, espressoPromise]);

  console.log("\n바리스타 : '주문하신 커피 (${completedCoffees.join(', ')}도 모두 나왔습니다!'");
}

startCafe();
```

```
stu30@seo-B650M-PG-Lightning:~/node_p$ node first_example.js
```

```
☕ 카페 오픈!
- [비동기 요청] '카페라떼' 주문 접수. 머신 작동! (예상 3초)
- [비동기 요청] '에스프레소' 주문 접수. 머신 작동! (예상 1초)
```

```
바리스타 : '커피 머신들은 돌려놔오니, 다른 일을 해볼까...'
    [동기 작업 시작] ==> 바리스타가 아이스티를 직접 만듭니다. (1초 소요)
    [동기 작업 완료] <== 아이스티 완성!
바리스타 : '손님, 주문하신 아이스티 먼저 나왔습니다!'
```

```
바리스타 : '이제 아까 시켜둔 커피들이 다 됐나 확인해봐야지.'
    △ [비동기 완료] '에스프레소' 완성!
    △ [비동기 완료] '카페라떼' 완성!
```

```
바리스타 : '주문하신 커피 (카페라떼, 에스프레소)도 모두 나왔습니다!'
```

Node.js 개요

=>

- function 키워드를 사용하는 함수를 더 짧고 간편하게 쓰기 위한 '줄임말' 또는 단축 문법
- 코드가 간결해져서 읽기 편해짐.

.then()

- Promise 객체에만 사용할 수 있는 특별한 명령어
- 이 약속이 성공적으로 지켜지면 다음에 이 코드를 실행해 줘라고 예약하는 역할을 함

프레임워크

프레임워크 (Framework)

- 어떤 일을 하기 위해 미리 구조와 규칙을 갖춰 놓은 뼈대 (틀)
- 즉, 개발자가 일일이 처음부터 모든 것을 만들 필요 없이 공통적으로 필요한 기능과 구조를 제공해 주는 도구임
프레임워크는 웹 애플리케이션을 더 빠르고, 쉽고, 안정적으로 만들 수 있도록 미리 준비된 규칙과 도구의 모음
- 프레임워크 없이 개발 (맨땅에 헤딩):
집을 짓는데 땅부터 파고 시멘트 직접 섞고 벽돌을 쌓아올리는 것과 동일
자유도는 높으나 시간이 매우 오래 걸리고 구조적으로 부실해질 위험이 큼
- 프레임워크를 사용한 개발
미리 설계된 튼튼한 뼈대위에 규격에 맞는 창문, 문틀을 제공받아 시작
개발자는 내부 인테리어에만 집중하면 됨
개발 속도가 매우 빠르고 기본적인 구조가 보장됨

Express 개요

Express

- Node.js 환경에서 동작하는 웹 애플리케이션 프레임워크

EJS (Embedded JavaScript)

- HTML 안에서 `<%= 변수 %>` 처럼 서버 값을 넣어 동적 HTML을 만드는 템플릿 엔진
Express에서 `app.set('view engine', 'ejs')`로 등록하고 `res.render('뷰이름', 데이터)`로 렌더링

```
stu30@seo-B650M-PG-Lightning:~$ mkdir node_express
stu30@seo-B650M-PG-Lightning:~$ cd node_express/
stu30@seo-B650M-PG-Lightning:~/node_express$ mkdir routes controllers models middlewares public views
stu30@seo-B650M-PG-Lightning:~/node_express$ touch app.js routes/index.js
stu30@seo-B650M-PG-Lightning:~/node_express$ tree
.
├── app.js
├── controllers
├── middlewares
├── models
├── public
├── routes
│   └── index.js
└── views
```

app.js → 시작점, 서버 설정

routes/ → 요청 URL 관리

controllers/ → 요청을 처리하는 핵심 로직

models/ → DB와의 연결 및 데이터 구조 정의

middlewares/ → 인증, 에러 처리

public/ → 정적 리소스(css, js, img)

views/ → 서버에서 동적 HTML 렌더링할 때 사용하는 템플릿

Express 개요

```
stu30@seo-B650M-PG-Lightning:~/node_express$ npm init -y
Wrote to /data/stu30/node_express/package.json:
```

```
{
  "name": "node_express",
  "version": "1.0.0",
  "description": "",
  "main": "app.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

```
stu30@seo-B650M-PG-Lightning:~/node_express$ npm install express
```

```
added 68 packages, and audited 69 packages in 503ms
```

```
16 packages are looking for funding
```

```
run `npm fund` for details
```

```
found 0 vulnerabilities
```

```
stu30@seo-B650M-PG-Lightning:~/node_express$ cat package.json
```

```
{
  "name": "node_express",
  "version": "1.0.0",
  "description": "",
  "main": "app.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "express": "^5.1.0"
  }
}
```

```
stu30@seo-B650M-PG-Lightning:~/node_express$ ls
app.js      middlewares  node_modules  package-lock.json  routes
controllers  models      package.json  public              views
```

Express 개요

```
stu30@seo-B650M-PG-Lightning:~/node_express$ touch views/index.ejs
```

```
stu30@seo-B650M-PG-Lightning:~/node_express$ ls
app.js      middlewares  node_modules  package-lock.json  routes
controllers models       package.json  public              views
stu30@seo-B650M-PG-Lightning:~/node_express$ cd public/
stu30@seo-B650M-PG-Lightning:~/node_express/public$ mkdir images
stu30@seo-B650M-PG-Lightning:~/node_express/public$ cd images/
stu30@seo-B650M-PG-Lightning:~/node_express/public/images$
```

```
stu30@seo-B650M-PG-Lightning:~/node_express/public/images$ ls
puppy.jpg
```

Express 개요

```
stu30@seo-B650M-PG-Lightning:~/node_express$ touch views/index.ejs
```

```
stu30@seo-B650M-PG-Lightning:~/node_express$ ls
app.js      middlewarees  node_modules  package-lock.json  routes
controllers models        package.json  public              views
stu30@seo-B650M-PG-Lightning:~/node_express$ nano app.js
```

```
GNU nano 7.2 app.js *
const path = require('path');
const express = require('express');
const app = express();

// 1) 뷰 엔진 설정 : EJS
app.set('view engine', 'ejs');
app.set('views', path.join(__dirname, 'views'));

// 2) 정적 파일 제공 (/public 하위 접근)
app.use(express.static(path.join(__dirname, 'public')));

// 3) 라우트 연결
const indexRouter = require('./routes/index');
app.use('/', indexRouter);

// 4) 서버 시작
const PORT = process.env.PORT || 3000;
app.listen(PORT, () => {
  console.log(`☑ 서버 실행 중 : http://localhost:${PORT}`);
});
```

Express 개요

```
stu30@seo-B650M-PG-Lightning:~/node_express$ cd routes/  
stu30@seo-B650M-PG-Lightning:~/node_express/routes$ ls  
index.js  
stu30@seo-B650M-PG-Lightning:~/node_express/routes$ nano index.js
```

```
const express = require('express');  
const router = express.Router();  
  
// DB 없이, 서버에서 뷰에 전달할 간단한 데이터  
router.get('/', (req, res) => {  
  res.render('index', {  
    title: '강아지 구경하세요',  
    subtitle: 'EJS 템플릿으로 화면 띄우기',  
    imageUrl: '/images/puppy.jpg' // public/images/puppy.jpg  
  });  
});  
  
module.exports = router;
```

Express 개요

```
stu30@seo-B650M-PG-Lightning:~/node_express/routes$ cd ..  
stu30@seo-B650M-PG-Lightning:~/node_express$ cd views/  
stu30@seo-B650M-PG-Lightning:~/node_express/views$ ls  
index.ejs  
stu30@seo-B650M-PG-Lightning:~/node_express/views$ nano index.ejs  
stu30@seo-B650M-PG-Lightning:~/node_express/views$ █
```

```
GNU nano 7.2                                index.ejs *
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <title><%= title %></title>
  <style>
    body { font-family: system-ui, -apple-system, Segoe UI, Roboto, Helvetica, Arial, sans-
    h1 { margin-bottom: .25rem; }
    .card {
      max-width: 640px; border: 1px solid #ddd; border-radius: 12px;
      padding: 24px; box-shadow: 0 4px 16px rgba(0,0,0,.06);
    }
    img { max-width: 100%; border-radius: 10px; display: block; margin-top: 12px; }
    .sub { color: #666; margin-bottom: 16px; }
    .footer { margin-top: 18px; color: #888; font-size: 14px; }
    code { background:#f6f8fa; padding:2px 6px; border-radius:6px; }
  </style>
</head>
<body>
  <div class="card">
    <h1><%= title %></h1>
    <p class="sub"><%= subtitle %></p>

    <div class="footer">
      이 페이지는 <code>EJS</code> 템플릿으로 서버에서 동적으로 렌더링되었습니다.
    </div>
  </div>
</body>
</html>
```

Express 개요

```
stu30@seo-B650M-PG-Lightning:~/node_express/views$ cd ..
stu30@seo-B650M-PG-Lightning:~/node_express$ ls
app.js      middlewares  node_modules  package-lock.json  routes
controllers models       package.json  public              views
```

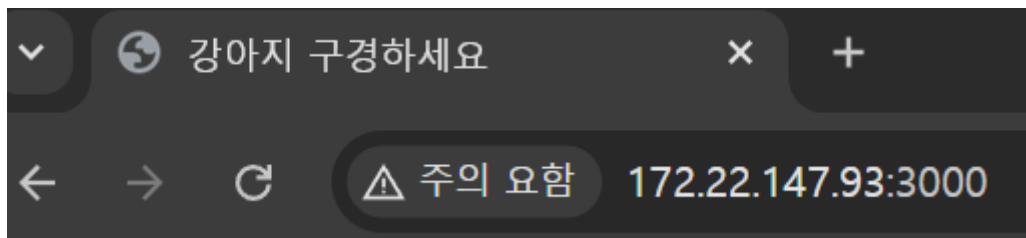
```
stu30@seo-B650M-PG-Lightning:~/node_express$ npm install ejs
added 8 packages, and audited 77 packages in 694ms

16 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
stu30@seo-B650M-PG-Lightning:~/node_express$ cat package.json
{
  "name": "node_express",
  "version": "1.0.0",
  "description": "",
  "main": "app.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "ejs": "^3.1.10",
    "express": "^5.1.0"
  }
}
stu30@seo-B650M-PG-Lightning:~/node_express$
```


Express 개요

```
stu30@seo-B650M-PG-Lightning:~/node_express$ node app.js  
☑ 서버 실행 중 : http://localhost:3000
```



Express, SQL

```
stu30@seo-B650M-PG-Lightning:~/node_express$ npm install mysql2 dotenv
added 13 packages, and audited 90 packages in 1s

18 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
stu30@seo-B650M-PG-Lightning:~/node_express$ cat package.json
{
  "name": "node_express",
  "version": "1.0.0",
  "description": "",
  "main": "app.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "dotenv": "^17.2.2",
    "ejs": "^3.1.10",
    "express": "^5.1.0",
    "mysql2": "^3.15.1"
  }
}
```

Express, SQL

```
stu30@seo-B650M-PG-Lightning:~/node express$ nano app.js
```

```
GNU nano 7.2 app.js *
const path = require('path');
const express = require('express');
const mysql = require('mysql2/promise'); // MariaDB 초기화용
const app = express();

/** =====
 * MariaDB 자동 초기화
 * - DB/테이블이 없으면 생성
 * ===== */
async function initMariaDB() {
  try {
    const connection = await mysql.createConnection({
      host: 'localhost',
      user: 'root',
      password: '123123' // ← 실습용으로 직접 입력
    });

    await connection.query('CREATE DATABASE IF NOT EXISTS puppy');
    await connection.query(`
      CREATE TABLE IF NOT EXISTS puppy.dogs (
        id INT AUTO_INCREMENT PRIMARY KEY,
        name VARCHAR(50),
        image VARCHAR(255)
      )
    `);

    console.log('MariaDB 초기화 완료');
    await connection.end();
  } catch (err) {
    console.error('MariaDB 초기화 실패:', err);
  }
}
initMariaDB();
```

```
/** =====
 * Express 기본 설정
 * ===== */
app.set('view engine', 'ejs');
app.set('views', path.join(__dirname, 'views'));

// 정적 파일 (/public 하위)
app.use(express.static(path.join(__dirname, 'public')));

// 폼 데이터 (x-www-form-urlencoded) 파싱
app.use(express.urlencoded({ extended: true }));

/** =====
 * 라우터
 * ===== */
const indexRouter = require('./routes/index'); // Step 1
const mariaRouter = require('./routes/maria'); // Step 2
app.use('/', indexRouter);
app.use('/maria', mariaRouter);

/** =====
 * 서버 시작
 * ===== */
const PORT = process.env.PORT || 3000;
app.listen(PORT, () => {
  console.log(`서버 실행 중 : http://localhost:${PORT}`);
});
```

```
stu30@seo-B650M-PG-Lightning:~/node_express$ cd models/  
stu30@seo-B650M-PG-Lightning:~/node_express/models$ nano maria.js
```

```
GNU nano 7.2                                maria.js *  
  
const mysql = require('mysql2/promise');  
  
const pool = mysql.createPool({  
  host: 'localhost',  
  user: 'root',  
  password: '123123',  
  database: 'puppy'  
});  
  
async function getAllDogs() {  
  const [rows] = await pool.query('SELECT * FROM dogs ORDER BY id DESC');  
  return rows;  
}  
  
async function addDog(name, image) {  
  await pool.query('INSERT INTO dogs (name, image) VALUES (?, ?)', [name, image]);  
}  
  
async function updateDog(id, name, image) {  
  await pool.query('UPDATE dogs SET name=?, image=? WHERE id=?', [name, image, id]);  
}  
  
async function deleteDog(id) {  
  await pool.query('DELETE FROM dogs WHERE id=?', [id]);  
}  
  
module.exports = { getAllDogs, addDog, updateDog, deleteDog };
```

Express, SQL

```
stu30@seo-B650M-PG-Lightning:~/node_express/models$ cd ..  
stu30@seo-B650M-PG-Lightning:~/node_express$ ls  
app.js      middlewares  node_modules  package-lock.json  routes  
controllers  models       package.json  public              views  
stu30@seo-B650M-PG-Lightning:~/node_express$ cd controllers/  
stu30@seo-B650M-PG-Lightning:~/node_express/controllers$ nano mariaController.js
```

```
GNU nano 7.2                                mariaController.js *
// controllers/mariaController.js
const maria = require('../models/maria');

async function listDogs(req, res) {
  const dogs = await maria.getAllDogs();
  res.render('maria', { dogs });
}

async function createDog(req, res) {
  const { name, image } = req.body;
  if (!name || !image) return res.redirect('/maria');
  await maria.addDog(name, image);
  res.redirect('/maria');
}

async function editDog(req, res) {
  const { id } = req.params;
  const { name, image } = req.body;
  if (!id) return res.redirect('/maria');
  await maria.updateDog(id, name, image);
  res.redirect('/maria');
}

async function removeDog(req, res) {
  const { id } = req.params;
  if (!id) return res.redirect('/maria');
  await maria.deleteDog(id);
  res.redirect('/maria');
}

module.exports = { listDogs, createDog, editDog, removeDog };
```

Express, SQL

```
stu30@seo-B650M-PG-Lightning:~/node_express/controllers$ cd ..
stu30@seo-B650M-PG-Lightning:~/node_express$ ls
app.js      middlewares  node_modules  package-lock.json  routes
controllers models      package.json  public              views
stu30@seo-B650M-PG-Lightning:~/node_express$ cd routes/
stu30@seo-B650M-PG-Lightning:~/node_express/routes$ ls
index.js
stu30@seo-B650M-PG-Lightning:~/node_express/routes$ nano maria.js
stu30@seo-B650M-PG-Lightning:~/node_express/routes$
```

```
GNU nano 7.2                                maria.js *
```

```
// routes/maria.js
const express = require('express');
const router = express.Router();
const mariaController = require('../controllers/mariaController');

router.get('/', mariaController.listDogs);
router.post('/add', mariaController.createDog);
router.post('/edit/:id', mariaController.editDog);
router.post('/delete/:id', mariaController.removeDog);

module.exports = router;
```

Express, SQL

```
stu30@seo-B650M-PG-Lightning:~/node_express/routes$ cd ..
stu30@seo-B650M-PG-Lightning:~/node_express$ ls
app.js      middlewares  node_modules  package-lock.json  routes
controllers models        package.json  public              views
stu30@seo-B650M-PG-Lightning:~/node_express$ cd views/
stu30@seo-B650M-PG-Lightning:~/node_express/views$ ls
index.ejs
stu30@seo-B650M-PG-Lightning:~/node_express/views$ nano maria.ejs
```



```
GNU nano 7.2          maria.ejs
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>강 아 지 구 경 하 세 요 - MariaDB</title>
  <style>
    body { font-family: system-ui, -apple-system, Segoe UI, Roboto, Helvetica, Arial, sans-
    img { max-width: 200px; border-radius: 8px; }
    ul { padding-left: 1rem; }
    li { margin: 12px 0; }
    form { margin: 6px 0; }
    input { margin-right: 6px; }
    a { text-decoration: none; margin-right: 10px; }
  </style>
</head>
<body>
  <h1>강 아 지 구 경 하 세 요 🏠 (MariaDB)</h1>
  <p>
    <a href="/">[홈]</a>
    <a href="/maria">[MariaDB]</a>
  </p>

  <h3>강 아 지 등 록</h3>
  <form action="/maria/add" method="POST">
    <input type="text" name="name" placeholder="강 아 지 이 름" required>
    <input type="text" name="image" placeholder="이 미 지 URL" required>
    <button type="submit">등 록</button>
  </form>
```

```
<h3>강 아 지   목   록 </h3>
<ul>
  <% if (!dogs || dogs.length === 0) { %>
    <li>등 록   된   강   아   지   가   없   습   니   다   .</li>
  <% } %>

  <% (dogs || []).forEach(dog => { %>
    <li>
      <strong>NO.<%= dog.id %> - <%= dog.name %></strong><br>
      "><br>

      <!-- 수   정   -->
      <form action="/maria/edit/<%= dog.id %>" method="POST">
        <input type="text" name="name" value="<%= dog.name %>">
        <input type="text" name="image" value="<%= dog.image %>">
        <button type="submit">수   정   </button>
      </form>

      <!-- 삭   제   -->
      <form action="/maria/delete/<%= dog.id %>" method="POST" onsubmit="return confirm('
        <button type="submit">삭   제   </button>
      </form>
    </li>
  <% }) %>
</ul>
</body>
</html>
```

Express, SQL

```
stu30@seo-B650M-PG-Lightning:~/node_express/views$ cd ..
stu30@seo-B650M-PG-Lightning:~/node_express$ ls
app.js      middlewares  node_modules  package-lock.json  routes
controllers models        package.json  public              views
stu30@seo-B650M-PG-Lightning:~/node_express$ node app.js
서버 실행 중 : http://localhost:3000
MariaDB 초기화 완료
```

```
stu30@seo-B650M-PG-Lightning:~$ mysql -u root -p
```

```
MariaDB [(none)]> USE puppy;
```

```
MariaDB [puppy]> SELECT * FROM dogs;
```

Express, MongoDB

```
stu30@seo-B650M-PG-Lightning:~$ ls
api.py  node_express  node_p  php_couchdb  php_mongodb  php_sql  snap  web-class
stu30@seo-B650M-PG-Lightning:~$ cd node_express/
stu30@seo-B650M-PG-Lightning:~/node_express$ ls
app.js  controllers  middlewares  models  node_modules  package.json  package-lock.json  public  routes  views
stu30@seo-B650M-PG-Lightning:~/node_express$ npm install mongoose multer

added 35 packages, and audited 125 packages in 3s

20 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
stu30@seo-B650M-PG-Lightning:~/node_express$ cat package.json
{
  "name": "node_express",
  "version": "1.0.0",
  "description": "",
  "main": "app.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "dotenv": "^17.2.2",
    "ejs": "^3.1.10",
    "express": "^5.1.0",
    "mongoose": "^8.18.2",
    "multer": "^2.0.2",
    "mysql2": "^3.15.1"
  }
}
```

Express, MongoDB

```
stu30@seo-B650M-PG-Lightning:~/node_express$ ls
app.js controllers middlewares models node_modules package.json package-lock.json public routes views
stu30@seo-B650M-PG-Lightning:~/node_express$ nano app.js
```

```
GNU nano 7.2 app.js
const path = require('path');
const express = require('express');
const mysql = require('mysql2/promise'); // MariaDB 초기화용
const app = express();

/** =====
 * MariaDB 자동 초기화
 * - DB/테이블이 없으면 생성
 * ===== */
async function initMariaDB() {
  try {
    const connection = await mysql.createConnection({
      host: 'localhost',
      user: 'root',
      password: '123123' // ← 실습용으로 직접 입력
    });

    await connection.query('CREATE DATABASE IF NOT EXISTS puppy');
    await connection.query(`
      CREATE TABLE IF NOT EXISTS puppy.dogs (
        id INT AUTO INCREMENT PRIMARY KEY,
        name VARCHAR(50),
        image VARCHAR(255)
      )
    `);

    console.log(' MariaDB 초기화 완료 ');
    await connection.end();
  } catch (err) {
    console.error(' MariaDB 초기화 실패:', err);
  }
}
initMariaDB();

/** =====
 * Express 기본 설정
 * ===== */
app.set('view engine', 'ejs');
app.set('views', path.join(__dirname, 'views'));

// 정적 파일 (/public 하위)
app.use(express.static(path.join(__dirname, 'public')));
app.use('/uploads', express.static(path.join(__dirname, 'uploads')));
```

```
// 폼 데이터 (x-www-form-urlencoded) 파싱
app.use(express.urlencoded({ extended: true }));

/** =====
 * 라우터
 * ===== */
const indexRouter = require('./routes/index');
const mariaRouter = require('./routes/maria');
const mongoRouter = require('./routes/mongo');

app.use('/', indexRouter);
app.use('/maria', mariaRouter);
app.use('/mongo', mongoRouter);

/** =====
 * 서버 시작
 * ===== */
const PORT = process.env.PORT || 3000;
app.listen(PORT, () => {
  console.log(`서버 실행 중 : http://localhost:${PORT}`);
});
```

Express, MongoDB

```
stu30@seo-B650M-PG-Lightning:~/node_express$ ls
app.js controllers middlewares models node_modules package.json package-lock.json public routes views
stu30@seo-B650M-PG-Lightning:~/node_express$ cd models/
stu30@seo-B650M-PG-Lightning:~/node_express/models$ ls
maria.js
stu30@seo-B650M-PG-Lightning:~/node_express/models$ nano mongo.js
```

```
GNU nano 7.2 mongo.js *
const mongoose = require('mongoose');

const MONGO_URI = 'mongodb://127.0.0.1:27017/puppy'; // 로컬 MongoDB
mongoose.connect(MONGO_URI, { dbName: 'puppy' }) // dbName은 중복이지만 안전하게
  .then(() => console.log('MongoDB 연결 성공'))
  .catch(err => console.error('MongoDB 연결 실패:', err));

const dogSchema = new mongoose.Schema({
  name: { type: String, required: true },
  imagePath: { type: String, required: true }, // 로컬 저장 경로 (/uploads/파일명)
  createdAt: { type: Date, default: Date.now }
});

module.exports = mongoose.model('MongoDog', dogSchema);
```

Express, MongoDB

```
stu30@seo-B650M-PG-Lightning:~/node_express/models$ cd ..
stu30@seo-B650M-PG-Lightning:~/node_express$ ls
app.js  controllers  middlewares  models  node_modules  package.json  package-lock.json  public  routes  views
stu30@seo-B650M-PG-Lightning:~/node_express$ cd controllers/
stu30@seo-B650M-PG-Lightning:~/node_express/controllers$ nano mongoController.js
```

```
GNU nano 7.2 mongoController.js *
const fs = require('fs');
const fsp = require('fs/promises');
const path = require('path');
const MongoDB = require('../models/mongo');

// 업로드 보관 폴더 보장
const UPLOAD_DIR = path.join(__dirname, '..', 'uploads');
if (!fs.existsSync(UPLOAD_DIR)) {
  fs.mkdirSync(UPLOAD_DIR, { recursive: true });
}

// 목록
async function listDogs(req, res) {
  const dogs = await MongoDB.find().sort({ createdAt: -1 }).lean();
  res.render('mongo', { dogs });
}

// 등록 (파일 필수)
async function createDog(req, res) {
  try {
    const { name } = req.body;
    if (!req.file) return res.redirect('/mongo'); // 파일 없으면 스킵
    const imagePath = `/uploads/${req.file.filename}`; // 정적 제공 경로
    await MongoDB.create({ name, imagePath });
    res.redirect('/mongo');
  } catch (err) {
    console.error(err);
    res.redirect('/mongo');
  }
}
```

```
// 수정 (이름 변경 + 이미지 교체 옵션)
async function editDog(req, res) {
  try {
    const { id } = req.params;
    const { name } = req.body;

    const dog = await MongoDB.findById(id);
    if (!dog) return res.redirect('/mongo');

    // 새 파일 올라왔으면 기존 파일 삭제 후 교체
    if (req.file) {
      const oldPath = path.join(__dirname, '..', dog.imagePath);
      const absOld = path.join(__dirname, '..', dog.imagePath.replace(/^\//, ''));
      try { await fsp.unlink(absOld); } catch (_) { /* 없을 수도 있음 */ }

      dog.imagePath = `/uploads/${req.file.filename}`;
    }

    if (name) dog.name = name;
    await dog.save();
    res.redirect('/mongo');
  } catch (err) {
    console.error(err);
    res.redirect('/mongo');
  }
}

// 삭제 (DB 삭제 + 파일 삭제)
async function removeDog(req, res) {
  try {
    const { id } = req.params;
    const dog = await MongoDB.findById(id);
    if (dog) {
      const absOld = path.join(__dirname, '..', dog.imagePath.replace(/^\//, ''));
      try { await fsp.unlink(absOld); } catch (_) { /* 없을 수도 있음 */ }
      await MongoDB.deleteOne({ _id: id });
    }
    res.redirect('/mongo');
  } catch (err) {
    console.error(err);
    res.redirect('/mongo');
  }
}

module.exports = { listDogs, createDog, editDog, removeDog, UPLOAD_DIR };
```

Express, MongoDB

```
stu30@seo-B650M-PG-Lightning:~/node_express/controllers$ cd ..
stu30@seo-B650M-PG-Lightning:~/node_express$ cd ls
-bash: cd: ls: No such file or directory
stu30@seo-B650M-PG-Lightning:~/node_express$ ls
app.js  controllers  middlewares  models  node_modules  package.json  package-lock.json  public  routes  views
stu30@seo-B650M-PG-Lightning:~/node_express$ cd routes/
stu30@seo-B650M-PG-Lightning:~/node_express/routes$ ls
index.js  maria.js
stu30@seo-B650M-PG-Lightning:~/node_express/routes$ nano mongo.js
stu30@seo-B650M-PG-Lightning:~/node_express/routes$ █
```


Express, MongoDB

```
GNU nano 7.2                                mongo.js *
const express = require('express');
const router = express.Router();
const path = require('path');
const multer = require('multer');
const { listDogs, createDog, editDog, removeDog, UPLOAD_DIR } = require('../controllers/mongoController');

const storage = multer.diskStorage({
  destination: (req, file, cb) => cb(null, UPLOAD_DIR),
  filename: (req, file, cb) => {
    const safe = Date.now() + '-' + file.originalname.replace(/\s+/g, '_');
    cb(null, safe);
  }
});

const upload = multer({
  storage,
  limits: { fileSize: 10 * 1024 * 1024 } // 10MB
});

// 목록
router.get('/', listDogs);

// 등록 (파일 필수): form field name="image"
router.post('/add', upload.single('image'), createDog);

// 수정 (파일 선택): name만 수정하거나, 파일 업로드로 교체 가능
router.post('/edit/:id', upload.single('image'), editDog);

// 삭제
router.post('/delete/:id', removeDog);

module.exports = router;
```

Express, MongoDB

```
stu30@seo-B650M-PG-Lightning:~/node_express/routes$ cd ..
stu30@seo-B650M-PG-Lightning:~/node_express$ ls
app.js  controllers  middlewares  models  node_modules  package.json  package-lock.json  public  routes  views
stu30@seo-B650M-PG-Lightning:~/node_express$ cd views/
stu30@seo-B650M-PG-Lightning:~/node_express/views$ ls
index.ejs  maria.ejs
stu30@seo-B650M-PG-Lightning:~/node_express/views$ nano mongo.ejs
stu30@seo-B650M-PG-Lightning:~/node_express/views$ █
```

Express, MongoDB

```
GNU nano 7.2                                mongo.ejs *
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>강 아 지 구 경 하 세 요 - MongoDB (파 일 업 로 드 )</title>
  <style>
    body { font-family: system-ui, -apple-system, Segoe UI, Roboto, Helvetica, Arial, sans-serif; margin: 40px; }
    img { max-width: 220px; border-radius: 8px; display: block; margin: 8px 0; }
    ul { padding-left: 1rem; }
    li { margin: 12px 0; }
    form { margin: 6px 0; }
    input { margin-right: 6px; }
    a { text-decoration: none; margin-right: 10px; }
  </style>
</head>
<body>
  <h1>강 아 지 구 경 하 세 요 📁 (MongoDB - 로 컬 파 일 업 로 드 )</h1>
  <p>
    <a href="/">[홈]</a>
    <a href="/maria">[MariaDB]</a>
    <a href="/mongo">[MongoDB]</a>
  </p>

  <!-- 등 록 -->
  <h3>강 아 지 등 록 (파 일 업 로 드 )</h3>
  <form action="/mongo/add" method="POST" enctype="multipart/form-data">
    <input type="text" name="name" placeholder="강 아 지 이 름 " required>
    <input type="file" name="image" accept="image/*" required>
    <button type="submit">등 록 </button>
  </form>
```

```
<!-- 목록 -->
<h3>강아지 목록 </h3>
<ul>
  <% if (!dogs || dogs.length === 0) { %>
    <li>등록된 강아지가 없습니다.</li>
  <% } %>
  <% (dogs || []).forEach(dog => { %>
    <li>
      <strong><%= dog.name %></strong><br>
      ">

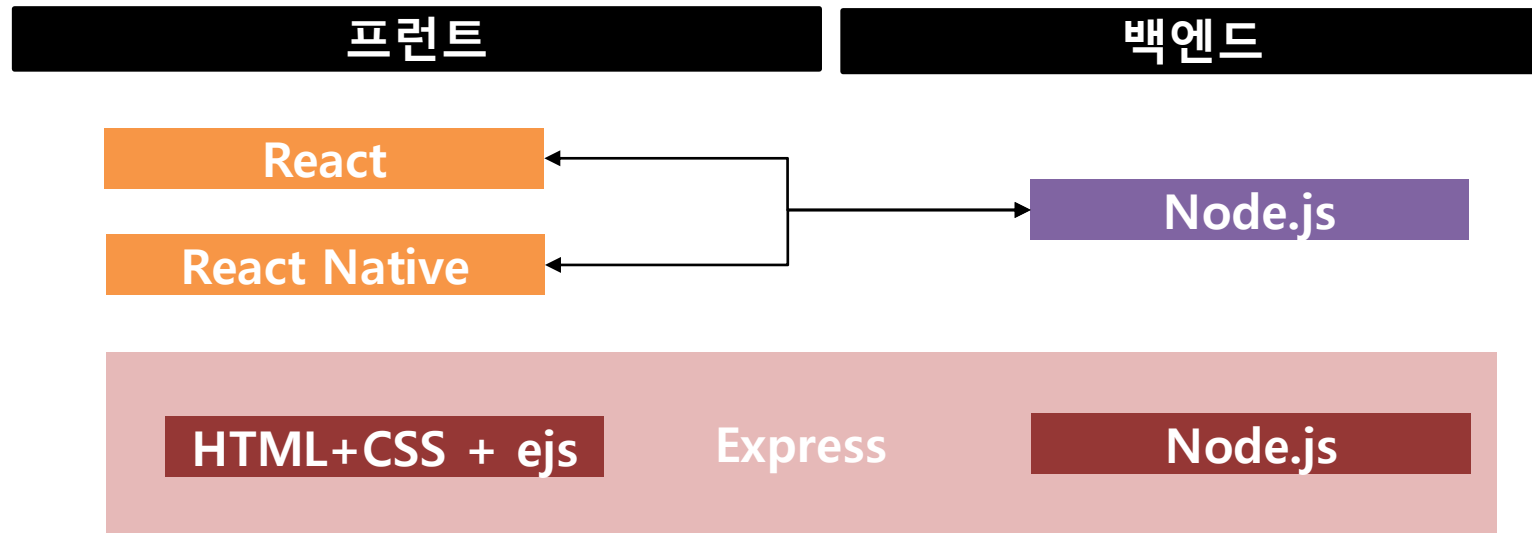
      <!-- 수정 : 이름만 또는 이미지 교체 -->
      <form action="/mongo/edit/<%= dog._id %>" method="POST" enctype="multipart/form-data">
        <input type="text" name="name" value="<%= dog.name %>">
        <input type="file" name="image" accept="image/*">
        <button type="submit">수정 </button>
      </form>

      <!-- 삭제 -->
      <form action="/mongo/delete/<%= dog._id %>" method="POST" onsubmit="return confirm('삭제 할까요?')">
        <button type="submit">삭제 </button>
      </form>
    </li>
  <% }) %>
</ul>
</body>
</html>
```

Express, MongoDB

```
stu30@seo-B650M-PG-Lightning:~/node_express/views$ cd ..
stu30@seo-B650M-PG-Lightning:~/node_express$ ls
app.js  controllers  middlewares  models  node_modules  package.json  package-lock.json  public  routes  views
stu30@seo-B650M-PG-Lightning:~/node_express$ node app.js
서버 실행 중 : http://localhost:3000
MongoDB 연결 성공
MariaDB 초기화 완료
```

React / React Native



React

- React는 사용자 인터페이스 (UI)를 만들기 위한 JavaScript 라이브러리 도구모음
- 웹사이트의 눈에 보이는 부분, 사용자가 클릭하고 상호작용하는 화면을 효율적으로 만들기 위해 페이스북(메타)에서 개발한 도구
- 웹 애플리케이션이 커지면 화면을 동적으로 바꾸는 일이 많아짐
버튼을 눌렀을 때 특정 부분만 바뀌어야 됨
새 데이터를 가져왔을 때 화면 일부만 갱신해야 됨
기존 jQuery 같은 방식은 복잡함
- 컴포넌트 기반으로 재사용 가능한 UI 조각들로 레고 블록으로 UI를 만들 수 있도록 해줌



```
stu30@seo-B650M-PG-Lightning:~$ ls
api.py  node_express  node_p  php_couchdb  php_mongodb  php_sql  snap  web-class
stu30@seo-B650M-PG-Lightning:~$ mkdir react_example
stu30@seo-B650M-PG-Lightning:~$ ls
api.py  node_express  node_p  php_couchdb  php_mongodb  php_sql  react_example  snap  web-class
stu30@seo-B650M-PG-Lightning:~$ cd react_example/
```

```
stu30@seo-B650M-PG-Lightning:~/react_example$ mkdir react
```

```
stu30@seo-B650M-PG-Lightning:~/react_example$ mkdir original
stu30@seo-B650M-PG-Lightning:~/react_example$ ls
original  react
```



```
stu30@seo-B650M-PG-Lightning:~/react_example$ cd original/  
stu30@seo-B650M-PG-Lightning:~/react_example/original$ npm init -y  
Wrote to /data/stu30/react_example/original/package.json:
```

```
{  
  "name": "original",  
  "version": "1.0.0",  
  "description": "",  
  "main": "index.js",  
  "scripts": {  
    "test": "echo \"Error: no test specified\" && exit 1"  
  },  
  "keywords": [],  
  "author": "",  
  "license": "ISC"  
}
```

```
stu30@seo-B650M-PG-Lightning:~/react_example/original$ nano package.json
```

```
GNU nano 7.2
{
  "name": "original",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "dependencies": {
    "express": "^4.19.2"
  },

  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

```
stu30@seo-B650M-PG-Lightning:~/react_example/original$ npm install

added 69 packages, and audited 70 packages in 935ms

14 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

```
stu30@seo-B650M-PG-Lightning:~/react_example/original$ ls
node_modules package.json package-lock.json
stu30@seo-B650M-PG-Lightning:~/react_example/original$ nano server.js
stu30@seo-B650M-PG-Lightning:~/react_example/original$
```

```
GNU nano 7.2 server.js *
const express = require("express");
const app = express();
const port = 3000;

app.use(express.static("public"));

app.get("/", (req, res) => {
  const name = "Node.js (Express)";
  const time = new Date().toLocaleString();

  // 서버가 HTML을 문자열로 생성해서 내려줍니다.
  res.send(`
<!DOCTYPE html>
<html lang="ko">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
    <title>Node SSR Example</title>
    <style>
      body { font-family: system-ui, sans-serif; padding: 40px; }
      h1 { margin-bottom: 8px; }
      .muted { color: #666; }
      button { padding: 10px 14px; border-radius: 8px; border: 1px solid #ddd; cursor: pointer; }
      button:hover { background: #f5f5f5; }
    </style>
  </head>
  <body>
    <h1>Hello ${name}!</h1>
    <p class="muted">이 페이지는 <strong>서버</strong>에서 HTML을 만들어 바로 내려줍니다.</p>
    <p>서버 시간이 필요할 때 유용합니다 : <strong>${time}</strong></p>

    <p>아래 버튼은 클라이언트에서 동작하는 간단한 JS입니다 (SSR과 무관):</p>
    <button onclick="alert('서버에서 만든 HTML에 클라이언트 JS도 섞을 수 있어요!')">알림</button>
  </body>
</html>
`);
});

app.listen(port, () => {
  console.log(`Server running: http://localhost:${port}`);
});
```

```
stu30@seo-B650M-PG-Lightning:~/react_example/original$ cd ..
stu30@seo-B650M-PG-Lightning:~/react_example$ ls
original  react
stu30@seo-B650M-PG-Lightning:~/react_example$ cd react/
stu30@seo-B650M-PG-Lightning:~/react_example/react$ ls
stu30@seo-B650M-PG-Lightning:~/react_example/react$ npm create vite@latest my-app
Need to install the following packages:
create-vite@8.0.1
Ok to proceed? (y) y

> npx
> create-vite my-app

|
|   Select a framework:
|   React
|
|   Select a variant:
|   JavaScript
|
|   Use rolldown-vite (Experimental)?:
|   No
|
|   Install with npm and start now?
|   Yes
|
|   Scaffolding project in /data/stu30/react_example/react/my-app...
|
|   Installing dependencies with npm...

added 152 packages, and audited 153 packages in 5s

32 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
|
|   Starting dev server...

> my-app@0.0.0 dev
> vite

VITE v7.1.7  ready in 77 ms

→ Local:   http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help
```

React

```
stu30@seo-B650M-PG-Lightning:~/react_example/react/my-app$ ls
eslint.config.js  index.html  node_modules  package.json  package-lock.json  public  README.md  src  vite.config.js
```

```
stu30@seo-B650M-PG-Lightning:~/react_example/react/my-app$ nano vite.config.js
```

```
GNU nano 7.2
import { defineConfig } from "vite";
import react from "@vitejs/plugin-react";

export default defineConfig({
  plugins: [react()],
  server: {
    host: true,
    port: 5173
  }
});
```

```
stu30@seo-B650M-PG-Lightning:~/react_example/react/my-app$ nano index.html
```

```
GNU nano 7.2 index.html *
<!doctype html>
<html lang="ko">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
    <title>React CSR Example</title>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.jsx"></script>
  </body>
</html>
```

```
stu30@seo-B650M-PG-Lightning:~/react_example/react/my-app$ nano src/main.jsx
```

```
GNU nano 7.2 src/main.jsx *
import React from "react";
import { createRoot } from "react-dom/client";
import App from "./App.jsx";

createRoot(document.getElementById("root")).render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

```
stu30@seo-B650M-PG-Lightning:~/react_example/react/my-app$ nano src/App.jsx
```

```
GNU nano 7.2 src/App.jsx *
import { useState } from "react";

export default function App() {
  const [count, setCount] = useState(0);
  const clientTime = new Date().toLocaleString();

  return (
    <div style={{ fontFamily: "system-ui, sans-serif", padding: 40 }}>
      <h1>Hello React (Vite)!</h1>
      <p style={{ color: "#666" }}>
        이 페이지는 <strong>브라우저 (클라이언트)</strong>에서 React가 렌더링합니다.
      </p>

      <p>
        클라이언트 기준 시간 : <strong>{clientTime}</strong>
      </p>

      <div style={{ marginTop: 20 }}>
        <button
          onClick={() => setCount((c) => c + 1)}
          style={{
            padding: "10px 14px",
            borderRadius: 8,
            border: "1px solid #ddd",
            cursor: "pointer",
            marginRight: 8
          }}
        >
          클릭 수 : {count}
        </button>
      </div>
    </div>
  );
}
```



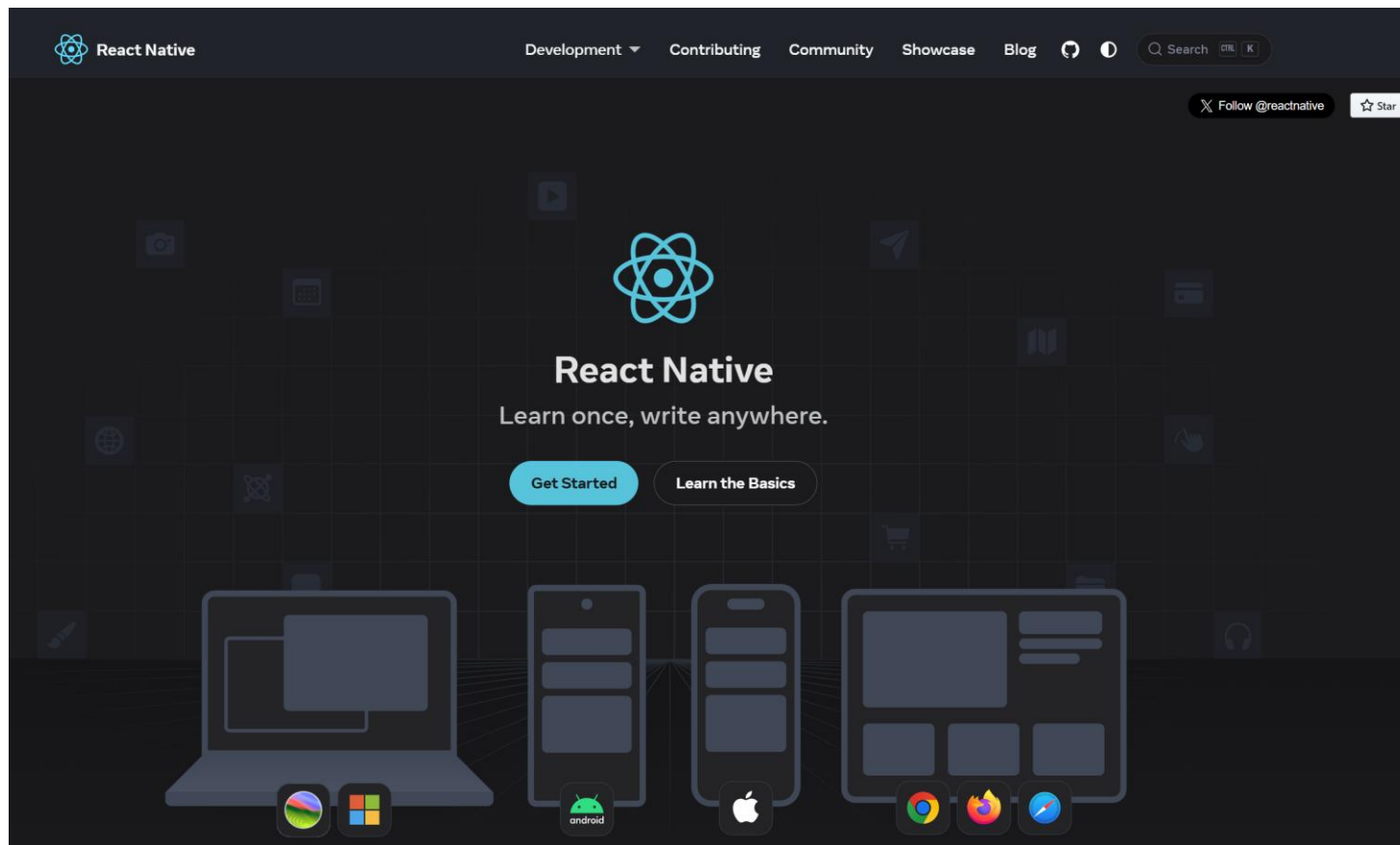
```
    <button
      onClick={() => alert("클 라 이 언 트 에 서 즉 시 반 응 합 니 다 !")}
      style={{
        padding: "10px 14px",
        borderRadius: 8,
        border: "1px solid #ddd",
        cursor: "pointer"
      }}
    >
      알 림
    </button>
  </div>

  <hr style={{ margin: "24px 0" }} />
  <p>
    ⚡ 차 이 한 눈 요 약 : <br />
    Node-SSR은 서 버 가 HTML을 만 들 어 보 내 고 , React-CSR은 브 라 우 저 가 JS로 화 면 을 그 립 니 다 .
  </p>
</div>
);
}
```

```
stu30@seo-B650M-PG-Lightning:~/react_example/react/my-app$ ls
eslint.config.js  index.html  node_modules  package.json  package-lock.json  public  README.md  src  vite.config.js
stu30@seo-B650M-PG-Lightning:~/react_example/react/my-app$ npm run dev
```

React Native

- React 문법으로 iOS/Android UI를 만드는 프레임워크
- 컴포넌트 <View>, <Text>, <Image> 등을 사용



React Native

```
stu30@seo-B650M-PG-Lightning:~$ ls
api.py  node_express  node_p  php_couchdb  php_mongodb  php_sql  react_example  snap  web-class
stu30@seo-B650M-PG-Lightning:~$ mkdir node_rn
stu30@seo-B650M-PG-Lightning:~$ ls
api.py  node_express  node_p  node_rn  php_couchdb  php_mongodb  php_sql  react_example  snap  web-class
stu30@seo-B650M-PG-Lightning:~$ cd node_rn/
```

```
stu30@seo-B650M-PG-Lightning:~/node_rn$ npm init -y
Wrote to /data/stu30/node_rn/package.json:

{
  "name": "node_rn",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

React Native

```
stu30@seo-B650M-PG-Lightning:~/node_rn$ npm i expo@^54
npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a good and tested way to coalesce async requests by a key value, which is much more comprehensive and powerful.
npm warn deprecated rimraf@3.0.2: Rimraf versions prior to v4 are no longer supported
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported

added 651 packages, and audited 652 packages in 3s

60 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
stu30@seo-B650M-PG-Lightning:~/node_rn$ cat package.json
{
  "name": "node_rn",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "expo": "^54.0.10"
  }
}
stu30@seo-B650M-PG-Lightning:~/node_rn$
```

React Native

```
stu30@seo-B650M-PG-Lightning:~/node_rn$ npx expo install react react-native
> Installing 2 SDK 54.0.0 compatible native modules using npm
> npm install

added 1 package, changed 1 package, and audited 653 packages in 538ms

61 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

```
stu30@seo-B650M-PG-Lightning:~/node_rn$ npm i -D babel-preset-expo @expo/ngrok@^4
npm warn deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.

added 32 packages, and audited 685 packages in 7s

70 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

React Native

```
stu30@seo-B650M-PG-Lightning:~/node_rn$ ls
node_modules package.json package-lock.json
stu30@seo-B650M-PG-Lightning:~/node_rn$ nano app.json
```

```
GNU nano 7.2
{
  "expo": {
    "name": "RN Demo",
    "slug": "rn-demo"
  }
}
```

```
stu30@seo-B650M-PG-Lightning:~/node_rn$ ls
app.json node_modules package.json package-lock.json
stu30@seo-B650M-PG-Lightning:~/node_rn$ nano babel.config.js
```

```
GNU nano 7.2
module.exports = function(api) {
  api.cache(true);
  return { presets: ['babel-preset-expo'] };
};
```

```
stu30@seo-B650M-PG-Lightning:~/node_rn$ ls
app.json  babel.config.js  node_modules  package.json  package-lock.json
stu30@seo-B650M-PG-Lightning:~/node_rn$ nano App.js
```

React Native

```
GNU nano 7.2 App.js *
import React, { useState } from "react";
import { SafeAreaView, View, Text, TextInput, Button, StyleSheet, Platform } from "react-native";

export default function App() {
  const [name, setName] = useState("React Native");
  const [count, setCount] = useState(0);

  return (
    <SafeAreaView style={styles.safe}>
      <View style={styles.wrap}>
        <Text style={styles.title}>Hello {name} </Text>
        <Text style={styles.sub}>지금 {Platform.OS.toUpperCase()} 네 이 티 브 로 렌 더 링 중 </Text>

        <TextInput
          value={name}
          onChangeText={setName}
          placeholder="이 름 입 력 "
          style={styles.input}
        />

        <Button title={`카운트 : ${count}`} onPress={() => setCount(c => c + 1)} />
      </View>
    </SafeAreaView>
  );
}

const styles = StyleSheet.create({
  safe: { flex: 1, backgroundColor: "#fff" },
  wrap: { flex: 1, padding: 24, gap: 16, justifyContent: "center" },
  title: { fontSize: 28, fontWeight: "800" },
  sub: { color: "#666" },
  input: {
    height: 44, borderWidth: 1, borderColor: "#ddd",
    borderRadius: 10, paddingHorizontal: 12,
  },
});
```



```
stu30@seo-B650M-PG-Lightning:~/node_rn$ nano index.js
```

```
GNU nano 7.2
import { registerRootComponent } from 'expo';
import App from './App';
registerRootComponent(App);
```

React Native

```
stu30@seo-B650M-PG-Lightning:~/node_rn$ npm pkg set scripts.start="expo start"
stu30@seo-B650M-PG-Lightning:~/node_rn$ npm pkg set scripts.start:tunnel="expo start --tunnel --port 8888"
stu30@seo-B650M-PG-Lightning:~/node_rn$ npm pkg set scripts.android="expo run:android"
stu30@seo-B650M-PG-Lightning:~/node_rn$ npm pkg set scripts.web="expo start --web"
stu30@seo-B650M-PG-Lightning:~/node_rn$ npm run start:tunnel
```

React Native

- Expo go -> 런타임 뷰어,

```
Starting project at /data/stu30/node_rn
Starting Metro Bundler
Tunnel connected.
Tunnel ready.
```



A square QR code with a black and white pixelated pattern, used for scanning with the Expo Go app.

```
> Metro waiting on exp://d6lg83g-anonymous-8888.exp.direct
> Scan the QR code above with Expo Go (Android) or the Camera app (iOS)

> Using Expo Go
> Press s | switch to development build

> Press a | open Android
> Press w | open web

> Press j | open debugger
> Press r | reload app
> Press m | toggle menu
> shift+m | more tools
> Press o | open project code in your editor

> Press ? | show all commands

Logs for your project will appear below. Press Ctrl+C to exit.
```

Q & A
