

이 튜토리얼에서는 Node.js를 사용하여 간단한 API를 만들고, HTML 페이지에서 해당 API를 호출하는 방법을 설명하겠습니다. 이 과정에서 Express.js를 사용하여 API 서버를 구축하고, Node.js에서 HTML 파일을 서빙(Serving)하는 방법도 함께 다룰 것입니다.

NVM을 설치하고 이를 이용해 Node를 설치하는 방법은 생략합니다.

## 1. Node.js와 Express 설치

먼저 Node.js와 npm(Node Package Manager)이 설치되어 있어야 합니다. Node.js 설치가 완료된 후, 프로젝트 디렉토리를 생성하고, Express.js를 설치합니다.

```
mkdir simple-api-project  
cd simple-api-project  
npm init -y  
npm install express
```

## 2. 간단한 API 서버 만들기

이제 Express를 사용하여 간단한 API를 만듭니다. API 서버는 `server.js` 파일에 작성됩니다.

## 2.1 server.js 파일 작성

server.js 파일을 생성하고 다음 코드를 작성합니다:

```
const express = require('express');
const path = require('path');
const app = express();
const port = 3000;

// JSON 데이터를 제공하는 간단한 API 엔드포인트
app.get('/api/greet', (req, res) => {
  res.json({ message: 'Hello, World!' });
});

// HTML 파일을 서빙하는 엔드포인트
app.get('/', (req, res) => {
  res.sendFile(path.join(__dirname, 'index.html'));
});

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

위 코드에서:

- `/api/greet` 엔드포인트는 JSON 형식으로 데이터를 반환하는 간단한 API입니다.
- `/` 엔드포인트는 `index.html` 파일을 클라이언트에게 서빙합니다.
- 서버는 `3000` 포트에서 실행됩니다.

### 3. HTML 파일에서 API 호출하기

이제 Node.js 서버에서 서빙할 `index.html` 파일을 작성합니다. 이 파일에서는 JavaScript를 사용하여 API를 호출하고, 결과를 페이지에 표시합니다.

## 3.1 index.html 파일 작성

index.html 파일을 생성하고 다음과 같이 작성합니다:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>API 호출 예제</title>
</head>
<body>

  <h1>API 호출 예제</h1>
  <button id="callApiButton">API 호출</button>
  <p id="responseText">여기에 API 응답이 표시됩니다.</p>

  <script>
    document.getElementById('callApiButton').addEventListener('click', function() {
      fetch('/api/greet')
        .then(response => response.json())
        .then(data => {
          document.getElementById('responseText').innerText = data.message;
        })
        .catch(error => console.error('Error:', error));
    });
  </script>

</body>
</html>
```

위 코드에서:

- `callApiButton` 버튼을 클릭하면 `/api/greet` API 엔드포인트를 호출합니다.
- `fetch` 함수로 API를 호출하고, 응답을 JSON으로 파싱하여 `responseText` 요소에 표시합니다.



## 4. Node.js 서버 실행 및 테스트

이제 API 서버를 실행하고, 웹 브라우저에서 결과를 확인할 수 있습니다.

```
node server.js
```

서버가 실행되면, 브라우저에서 `http://localhost:3000` 에 접속하여 HTML 페이지를 확인합니다. "API 호출" 버튼을 클릭하면 API에서 반환된 메시지가 화면에 표시됩니다.



localhost:3000

# API 호출 예제

API 호출

여기에 API 응답이 표시됩니다.



localhost:3000

# API 호출 예제

API 호출

Hello, World!

## 5. 결론

이 튜토리얼에서는 Node.js와 Express.js를 사용하여 간단한 API를 만들고, HTML 파일에서 해당 API를 호출하는 방법을 배웠습니다. 이 기본적인 구조를 바탕으로 더 복잡한 API를 만들고 다양한 클라이언트에서 API를 호출하는 방식으로 확장할 수 있습니다. Express.js를 활용하면 간단한 API에서부터 복잡한 백엔드 시스템까지 다양한 서버 애플리케이션을 개발할 수 있습니다.