



fetch API

Fetch 함수 개요

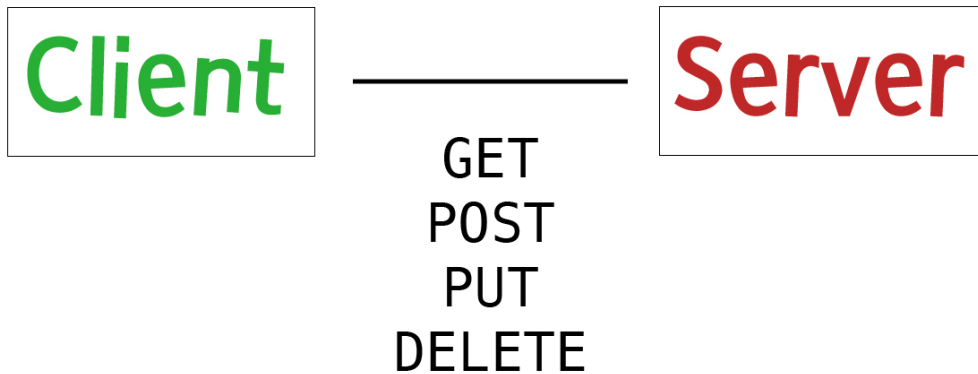
- *피치(Fetch)**는 데이터를 가져오거나 서버에 데이터를 보내는 데 사용됩니다. 주로 **프론트엔드**와 **백엔드** 간의 통신에서 사용되며, 비동기적 네트워크 요청을 처리하는 함수입니다. Fetch 함수는 JavaScript의 내장 함수로 HTTP 요청을 다루는 데 사용되며, Promise 기반의 비동기 통신을 지원합니다.

Fetch 함수 기본 구조

```
fetch('URL', { method: 'GET' })  
  .then(response => response.json())  
  .then(data => console.log(data))  
  .catch(error => console.error('Error:', error));
```

주요 HTTP 메소드

HTTP Method



1. GET 메소드:

- 주로 데이터를 가져올 때 사용됩니다.
- URL에 **key**와 **value** 값이 함께 전달되어 보안상 불리할 수 있습니다.
- 예시: 게시글 목록을 서버에서 받아올 때.

2. POST 메소드:

- 데이터를 생성하거나 서버로 보낼 때 사용됩니다.
- 데이터를 URL에 노출시키지 않고 ****본문(Body)****에 데이터를 넣어 전송하여 보안이 강화됩니다.
- 예시: 회원가입 시 사용자 데이터를 서버로 보낼 때.

3. PUT 메소드:

- 데이터를 업데이트할 때 사용됩니다.
- POST와 마찬가지로 Body에 데이터를 담아 보내므로 보안이 강화됩니다.

4. DELETE 메소드:

- 데이터를 삭제할 때 사용됩니다.

- URL에 데이터를 삭제할 리소스를 지정하여 서버에서 데이터를 삭제할 수 있습니다.

Promise와 Fetch 함수

Fetch 함수는 **Promise 객체**를 반환합니다. 이는 데이터 요청이 성공하면 **resolve**로 처리되고, 실패하면 **reject**로 처리됩니다. Promise는 비동기적인 데이터 통신에서 성공과 실패를 처리할 수 있는 유용한 구조입니다.

- **then()**: 요청이 성공했을 때 실행됩니다.
- **catch()**: 요청이 실패했을 때 실행됩니다.

Fetch 사용 예시

```
// GET 요청 예시
fetch('<https://jsonplaceholder.typicode.com/posts>')
  .then(response => response.json())
  .then(data => console.log(data))
  .catch(error => console.error('Error:', error));
```

Fetch와 JSON

서버와 통신할 때, 데이터를 주고받는 형식은 대부분 ****JSON (JavaScript Object Notation)****을 사용합니다. JSON은 데이터를 키와 값 형태로 표현하는 객체 기반의 데이터 형식입니다.
