



Full-Stack Service Programming

Lecture 2

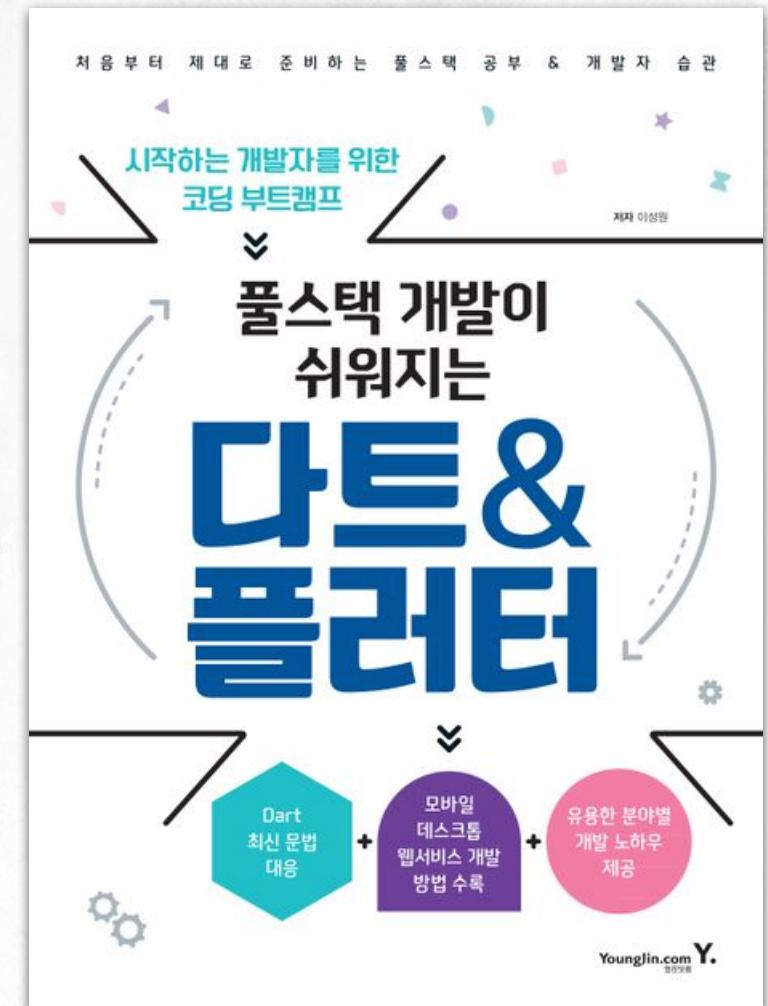
# RESTful API Client & Server 개발

2023. 09. 01

Sungwon Lee

Department of Software Convergence

● Volume.D HTTP 서버와 클라이언트 개발



# TEXT BOOK & SOURCE CODE

## 소스 코드

- <https://github.com/drsungwon/DART-FLUTTER-BOOK>

The screenshot shows a GitHub repository page for the user drsungwon named DART-FLUTTER-BOOK. The repository is public and has 1 branch and 0 tags. The 'Code' tab is selected. A commit from drsungwon dated May 9 at 62b9a46 is shown, updating the README.md file. The repository has 12 commits in total. The 'About' section includes a description in Korean: '플스택 개발이 쉬워지는 다트 & 플러터 (영진단 캠)'. It also lists the Readme, MIT license, 3 stars, 1 watching, and 1 fork. There is a link to report the repository. The 'Releases' section indicates no releases have been published.

GitHub - drsungwon/DART-FLUTTER-BOOK

Product Solutions Open Source Pricing

Search Sign in Sign up

drsunwon / DART-FLUTTER-BOOK Public

Notifications Fork 1 Star 3

Code Issues Pull requests Actions Projects Security Insights

main 1 branch 0 tags Go to file Code

drsunwon Update README.md 62b9a46 on May 9 12 commits

BOOKTITLE Add files via upload 2 months ago

volume-B-chapter-02 Add files via upload 2 months ago

volume-B-chapter-03 Add files via upload 2 months ago

volume-B-chapter-04 Add files via upload 2 months ago

volume-B-chapter-05 Add files via upload 2 months ago

volume-B-chapter-06 Add files via upload 2 months ago

volume-B-chapter-07 Add files via upload 2 months ago

volume-B-chapter-08 Add files via upload 2 months ago

About

플스택 개발이 쉬워지는 다트 & 플러터 (영진단 캠)

Readme

MIT license

3 stars

1 watching

1 fork

Report repository

Releases

No releases published

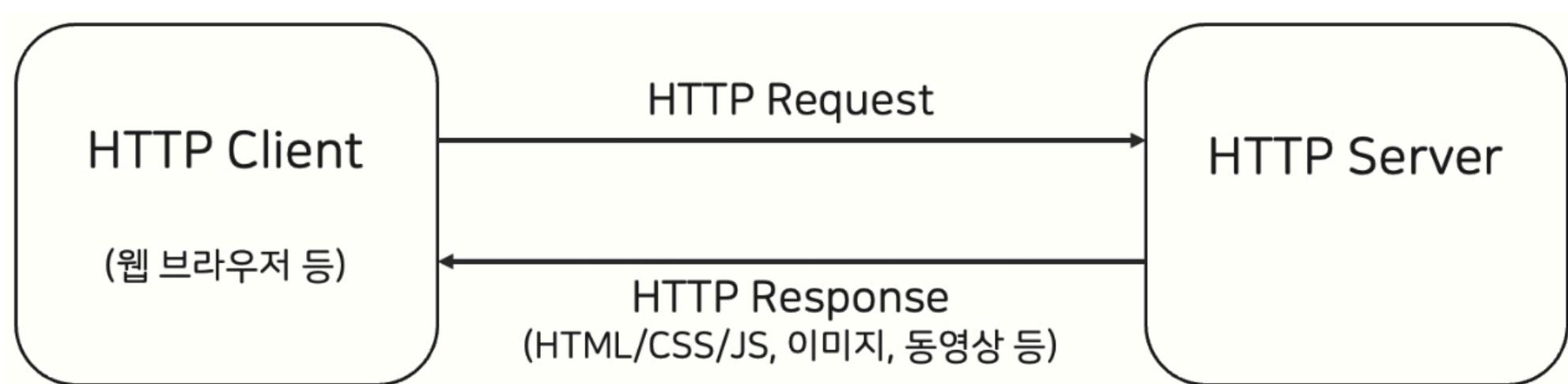
## HTTP 프로토콜 이해하기

---

- 수업에 앞서

- HTTP Client & Server는 DartPad에서 실행이 안되며, Native/App/Desktop 환경에서 동작함
- 웹 파이션 프로그래밍 수업의 IP 주소, Port 번호, HTTP 프로토콜의 기본 개념 등을 이해함

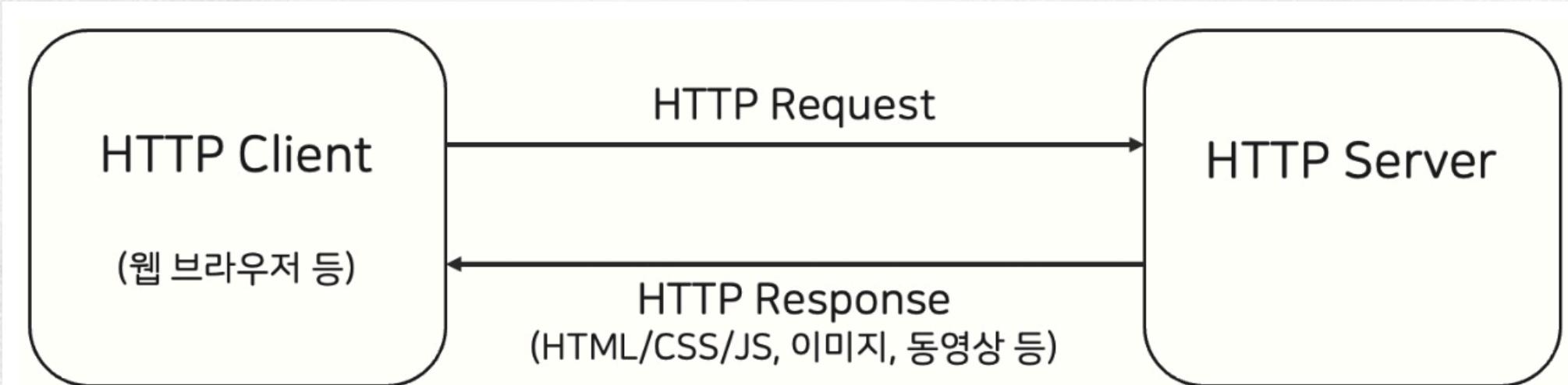
# HTTP Request (Methods) & Response



[그림 6] HTTP Request/Response 개념도

HTTP method	Description
GET	Send named resource from the server to the client.
PUT	Store data from client into a named server resource.
DELETE	Delete the named resource from a server.
POST	Send client data into a server gateway application.
HEAD	Send just the HTTP headers from the response for the named resource.

# HTTP Request (Methods) & Response

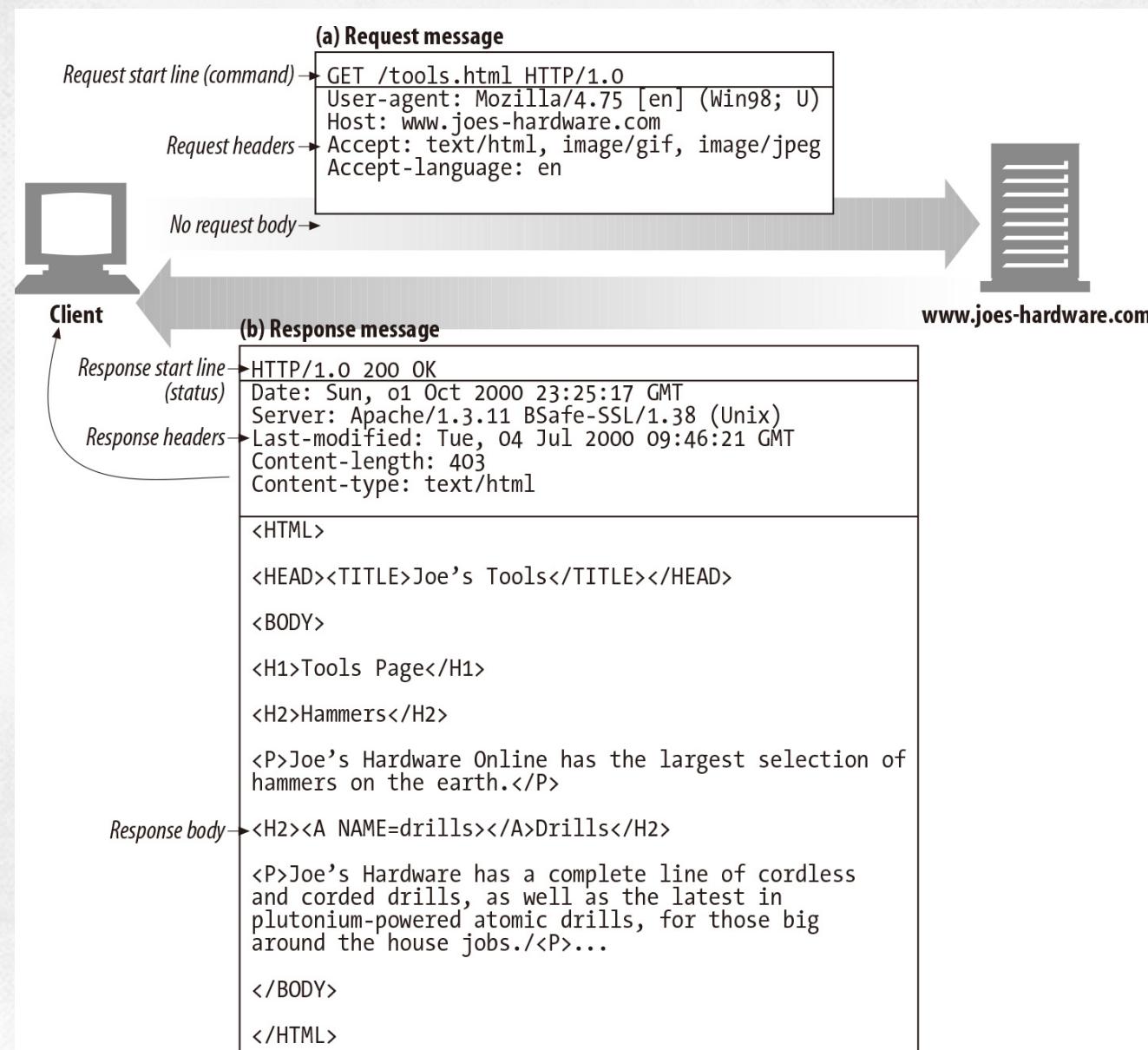


[그림 6] HTTP Request/Response 개념도

HTTP status code	Description
200	OK. Document returned correctly.
302	Redirect. Go someplace else to get the resource.
404	Not Found. Can't find this resource.

# VOLUME.D CHAPTER.2

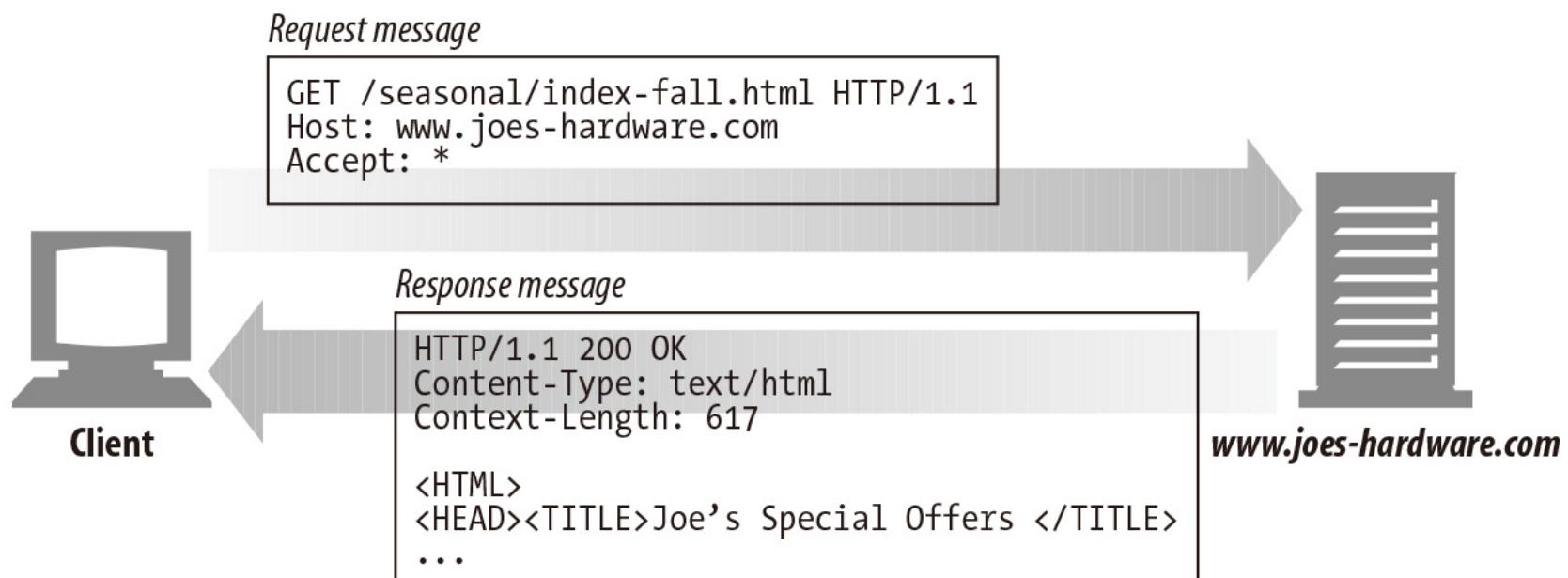
## HTTP Request & Response 예시



## HTTP GET

### ❑ GET

- The most common method
- Ask a server to send a resource

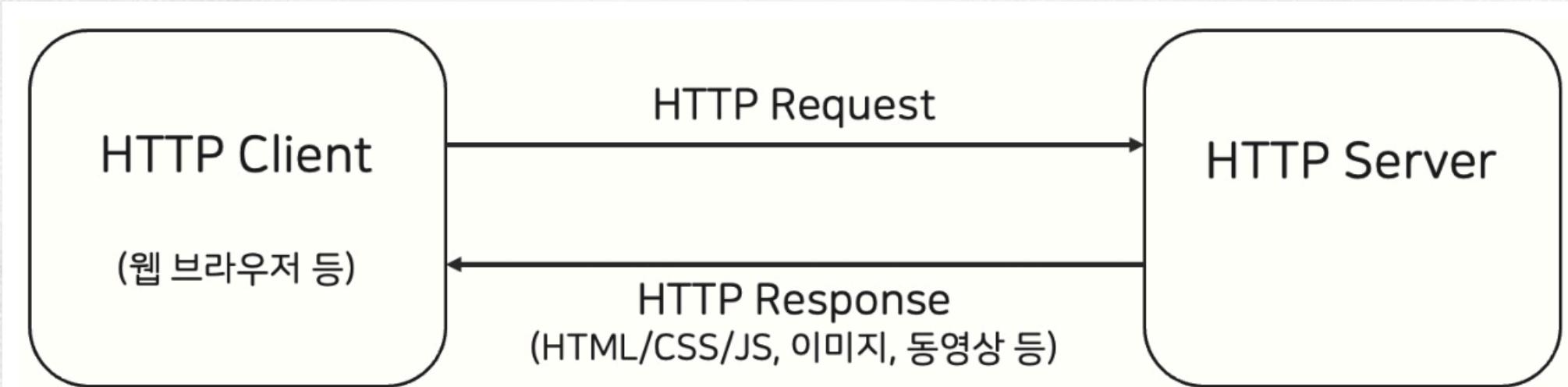


## HTTP GET 프로그램 이해하기 (리뷰 및 실습)

---

- Step.1 : 같은 디렉토리에 위치함
  - volume-D-chapter-02.dart
  - sample.txt
- Step.2 : volume-D-chapter-02.dart 실행함
- Step.3 : 웹 브라우저 실행 및 URL 창에 다음 내용 입력 & 실행함
  - 127.0.0.1:4040
  - 127.0.0.1:4040/add,4,3
  - 127.0.0.1:4040/sample.txt
  - 127.0.0.1:4040/applemango

# HTTP Client & Server 개발하기



[그림 6] HTTP Request/Response 개념도

HTTP method	Description
GET	Send named resource from the server to the client.
PUT	Store data from client into a named server resource.
DELETE	Delete the named resource from a server.
POST	Send client data into a server gateway application.
HEAD	Send just the HTTP headers from the response for the named resource.

## HTTP PUT

### □ PUT

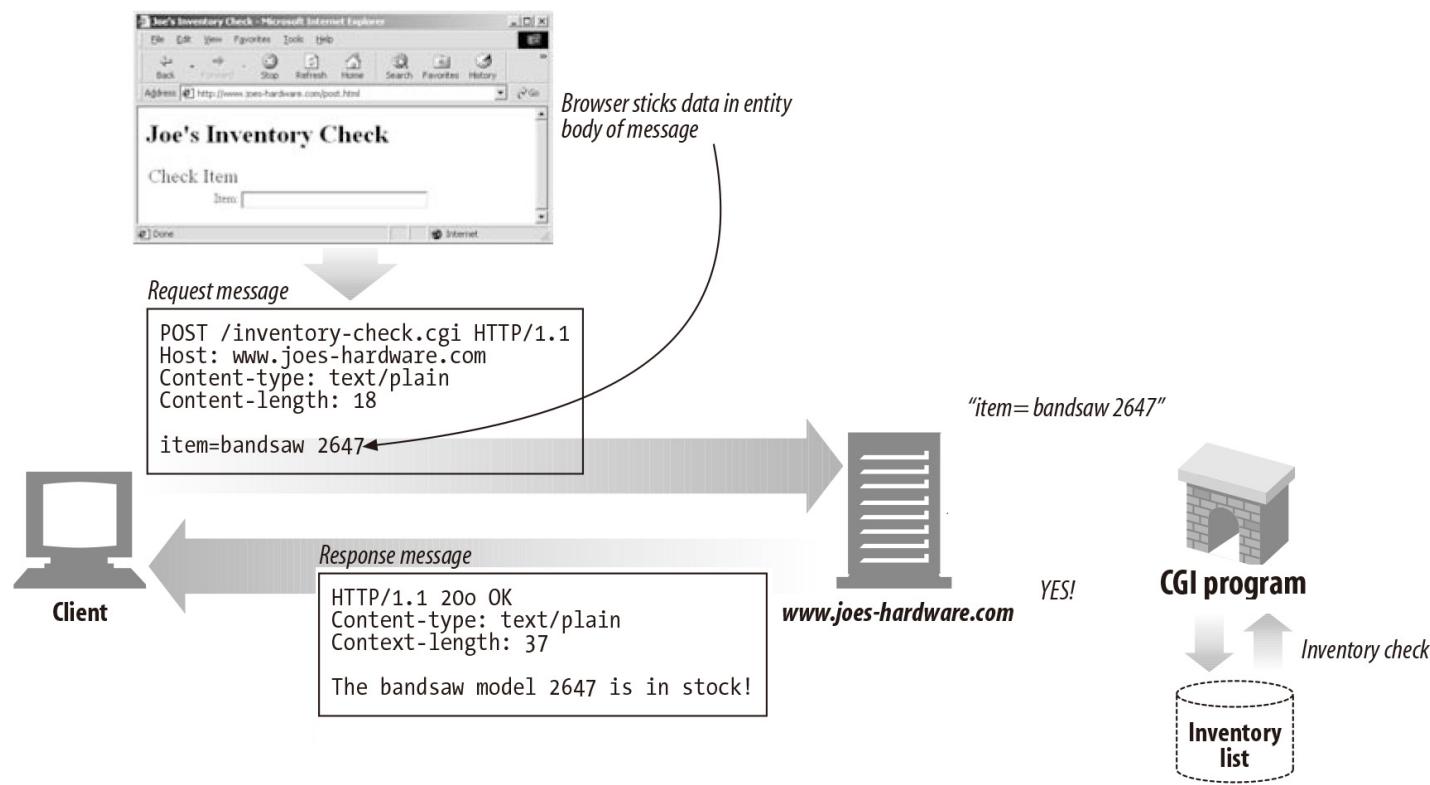
- Writes documents to a server
- Many servers require password before performing a PUT



# VOLUME.D CHAPTER.3

## HTTP POST

- POST
  - Send input data to the server

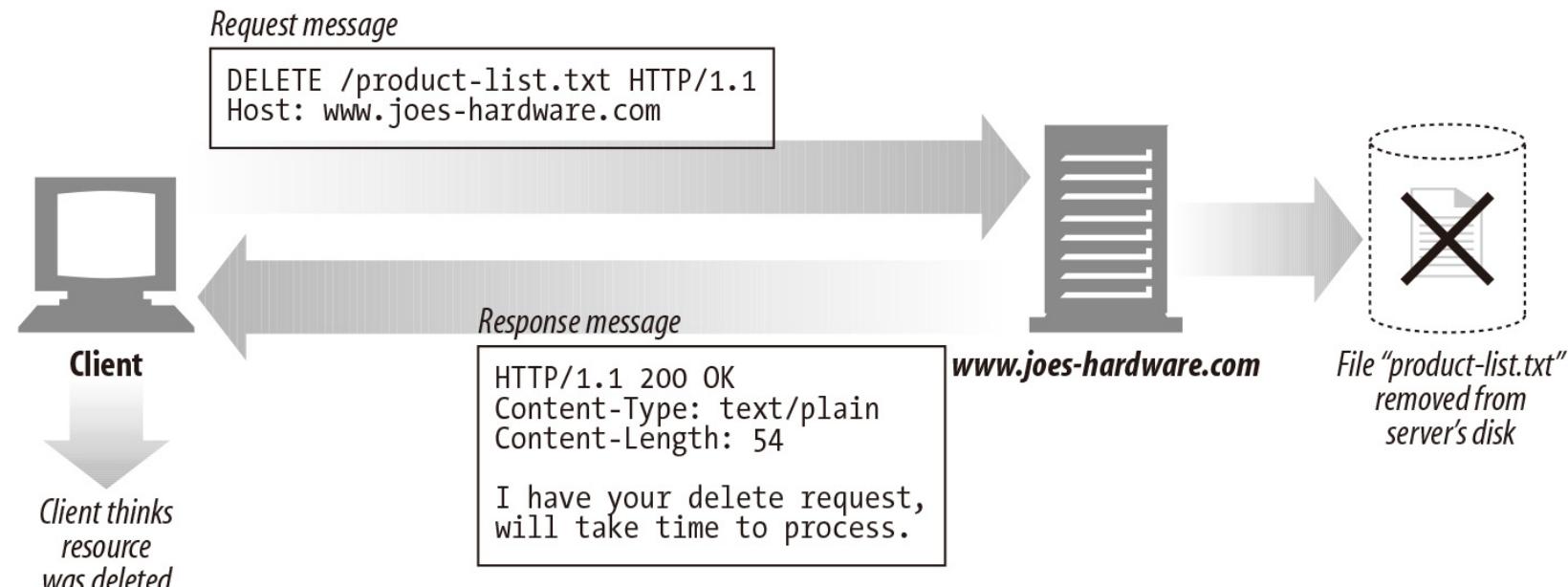


# VOLUME.D CHAPTER.3

## HTTP DELETE

### ❑ DELETE

- Ask the server to delete the resource specified by request URL



## HTTP Client & Server 개발하기 (리뷰 & 실습)

---

- 수행 결과 사전 리뷰

- volume-D-chapter-03-client-result.txt
- volume-D-chapter-03-server-result.txt

- Server 실행 (CLI 환경 권장)

- volume-D-chapter-03-server.dart
- \* sample.txt는 서버 소스코드와 같은 디렉터리에 위치함

- Client 실행 (Visual Code 환경 권장)

- volume-D-chapter-03-client.dart
- \* 서버와 같은 컴퓨터에서 실행함

## ● JavaScript Object Notation

- JS 언어의 데이터 표현 방식으로 다른 언어들에 확산됨
- 웹 브라우저는 JS 번역기를 내장하기에 웹 서비스에 유리함
- 문자열로 이루어져 있으며, JSON 객체로 변환이 가능함
- 확장자 .json과 타입 application/json 사용이 활용됨
- 무상태, 실시간 서버 대 브라우저 통신 프로토콜로 많이 활용됨
- 기본적인 프로그래밍 언어의 데이터 형태를 지원함
  - \* 정수, 실수 (고정/부동 소수점), 문자열
  - \* 배열 : [ a, b, c ]
  - \* 객체 : { "member1" : "value1", "member2" : "value2" }
- 순수한 '데이터 포맷'임 (메서드 표현 불가)
- 문자열과 프로퍼티의 이름 작성시 큰 따옴표만을 사용해야 함

## JSON example

```
{
  "squadName": "Super hero squad",
  "homeTown": "Metro City", ← superHeroes['homeTown']
  "formed": 2016,
  "secretBase": "Super tower",
  "active": true, ← superHeroes['active']
  "members": [
    {
      "name": "Molecule Man",
      "age": 29,
      "secretIdentity": "Dan Jukes",
      "powers": [
        "Radiation resistance",
        "Turning tiny",
        "Radiation blast"
      ]
    },
    {
      "name": "Madame Uppercut",
      "age": 39,
      "secretIdentity": "Jane Wilson",
      "powers": [
        "Million tonne punch",
        "Damage resistance",
        "Superhuman reflexes" ← superHeroes
[ 'members'][1][ 'powers'][2]
      ]
    },
    {
      "name": "Eternal Flame",
      "age": 1000000,
      "secretIdentity": "Unknown",
      "powers": [
        "Immortality",
        "Heat Immunity",
        "Inferno",
        "Teleportation",
        "Interdimensional travel"
      ]
    }
  ]
}
```

superHeroes

- superHeroes['homeTown']

- superHeroes['active']

superHeroes  
['members'][1]['powers'][2]

# VOLUME.D CHAPTER.4

## JSON 활용하기

The image shows two separate browser windows displaying the Dart API documentation for the `dart:convert` library.

**Left Window:** The title bar says "JSON and serialization | Flutter" and "dart:convert library - Dart API". The URL is <https://api.dart.dev/stable/3.0.5/dart-convert/dart-convert-library.html>. The page title is "dart:convert library". It describes encoders and decoders for converting between different data representations, including JSON and UTF-8. It includes code snippets for using the library and lists classes like `AsciiCodec`.

**Right Window:** The title bar says "JSON and serialization | Flutter" and "json constant - dart:convert library". The URL is <https://api.dart.dev/stable/3.0.5/dart-convert/json-constant.html>. The page title is "json top-level constant". It describes the `JsonCodec const json` constant, which is an instance of the default implementation of `JsonCodec`. It provides convenient access to most common JSON use cases. Examples show how to use `jsonEncode` and `jsonDecode` functions.

# VOLUME.D CHAPTER.4

## JSON 활용하기

The screenshot shows a web browser window with the URL <https://docs.flutter.dev/data-and-backend/json>. The page title is "JSON and serialization". The left sidebar has a "Data & backend" section expanded, showing "State management", "Networking & http", "JSON and serialization" (which is the current page), "Firebase", and "Google APIs". The main content area starts with a paragraph about the necessity of JSON for mobile apps. It then describes the guide's purpose: to look into ways of using JSON with Flutter across different scenarios. A callout box provides Terminology: **Encoding** and **serialization** are the same thing—turning a data structure into a string. **Decoding** and **deserialization** are the opposite process—turning a string into a data structure. The page also notes that **serialization** commonly refers to the entire process of translating data structures to and from a more easily readable format. To avoid confusion, the doc uses "serialization" for the overall process and "encoding" and "decoding" for specific processes. The right sidebar lists several related topics: Which JSON serialization method is right for me?, Use manual serialization for smaller projects, Use code generation for medium to large projects, Is there a GSON/Jackson /Moshi equivalent in Flutter?, Serializing JSON manually using dart:convert, Serializing JSON inline, Serializing JSON inside model classes, and Serializing JSON using code.

# JSON and serialization

Data & backend > JSON and serialization

It is hard to think of a mobile app that doesn't need to communicate with a web server or easily store structured data at some point. When making network-connected apps, the chances are that it needs to consume some good old JSON, sooner or later.

This guide looks into ways of using JSON with Flutter. It covers which JSON solution to use in different scenarios, and why.

**Terminology:** *Encoding* and *serialization* are the same thing—turning a data structure into a string. *Decoding* and *deserialization* are the opposite process—turning a string into a data structure. However, *serialization* also commonly refers to the entire process of translating data structures to and from a more easily readable format.

To avoid confusion, this doc uses “*serialization*” when referring to the overall process, and “*encoding*” and “*decoding*” when specifically referring to those processes.

## Contents

- Which JSON serialization method is right for me?
- Use manual serialization for smaller projects
- Use code generation for medium to large projects
- Is there a GSON/Jackson /Moshi equivalent in Flutter?
- Serializing JSON manually using `dart:convert`
- Serializing JSON inline
- Serializing JSON inside model classes
- Serializing JSON using code

## JSON 활용하기 (리뷰 & 실습)

---

- 수행 결과 사전 리뷰

- volume-D-chapter-04-client-result.txt
  - volume-D-chapter-04-server-result.txt
- \* Dart에서 JSON은 Map으로 변환됨

- Server 실행 (CLI 환경 권장)

- volume-D-chapter-04-server.dart

- Client 실행 (Visual Code 환경 권장)

- volume-D-chapter-04-client.dart

\* 서버와 같은 컴퓨터에서 실행함

# REST API 기반 CRUD 개발하기

---

- REST

- REpresentational State Transfer
- 웹에서 데이터를 전송하고 처리하는 방법을 정의하는 철학

- RESTful API

- REST 철학을 어떻게 실행할 것인지를 정의하는 구현 가능 규격

# RESTful API 구현을 위한 CRUD 방법론

- JSON 포맷으로 데이터를 주고받음
- 통상 CRUD로 불리는 아래의 원칙에 따라서 HTTP 메서드를 사용
  - Create : POST를 사용하여 새로운 데이터를 생성
  - Read : GET을 사용하여 정보를 읽음
  - Update : PUT을 사용하여 기존에 이미 만들어진 정보를 수정
  - Delete : DELETE를 사용하여 이미 만들어진 정보를 삭제
- 웹 주소(URL)를 /{기능 이름}/{Key 값}의 형태로 작성하여 서버가 제공하는 기능 중 특정 기능을 요청하고, 기능의 대상이 되는 데이터는 Key에 의해 식별
  - 예를 들어, /membership/0001이면 가입자 '멤버십'에 관한 서비스이며, '0001' 가입자에 C/R/U/D 작업을 수행

# REST API 기반 CRUD 개발하기 (리뷰 및 실습)

---

- 수행 결과 사전 리뷰

- volume-D-chapter-05-client-result.txt
- volume-D-chapter-05-server-result.txt

- Server 실행 (CLI 환경 권장)

- volume-D-chapter-05-server.dart

- Client 실행 (Visual Code 환경 권장)

- volume-D-chapter-05-client.dart

- \* 서버와 같은 컴퓨터에서 실행함



**Thank you**