

초격차 패키지 Online.

Chapter 7.

Todo 앱

- http 통신

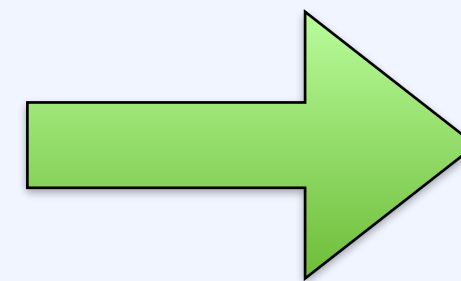
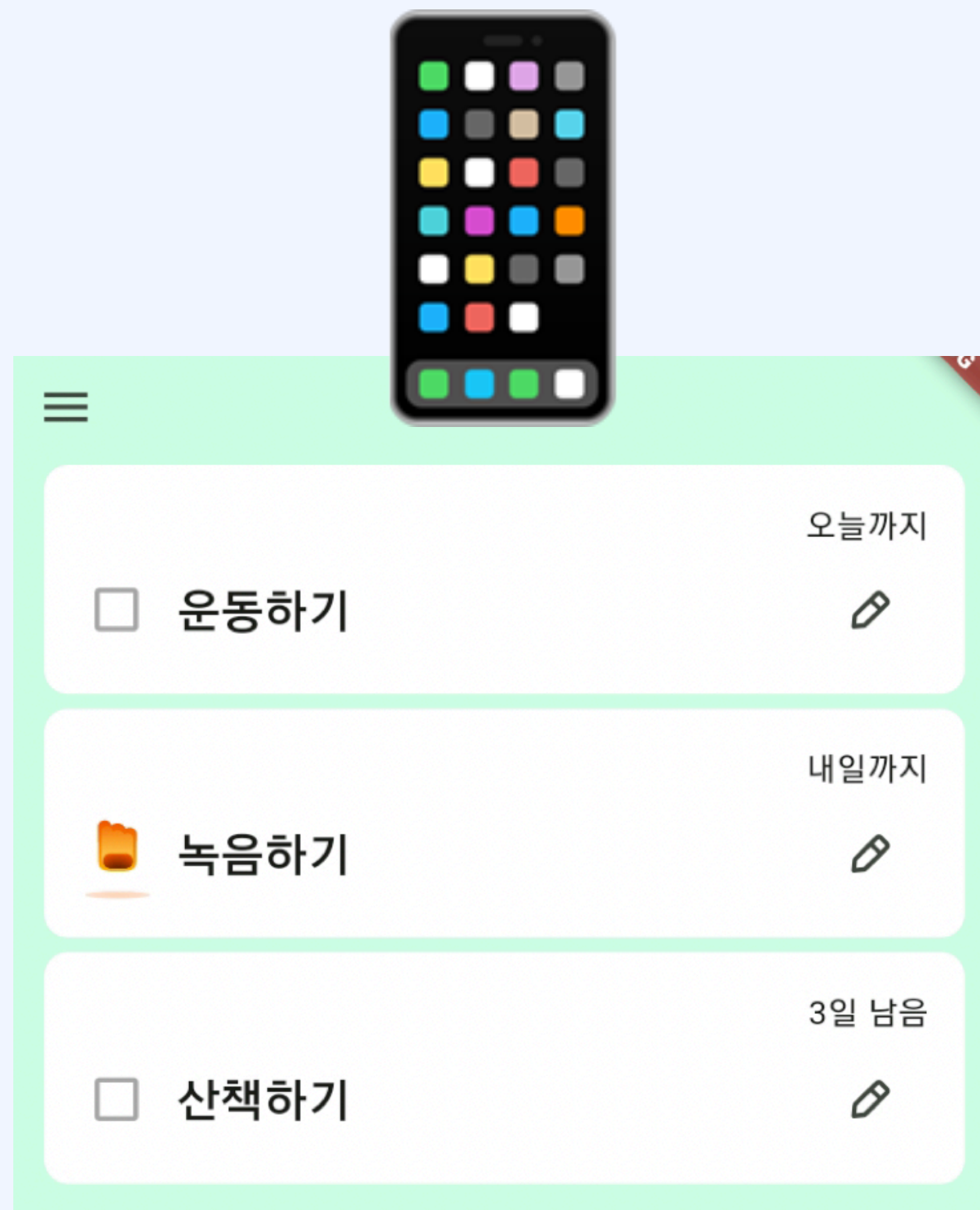
PART3 | 활용

1. 서버와 통신하기
2. Flutter에서 제공하는 통신의 종류
3. Todo 콘텐츠를 서버에 저장해보자
4. API Client 코드 최적화 하기

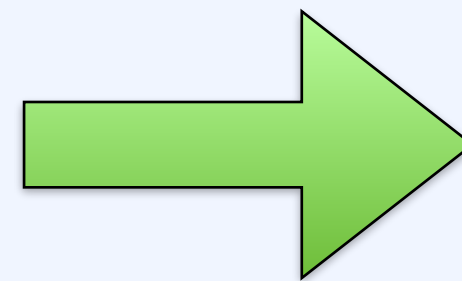
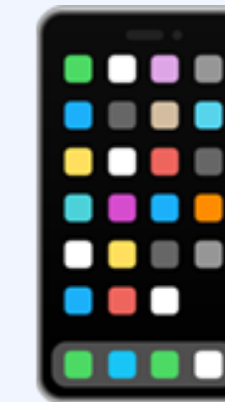
Todo 앱 - http 통신

1 서버와 통신하기

저장하는 방법



1. Local

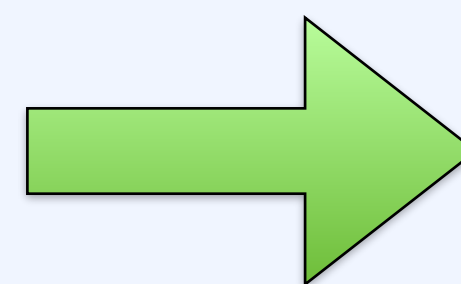
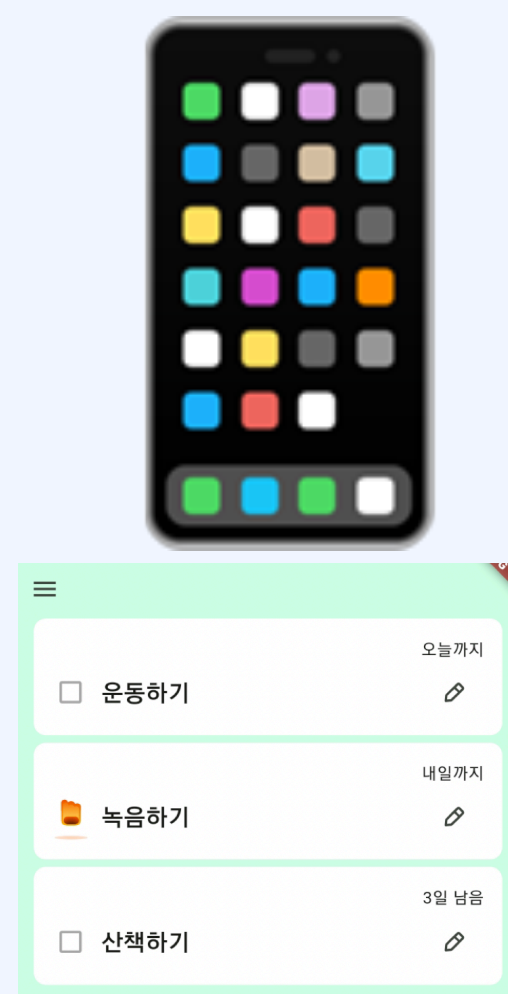


2. Network - 별도의 서버



Dart 함수 명령
=> Http 통신으로 데이터 전달
=> 원격에 있는 서버 DB파일에 저장

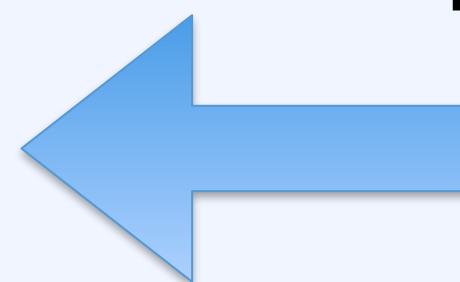
HTTP

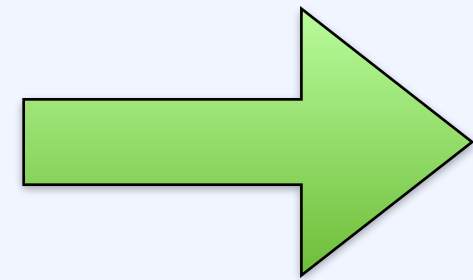


HTTP Request



HTTP Response





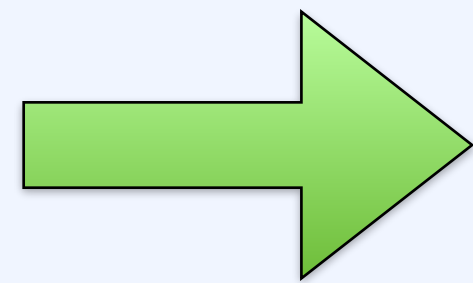
HTTP Request

1. Start Line

- Method: GET, POST, PUT, DELETE, HEAD, PATCH, CONNECT, OPTIONS, TRACE
- Request Target: 전송 주소
- HTTP Version: 1.1 / 2.0 / 3.0 등

GET - http://{host}/path/todo
=> Data를 받아옴

POST - http://{host}/path/todo
=> Data를 추가함



HTTP Request

2. Header

- **User-Agent**

Mozilla/5.0 (Windows NT 10.0; Win64; x64)

AppleWebKit/537.36 (KHTML, like Gecko)

PostmanRuntime/7.26.5

- **Authorization:** 로그인 인증 토큰

- **Cookie:** Key-Value 형태의 긴 여러 데이터

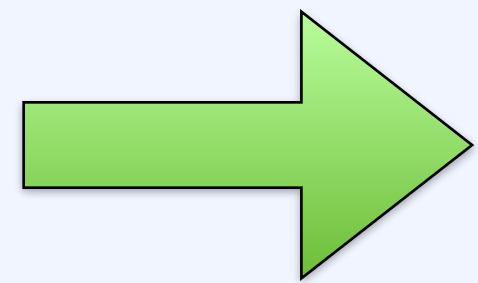
- **Content-Type:** 보낸 Body Data의 데이터 유형

text/html; charset=utf-8

text/xml

image/png

application/json

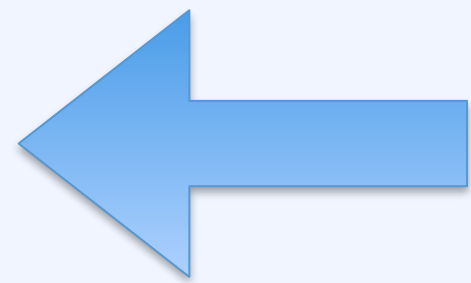


HTTP Request

3. Body: Content-Type에서 선언한 데이터

- empty (비어있는 채로)
- text
- Json
- html
- xml
- image
- file
- etc...

HTTP Response



1. Start Line

- **Status Code:**

200: 성공

404: Not Found (존재하지 않음)

401: 권한 없음 (로그인 정보 없음)

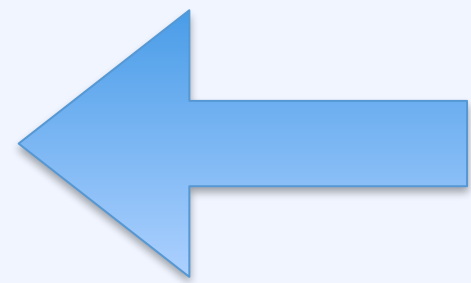
500: 서버 내부 에러

... (<https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>)

- **Status Text: Status Code에 대한 설명**

- **HTTP Version: 1.1 / 2.0 /3.0 등**

HTTP Response



2. Header

- **Server:** 서버 종류
- **Content-Disposition:** 내용을 어떻게 보여줄지 안내
- **Cookie:** Key-Value형태의 긴 여러 데이터

3. Body: HTTP Request body와 동일

REST API

앱개발자 => 서버 개발자
RESTful한 형태로 수정 부탁드립니다.

REST API

Representational State Transfer Application Programming Interface

"Architectural Styles and the Design of Network-based Software Architectures", 2000

Roy Thomas Fielding

<https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>

REST API

이해하기 쉬운 & 사용하기 쉬운

API 주소/구조 설계

Base Url: https://api.todo.com

1. TODO 리스트 전부 요청

GET https://api.todo.com/todos

2. TODO id로 1개만 요청

GET https://api.todo.com/todos/{todo_id}

3. TODO 신규 입력 요청

POST https://api.todo.com/todos

4. TODO id(주소) + 컨텐츠(Body) 로 1개 수정 요청

PUT https://api.todo.com/todos/{todo_id}

5. TODO id로 삭제 요청

DELETE https://api.todo.com/todos/{todo_id}