

# Flutter ChatGPT 앱 나만의 비서앱

3 리뷰 및 돌아보기

## 돌아보기

### 학습목표

1. Flutter Framework를 사용하여 ChatGPT 앱을 구현한다.
2. ChatGPT API를 호출하는 방법을 학습한다.
3. 간단한 애니메이션 구현을 학습한다.
4. Future, Stream 의 구현을 방법을 학습해본다.

## 돌아보기 애니메이션



```
setupAnimations() {  
  animationController = AnimationController(  
    duration: const Duration(milliseconds: 2500), vsync: this);  
  _characterCount = StepTween(begin: 0, end: _currentString.length).animate(  
    CurvedAnimation(parent: animationController, curve: Curves.easeIn));  
  animationController.addListener(() {  
    Vibration.vibrate();  
    setState(() {});  
  });  
  animationController.addStatusListener((status) {  
    if (status == AnimationStatus.completed) {  
      Future.delayed(const Duration(seconds: 1)).then((value) {  
        animationController.reverse();  
      });  
    } else if (status == AnimationStatus.dismissed) {  
      Future.delayed(const Duration(seconds: 1)).then((value) {  
        animationController.forward();  
      });  
    }  
  });  
  animationController.forward();  
}
```

## 돌아보기

### 데이터 모델 생성

○ ○ ○

```
Messages({
    required this.role,
    required this.content,
});

late final String role;
late final String content;
```

○ ○ ○

```
ChatCompletionModel({
    required this.model,
    required this.messages,
    required this.stream,
});

late final String model;
late final List<Messages> messages;
late final bool stream;
```

## 돌아보기

### http post 통신을 위한 코드

○ ○ ○

```
final url = Uri.https("api.openai.com", "/v1/chat/completions");
final resp = await http.post(
  url,
  headers: {
    "Authorization": "Bearer $apiKey",
    "Content-Type": "application/json",
  },
  body: jsonEncode(openAiModel.toJson()),
);
```

## 돌아보기

### 응답 데이터 처리

○ ○ ○

```
final jsonData = jsonDecode(utf8.decode(resp.bodyBytes)) as Map;  
String role = jsonData["choices"][0]["message"]["role"];  
String content = jsonData["choices"][0]["message"]["content"];
```

## 돌아보기

### Stream으로 요청하기

○ ○ ○

```
final byteStream = resp.stream.asyncExpand(
    (event) => Rx.timer(
        event,
        const Duration(milliseconds: 50),
    ),
);

if (resp.statusCode != 200) {
    var error = "";
    await for (final byte in byteStream) {
        final decoded = utf8.decode(byte, allowMalformed: false).trim();
        final map = jsonDecode(decoded) as Map;
        final errorMessage = map["error"]["message"] as String;
        error += errorMessage;
    }
    throw Exception("($statusCode) ${error.isEmpty ? "Bad Response" : error}");
}
```



## 더 나아가기 Next Step

1. 히스토리를 로컬 스토리지에 누적해보기
2. 이미지 생성 API 추가해보기
3. 테마 변경 해보기

구현이란 100% 정답이 없습니다.

많이 만들어보고 **반복**해보면서 시스템에 맞는 여러분만의 적합한 방법을 찾아보세요!

**끝까지 고생많으셨습니다.**

**오늘보다 더 나은 내일을 위해 화이팅입니다.**