

show_toast.dart

소스 코드

전체 코드

```
1  import 'package:flutter/material.dart';
2
3  void showToastMessage(BuildContext context, String message) {
4    final SnackBar snackBar = SnackBar(
5      content: Text(message),
6      duration: const Duration(seconds: 2), // How long the toast is displayed
7      action: SnackBarAction(
8        label: 'Undo', // Optional action
9        onPressed: () {
10          // Some code to undo the change.
11          print('Undo action pressed');
12        },
13      ),
14      behavior: SnackBarBehavior.floating, // Makes it float above bottom nav bar
15      shape: RoundedRectangleBorder( // Optional: for rounded corners
16        borderRadius: BorderRadius.circular(10.0),
17      ),
18      margin: const EdgeInsets.all(10.0), // Optional: if behavior is floating
19    );
20
21    // Find the ScaffoldMessenger in the widget tree and use it to show a SnackBar.
22    ScaffoldMessenger.of(context).showSnackBar(snackBar);
23  }
```

1 파일 개요

파일명: lib/common/utils/show_toast.dart

역할: Flutter에서 간단한 토스트 메시지를 화면에 표시하는 유틸 함수 정의

주요 목적:

- `SnackBar`를 이용해 짧은 메시지를 화면에 띄움
- 사용자가 메시지를 확인할 수 있도록 **간단한 UI** 제공
- **전역 함수** 형태로 필요 시 언제든지 호출 가능

사용 맥락 예시:

- 버튼 클릭 후 알림 메시지 표시
- 사용자 입력 완료, 오류, 성공 메시지 안내
- Flutter 화면 어디서든 간편하게 호출 가능

2 주요 기능

1. 토스트 메시지 표시

- `ScaffoldMessenger.of(context).showSnackBar(snackBar)` 사용
- 지정된 `duration` 동안 화면에 표시 (`Duration(seconds: 2)`)

2. 옵션 기능

- `action` (예: Undo 버튼) 제공 가능
- `behavior: SnackBarBehavior.floating` → 화면 아래 네비게이션 위에 떠 있도록 표시
- `shape` → 테두리 둥글게 (`borderRadius: 10`)
- `margin` → 화면 여백 설정 (`EdgeInsets.all(10)`)

3. 유연성

- 함수 매개변수로 `context` 와 `message` 전달
- 간단하게 호출 가능: `showToastMessage(context, "Hello world")`

3 구조 분석

```
1 showToastMessage(context, message)
2   └─ SnackBar snackBar = SnackBar(...)
3     └─ content: Text(message)
4     └─ duration: 2초
5     └─ action: SnackBarAction (label + onPressed)
6     └─ behavior: floating
7     └─ shape: RoundedRectangleBorder (borderRadius: 10)
8     └─ margin: EdgeInsets.all(10)
9   └─ ScaffoldMessenger.of(context).showSnackBar(snackBar)
```

특징:

- 간단하고 직관적 → SnackBar 기능만 사용
- UI 옵션 포함 → 동적 메시지 표시 가능
- 독립적 함수 → 재사용 용이

4 동작 흐름

1. 화면 위젯에서 호출

```
1 showToastMessage(context, "저장되었습니다.");
```

- 함수 내부에서 `SnackBar` 객체 생성
- 메시지와 옵션(`duration`, `action`, `shape`, `margin`) 설정
- `ScaffoldMessenger.of(context)` 로 `SnackBar` 표시
- 지정 시간(2초) 후 자동 사라짐

5 장점

- 간단한 호출: context + message만 있으면 표시 가능
- **SnackBar 기본 기능 활용**: floating, shape, action
- **UI 일관성**: margin과 둥근 모서리 적용 가능
- **재사용성**: 프로젝트 전체 어디서든 호출 가능

6 단점 / 개선점

1. 커스텀 UI 제한

- SnackBar 기본 스타일 위주 → 디자인 자유도 낮음
- 개선: Overlay나 Riverpod 기반 커스텀 토스트 구현

2. 중첩 호출 처리 미지원

- 연속 호출 시 이전 SnackBar 자동 취소되지 않음
- 개선: 기존 SnackBar 제거 후 새 메시지 표시

3. 애니메이션 / 위치 고정

- floating 동작만 가능, top 표시 불가
- 개선: Overlay 사용 시 화면 상단 또는 특정 위치 표시 가능

4. 전역 상태 연동 없음

- 메시지 상태를 Notifier/Provider로 관리하지 않음
- 개선: `toastProvider` 와 연계하여 상태 기반 표시 가능

7 개선된 구조 예시 (개념)

```
1 void showCustomToast(BuildContext context, String message, {Duration duration = const
  Duration(seconds: 2)}) {
2   // 기존 SnackBar 대신 overlay + 상태 관리 적용 가능
3 }
```

- Overlay 위젯을 사용하면 화면 상단, 중앙 등 원하는 위치 표시 가능
- 여러 토스트 메시지 큐 처리 가능
- 메시지 스타일, 배경색, 애니메이션 자유도 향상