Do it! 플러터 앱 프로그래밍

10장 **데이터베이스에 데이터 저장하기**

목차

10-1 데이터베이스 만들기

실습하기

Pub.dev 에서 sqflite 와 path 검색하여 패키지 설장하기

Pubspec.yaml 파일에 추가하기

• pubspec.yaml

```
(...생략...)
dependencies:
  flutter:
    sdk: flutter
  sqflite: ^2.0.0+3  # SQLite 관련 패키지
  path: ^1.8.0       # 내부 저장소 관련 패키지
(...생략...)
```

10-1 데이터베이스 만들기

실습하기

데이터 베이스에 사용할 클래스 생성하기

Todo 클래스 생성

Map 형태로 만들어서 데이터 저장 및 불러오기 가능

```dart
class Todo {
  String? title;
  String? content;
  int? active;
  int? id;

  Todo({this.title, this.content, this.active, this.id});

  Map<String, dynamic> toMap() {
    return {
      'id': id,
      'title': title,
      'content': content,
      'active': active,
    };
  }
}
```

10-1 데이터베이스 만들기

실습하기

import 를 이용하여 패키지 추가하기

데이터베이스 초기화하기위해
initDatabase()함수를 만든 후 코드 작성하기

```dart
import 'package:path/path.dart';
import 'package:sqflite/sqflite.dart';
```

```dart
Future<Database> initDatabase() async {
  return openDatabase(
    join(await getDatabasesPath(), 'todo_database.db'),
    onCreate: (db, version) {
      return db.execute(
        "CREATE TABLE todos(id INTEGER PRIMARY KEY AUTOINCREMENT, "
            "title TEXT, content TEXT, active BOOL)",
      );
    },
    version: 1,
  );
}
```

표 10-1 todos 테이블 구조

칼럼명	형식	비고
id	INTEGER (기본 키, 자동 증가)	순번, 기본 키, 자동 증가
title	TEXT	제목
content	TEXT	내용
active	BOOL	완료 여부

10-2 데이터베이스에서 데이터 처리하기

실습하기

insertTodo 함수 만들기

Database.insert 를 통해서

todos 라는 데이터베이스에 데이터를 넣을 수 있도록 합니다

conflictAlgorithm 의 경우 충돌났을때 알고리즘입니다

```
void _insertTodo(Todo todo) async {
  final Database database = await widget.db;
  await database.insert('todos', todo.toMap(),
      conflictAlgorithm: ConflictAlgorithm.replace);
}
```

10-2 데이터베이스에서 데이터 처리하기

실습하기

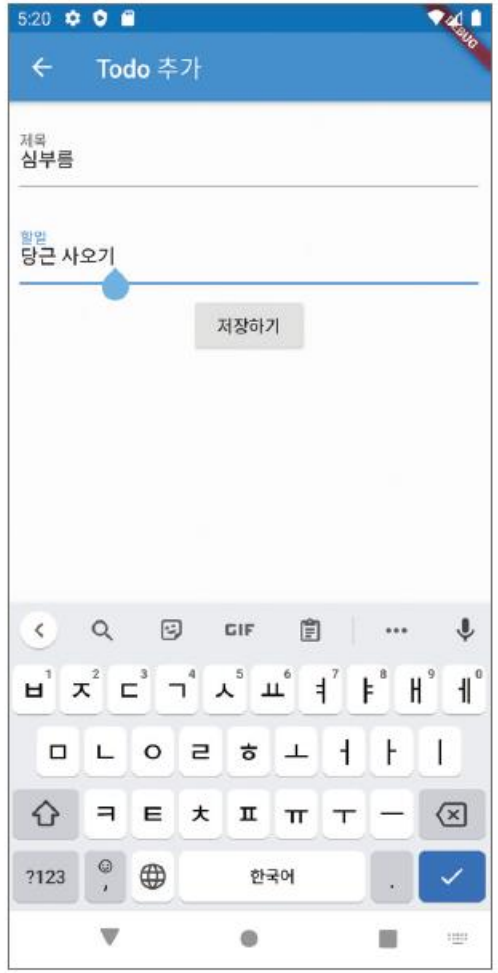
insertTodo 함수 만들기

Database.insert 를 통해서

todos 라는 데이터베이스에 데이터를 넣을 수 있도록 합니다

conflictAlgorithm 의 경우 충돌났을때 알고리즘입니다

▶ 실행 결과



10-2 데이터베이스에서 데이터 처리하기

실습하기 – 검색하기

getTodos 함수 만들기

Database.를 통해서

todos 라는 데이터베이스에 데이터를 가져와서 List형태로 만들어 주도록 합니다

• lib/main.dart

```
(...생략...)
Future<List<Todo>> getTodos() async {
  final Database database = await widget.db;
  final List<Map<String, dynamic>> maps = await database.query('todos');

  return List.generate(maps.length, (i) {
    bool active = maps[i]['active'] == 1 ? true : false;
    return Todo(
        title: maps[i]['title'].toString(),
        content: maps[i]['content'].toString(),
        active: active,
        id: maps[i]['id']);
  });
}
(...생략...)
```

10-2 데이터베이스에서 데이터 처리하기

실습하기 – 수정하기

_updateTodo 함수 만들기

Database.update 를 통해서

Todos 라는 데이터베이스에 데이터를 수정 합니다.

Where 절을 통해 지정된 컬럼값만 변화를 주도록 합니다

• lib/main.dart

```
(...생략...)
void _updateTodo(Todo todo) async {
  final Database database = await widget.db;
  await database.update(
    'todos',
    todo.toMap(),
    where: 'id = ?',
    whereArgs: [todo.id],
  );
  setState(() {
    todoList = getTodos();
  });
}
(...생략...)
```

id값으로 수정할 데이터 찾기

10-2 데이터베이스에서 데이터 처리하기

실습하기 – 삭제하기

_deleteTodo 함수 만들기

Database.delete 를 통해서

Todos 라는 데이터베이스에 데이터를 삭제 합니다

Where 절을 통해 지정된 컬럼값만 데이터를 삭제할 수 있도록 처리해줍니다

• lib/main.dart

```
(...생략...)
void _deleteTodo(Todo todo) async {
  final Database database = await widget.db;
  await database.delete('todos', where: 'id=?', whereArgs: [todo.id]);
  setState(() {
    todoList = getTodos();
  });
}
(...생략...)
```

10-3 질의문으로 추가 기능 만들기

실습하기

Query 를 이용하여 직접 데이터베이스를 수정하고 다룰수도 있습니다.

Database.rawQuery를 이용하여 sql 쿼리를 작성해주도록 합니다.

```
Future<List<Todo>> getClearList() async {
  final Database database = await widget.database;
  List<Map<String, dynamic>> maps = await database
      .rawQuery('select title, content, id from todos where active=1');

  return List.generate(maps.length, (i) {
    return Todo(
        title: maps[i]['title'].toString(),
        content: maps[i]['content'].toString(),
        id: maps[i]['id']);
  });
}
```

10-3 질의문으로 추가 기능 만들기

실습하기

모든 데이터를 삭제하는 Query 입니다

Query 를 이용하여 직접 데이터베이스를 수정하고 다룰수도 있습니다.

Query를 이용하여 데이터가 변경된 이후에는

setState() 함수를 이용하여 UI에 변경이 되었다는 것을 알려야합니다

```
void _removeAllTodos() async {
  final Database database = await widget.database;
  database.rawDelete('delete from todos where active=1');
  setState(() {
    clearList = getClearList();
  });
}
```