

# s\_setting.dart

## 소스 코드

### 전체 코드

```
1  import 'package:buyoil/common/app_strings.dart';
2  import 'package:buyoil/view/widget/setting/box_setting.dart';
3  import 'package:easy_localization/easy_localization.dart';
4  import 'package:flutter/material.dart';
5  import 'package:flutter_riverpod/flutter_riverpod.dart';
6  import '../common/app_colors.dart';
7  import '../common/app_styles.dart';
8  import '../viewmodel/vm_setting.dart';
9  import '../widget/setting/btn_setting.dart';
10 import '../widget/w_header.dart';
11
12 class SettingScreen extends ConsumerStatefulWidget {
13   const SettingScreen({Key? key}) : super(key: key);
14
15   @override
16   ConsumerState<ConsumerStatefulWidget> createState() => SettingScreenState();
17 }
18
19 class SettingScreenState extends ConsumerState<SettingScreen> {
20
21   @override
22   widget build(BuildContext context) {
23     final notifier = ref.watch(settingProvider.notifier);
24     final state = ref.watch(settingProvider);
25     return Scaffold(
26       body: Container(
27         width: double.maxFinite,
28         height: double.maxFinite,
29         child: Column(
30           children: [
31             HeaderWidget(),
32             Expanded(
33               child: Column(
34                 children: [
35                   Spacer(flex: 49),
36                   Center(
37                     child: ElevatedButton(
38                       onPressed: () {},
39                       style: ButtonStyle(
40                         backgroundColor:
41 widgetStatePropertyAll(AppColors.PRIMARY),
42                         padding:
43 widgetStatePropertyAll(EdgeInsets.symmetric(horizontal: 20)),
44                         shape:
45 widgetStatePropertyAll(RoundedRectangularBorder(borderRadius: BorderRadius.circular(5)))
```

```

43         ),
44         child: Container(
45             height: 91,
46             // padding: EdgeInsets.all(20),
47
48             child: Row(
49                 mainAxisAlignment: MainAxisAlignment.min,
50                 children: [
51                     Image.asset(AppStrings.assetPath + "img_gear.png",
width: 52.5, height: 52.5,),
52                     SizedBox(width: 13.5,),
53                     Text(AppStrings.settingsMenu.tr(), style:
AppStyles.tsSetting,)
54                 ],
55             ),
56         ),
57     ),
58 ),
59 Spacer(flex: 163),
60 Container(
61     height: 203 + 31 + 91,
62     child: Row(
63         mainAxisAlignment: MainAxisAlignment.spaceBetween,
64         children: [
65             SizedBox(width: 40,),
66             Expanded(
67                 flex: 1,
68                 child: Column(
69                     mainAxisAlignment: MainAxisAlignment.min,
70                     children: [
71                         SettingBox(color: AppColors.FF868686, text:
AppStrings.oilContainerLabel.tr()),
72                         SizedBox(height: 31,),
73                         SettingBox(text: "${state.oilContainer}kg", height:
203),
74                     ],
75                 ),
76             ),
77             SizedBox(width: 28,),
78             Expanded(
79                 flex: 1,
80                 child: Column(
81                     mainAxisAlignment: MainAxisAlignment.min,
82                     children: [
83                         SettingBox(color: AppColors.FF868686, text:
AppStrings.oilContainerLabel.tr()),
84                         SizedBox(height: 31,),
85                         SettingBox(text: "Oil:${state.measureOil}g\nwater:
${state.measurewater}g", height: 203),
86                     ],
87                 ),
88             ),
89             SizedBox(width: 28,),

```

```

90         Expanded(
91             flex: 1,
92             child: Column(
93                 mainAxisAlignment: MainAxisAlignment.min,
94                 children: [
95                     SettingBox(color: AppColors.FF868686, text:
AppStrings.oilContainerLabel.tr()),
96                     SizedBox(height: 31,),
97                     SettingButton(
98                         state: state.isMotorOn,
99                         onTap: () {
100                             notifier.toggleMotor();
101                         },
102                     ),
103                 ],
104             ),
105         ),
106         SizedBox(width: 28,),
107         Expanded(
108             flex: 1,
109             child: Column(
110                 mainAxisAlignment: MainAxisAlignment.min,
111                 children: [
112                     SettingBox(color: AppColors.FF868686, text:
AppStrings.oilContainerLabel.tr()),
113                     SizedBox(height: 31,),
114                     SettingButton(
115                         state: state.isValveOn,
116                         onTap: () {
117                             notifier.toggleValve();
118                         },
119                     ),
120                 ],
121             ),
122         ),
123         SizedBox(width: 40,),
124     ],
125 ),
126 ),
127     Spacer(flex: 93),
128 ],
129 ),
130 ),
131 ],
132 ),
133 ),
134 );
135 }
136
137 void afterLayout() {
138     // ref.listenManual(SettingProvider, (_, state) {
139     //     if(state is UIStateSettingCompleted) {
140     //         context.goNamed(RouteGroup.Setting.name);

```

```

141     // }
142     // });
143   }
144 }

```

## SettingScreen 구조 분석

본 파일은 설정 화면을 구성하는 View 레이어 코드이며, Riverpod 기반 상태 관리 구조에서 `settingProvider`의 읽기 및 액션 트리거를 수행한다. UI는 장치의 오일 용기 무게, 측정된 오일/물 값, 모터 및 밸브의 동작 여부를 시각적으로 표현하고, 이에 대한 제어 버튼을 제공하는 형태로 구성된다.

본 파일은 MVVM 아키텍처 관점에서 **View 계층의 UI 렌더링 및 인터랙션 이벤트 전달만** 담당하며, 실제 로직은 `vm_setting.dart`에서 관리된다.

## 클래스 구성

### SettingScreen extends ConsumerStatefulWidget

- Riverpod `ConsumerStatefulWidget` 형태로, 상태 변화에 반응하는 UI 빌드를 수행한다.
- 외부에서 Key만 전달하며 추가 파라미터는 없다.

### SettingScreenState extends ConsumerState<SettingScreen>

- 화면 렌더링의 중심이 되는 State 클래스이다.
- Provider의 `state` 및 `notifier`를 직접 참조한다.
  - `state = ref.watch(settingProvider)`
  - `notifier = ref.watch(settingProvider.notifier)`
- UI 외의 비즈니스 로직을 포함하지 않는다.

## UI 계층 구조

다음은 전체 UI 구조 트리다:

```

1 Scaffold
2   └ Container (full size)
3     └ Column
4       └ HeaderWidget
5         └ Expanded
6           └ Column
7             └ Spacer(flex: 49)
8             └ Center
9               └ ElevatedButton (설정 메뉴 버튼)
10            └ Spacer(flex: 163)
11            └ Container (설정 카드 묶음 영역)
12              └ Row
13                └ Expanded (오일 용기 용량)
14                  └ Column
15                    └ SettingBox(title)

```

```

16 |         |         |         ↳ SettingBox(value)
17 |         |         |         ↳ Expanded (오일/물 측정값)
18 |         |         |         ↳ Expanded (모터 on/off 버튼)
19 |         |         |         ↳ Expanded (밸브 on/off 버튼)
20 |         ↳ Spacer(flex: 93)

```

## 주요 구성 요소 분석

### 1) 설정 메뉴 버튼 영역

```

1 | ElevatedButton(
2 |     onPressed: () {},
3 |     style: ButtonStyle(...),
4 |     child: Row(...)
5 | )

```

- 스타일은 `PRIMARY` 색상 기반.
- 아이콘 + 텍스트 조합으로 구성.
- 현재 `onPressed`는 비어 있으며 추후 기능 확장 가능성을 고려한 placeholder 역할.

### 2) 오일 용기 용량 표시

```

1 | SettingBox(color: ..., text: AppStrings.oilContainerLabel.tr())
2 | SettingBox(text: "${state.oilContainer}kg", height: 203)

```

- 상단 박스는 제목 역할.
- 하단 박스는 실시간 상태 값 표시 역할.
- `state.oilContainer` 를 그대로 표기하므로 값 관리는 ViewModel이 담당한다.

### 3) 오일/물 측정값 표시

```

1 | SettingBox(color: ...)
2 | SettingBox(text: "Oil:${state.measureOil}g\nWater: ${state.measureWater}g")

```

- HX711 등을 통해 수집한 측정값으로 보이며, ViewModel에서 받아온 값을 그대로 반영한다.
- 단위(g)는 고정 문자열로 삽입되어 있다.

### 4) 모터 제어 버튼

```

1 | SettingButton(
2 |     state: state.isMotorOn,
3 |     onTap: () {
4 |         notifier.toggleMotor();
5 |     },
6 | )

```

- 현재 모터의 상태 값(state.isMotorOn)을 시각적으로 표현.
- onTap 시 ViewModel의 toggleMotor() 호출.
- View는 단순 이벤트 전달 기능만 수행.

## 5) 밸브 제어 버튼

```
1 SettingButton(
2   state: state.isValveOn,
3   onTap: () { notifier.togglevalve(); }
4 )
```

- 모터 제어 버튼과 동일한 구조.
- 내부의 실제 동작(on/off 명령 전송)은 ViewModel에서 수행한다.

## ViewModel 연동 구조

본 화면이 참조하는 Provider:

```
1 final notifier = ref.watch(settingProvider.notifier);
2 final state = ref.watch(settingProvider);
```

- state 는 UI가 렌더링할 표시 데이터(oilContainer, measureOil, measureWater, isMotorOn, isValveOn)를 제공한다.
- notifier 는 장치 제어 계층(toggleMotor, toggleValve)을 호출하는 entry point이다.

본 View는 비즈니스 로직을 전혀 갖지 않으며, 모든 장치 제어는 ViewModel의 책임이다.

## 아키텍처상 책임

SettingScreen의 책임은 다음 3개로 명확하게 구분된다:

### 1. 상태 데이터 표시

- ViewModel의 상태 값(oilContainer, measureOil, etc.)을 UI에 반영한다.

### 2. 사용자 입력 이벤트 전달

- 모터 on/off, 밸브 on/off 버튼 터치 이벤트를 ViewModel로 전달한다.

### 3. 정적 레이아웃 구성

- 헤더, 설정 버튼, 4개의 설정 요소 박스 등을 배치한다.

그 외:

- 장치 제어 명령 생성
  - Serial 통신 흐름 관리
  - 측정값 수집/검증
  - 비즈니스 로직
- 은 모두 ViewModel 및 Service 레이어에서 수행해야 하며, 본 View는 담당하지 않는다.

---

## 요약

---

- SettingScreen은 설정 페이지를 담당하는 **View 레이어**이다.
- Riverpod Provider를 기반으로 UI 상태를 렌더링한다.
- 모터·밸브 제어 버튼은 단순히 이벤트를 ViewModel로 전달하는 역할만 수행한다.
- 오일 용기 용량과 측정값은 ViewModel에서 전달받아 표시한다.
- 아키텍처 구조상 본 화면은 **상태 표시 + 이벤트 전달**에만 집중한다.