

router.dart

소스 코드

전체 코드

```
1  import 'package:buyoil/model/ui_state_usb_port.dart';
2  import 'package:buyoil/view/screen/s_driver.dart';
3  import 'package:buyoil/view/screen/s_opening_door.dart';
4  import 'package:buyoil/view/screen/s_setting.dart';
5  import 'package:buyoil/view/screen/s_splash.dart';
6  import 'package:buyoil/view/screen/s_step1.dart';
7  import 'package:buyoil/view/screen/s_step2.dart';
8  import 'package:buyoil/view/screen/s_step3.dart';
9  import 'package:buyoil/view/screen/s_step4.dart';
10 import 'package:buyoil/viewmodel/vm_serial_port.dart';
11 import 'package:flutter/material.dart';
12 import 'package:flutter_riverpod/flutter_riverpod.dart';
13 import 'package:go_router/go_router.dart';
14 import 'package:riverpod_annotation/riverpod_annotation.dart';
15
16 part 'router.g.dart';
17
18 final GlobalKey<NavigatorState> rootNavigatorKey = GlobalKey<NavigatorState>();
19
20 enum RouteGroup {
21   Splash("/splash", "splash"),
22   Driver("/driver", "driver"),
23   Step1("/step1", "step1"),
24   OpeningDoor("/openingDoor", "openingDoor"),
25   Step2("/step2", "step2"),
26   Step3("/step3", "step3"),
27   Step4("/step4", "step4"),
28   Setting("/setting", "setting");
29
30   final String path;
31   final String name;
32
33   const RouteGroup(this.path, this.name);
34 }
35
36
37 @Riverpod(keepAlive: true)
38 GoRouter router(Ref ref) {
39   return GoRouter(
40     initialLocation: RouteGroup.Splash.path,
41     navigatorKey: rootNavigatorKey,
42     routes: [
43       GoRoute(
44         path: RouteGroup.Splash.path,
45         name: RouteGroup.Splash.name,
```

```

46     builder: (context, state) => const SplashScreen(),
47   ),
48   GoRoute(
49     path: RouteGroup.Driver.path,
50     name: RouteGroup.Driver.name,
51     builder: (context, state) => const DriversScreen(),
52   ),
53   GoRoute(
54     path: RouteGroup.OpeningDoor.path,
55     name: RouteGroup.OpeningDoor.name,
56     builder: (context, state) => const OpeningDoorScreen(),
57   ),
58   GoRoute(
59     path: RouteGroup.Step1.path,
60     name: RouteGroup.Step1.name,
61     builder: (context, state) => const Step1Screen(),
62   ),
63   GoRoute(
64     path: RouteGroup.Step2.path,
65     name: RouteGroup.Step2.name,
66     builder: (context, state) => const Step2Screen(),
67   ),
68   GoRoute(
69     path: RouteGroup.Step3.path,
70     name: RouteGroup.Step3.name,
71     builder: (context, state) => const Step3Screen(),
72   ),
73   GoRoute(
74     path: RouteGroup.Step4.path,
75     name: RouteGroup.Step4.name,
76     builder: (context, state) {
77       double water = double.tryParse(state.uri.queryParameters["water"]??"") ??
0.0;
78       double oil = double.tryParse(state.uri.queryParameters["oil"]??"") ?? 0.0;
79       return Step4Screen(water: water, oil: oil);
80     },
81   ),
82   GoRoute(
83     path: RouteGroup.Setting.path,
84     name: RouteGroup.Setting.name,
85     builder: (context, state) => const SettingScreen(),
86   ),
87 ]
88 );
89 }

```

1. import 된 라이브러리 분석

✓ Flutter 기본

```
1 | import 'package:flutter/material.dart';
```

✓ 상태관리 (Riverpod)

```
1 | import 'package:flutter_riverpod/flutter_riverpod.dart';
2 | import 'package:riverpod_annotation/riverpod_annotation.dart';
```

- `@Riverpod` 어노테이션을 사용하여 Riverpod 코드를 자동 생성
- `router.g.dart` 파일을 자동 생성

✓ 라우팅 (GoRouter)

```
1 | import 'package:go_router/go_router.dart';
```

- 현재 Flutter에서 권장하는 Router API
- Declarative Routing

✓ 프로젝트 내부 파일 (Screens/ViewModels/Model)

```
1 | import 'package:buyoil/model/ui_state_usb_port.dart';
2 | import 'package:buyoil/view/screen/...';
3 | import 'package:buyoil/viewmodel/vm_serial_port.dart';
```

- 각 페이지들을 router에서 연결하기 위한 import

2. RouteGroup Enum 분석

```
1 | enum RouteGroup {
2 |     Splash("/splash", "splash"),
3 |     Driver("/driver", "driver"),
4 |     ...
5 | }
```

✓ 역할:

- 라우터 경로(path)와 이름(name)을 한 곳에 정리
- `enum` 기반으로 정리하여 오타, 문자열 하드코딩 방지
- 유지보수 쉬움

예:

```
1 RouteGroup.Splash.path // "/splash"
2 RouteGroup.Splash.name // "splash"
```

3. Global Navigator Key

```
1 final GlobalKey<NavigatorState> rootNavigatorKey = GlobalKey<NavigatorState>();
```

✓ 역할

- 전역 네비게이션 제어
- 특히 dialog, bottomSheet, overlay 등 띄우기 위해
`Navigator.of(rootNavigatorKey.currentContext!)`
이런 방식으로 접근 가능

메인 코드에서 사용하고 있었음:

```
1 Navigator.of(rootNavigatorKey.currentContext!).pop();
```

4. Router Provider (GoRouter + Riverpod)

```
1 @Riverpod(keepAlive: true)
2 GoRouter router(Ref ref) {
```

✓ 설명

- 라우터를 Riverpod Provider로 등록
- 전역에서 `ref.watch(routerProvider)` 로 접근 가능
- `keepAlive: true`
→ 라우터가 앱 실행 내내 dispose되지 않음

GoRouter 생성

```
1 return GoRouter(
2   initialLocation: RouteGroup.Splash.path, // 앱 시작 화면
3   navigatorKey: rootNavigatorKey,          // 전역 네비게이터
4   routes: [
5     GoRoute(...)
6   ]
7 );
```

5. 모든 GoRoute 구조 분석

각 스크린은:

```
1 GoRoute(  
2   path: "/step1",  
3   name: "step1",  
4   builder: (context, state) => const Step1Screen(),  
5 )
```

형태로 라우팅.

6. QueryParameter 처리 (Step4)

```
1 double water = double.tryParse(state.uri.queryParameters["water"] ?? "") ?? 0.0;  
2 double oil = double.tryParse(state.uri.queryParameters["oil"] ?? "") ?? 0.0;
```

✓ URL 예시:

```
1 /step4?water=10.5&oil=3.7
```

✓ 결과:

- water: 10.5
- oil: 3.7

값 전달 방식이 Query Parameter 기반.

7. 전체 주석 버전 router.dart

아래는 전체 파일을 자세한 주석과 함께 정리한 버전입니다.

```
1 // -----  
2 // 라우팅 관련 import  
3 // -----  
4 import 'package:buyoil/model/ui_state_usb_port.dart';  
5 import 'package:buyoil/view/screen/s_driver.dart';  
6 import 'package:buyoil/view/screen/s_opening_door.dart';  
7 import 'package:buyoil/view/screen/s_setting.dart';  
8 import 'package:buyoil/view/screen/s_splash.dart';  
9 import 'package:buyoil/view/screen/s_step1.dart';  
10 import 'package:buyoil/view/screen/s_step2.dart';  
11 import 'package:buyoil/view/screen/s_step3.dart';  
12 import 'package:buyoil/view/screen/s_step4.dart';  
13 import 'package:buyoil/viewmodel/vm_serial_port.dart';  
14 import 'package:flutter/material.dart';  
15
```

```

16 // 상태관리 (Riverpod)
17 import 'package:flutter_riverpod/flutter_riverpod.dart';
18 import 'package:go_router/go_router.dart';
19 import 'package:riverpod_annotation/riverpod_annotation.dart';
20
21 // Riverpod code-gen
22 part 'router.g.dart';
23
24 /// 전역 Navigator 접근 키
25 /// dialog / toast / route 이동 시 사용
26 final GlobalKey<NavigatorState> rootNavigatorKey = GlobalKey<NavigatorState>();
27
28
29 /// 모든 라우트를 enum으로 관리하여 유지보수 용이
30 enum RouteGroup {
31   Splash("/splash", "splash"),
32   Driver("/driver", "driver"),
33   Step1("/step1", "step1"),
34   OpeningDoor("/openingDoor", "openingDoor"),
35   Step2("/step2", "step2"),
36   Step3("/step3", "step3"),
37   Step4("/step4", "step4"),
38   Setting("/setting", "setting");
39
40   final String path; // URL 경로
41   final String name; // 라우트 이름
42
43   const RouteGroup(this.path, this.name);
44 }
45
46
47 /// Riverpod + GoRouter 결합된 Router Provider
48 /// 앱 전역 라우팅을 담당
49 @Riverpod(keepAlive: true)
50 GoRouter router(Ref ref) {
51   return GoRouter(
52     // 앱 시작 시 가장 먼저 보여줄 화면
53     initialLocation: RouteGroup.Splash.path,
54
55     // 모든 화면 이동을 통합 제어하기 위한 Navigator key
56     navigatorKey: rootNavigatorKey,
57
58     // 페이지 목록
59     routes: [
60       // Splash screen
61       GoRoute(
62         path: RouteGroup.Splash.path,
63         name: RouteGroup.Splash.name,
64         builder: (context, state) => const SplashScreen(),
65       ),
66
67       // Driver screen
68       GoRoute(

```

```

69     path: RouteGroup.Driver.path,
70     name: RouteGroup.Driver.name,
71     builder: (context, state) => const DriverScreen(),
72 ),
73
74 // Opening Door
75 GoRoute(
76     path: RouteGroup.OpeningDoor.path,
77     name: RouteGroup.OpeningDoor.name,
78     builder: (context, state) => const OpeningDoorScreen(),
79 ),
80
81 // Step1
82 GoRoute(
83     path: RouteGroup.Step1.path,
84     name: RouteGroup.Step1.name,
85     builder: (context, state) => const Step1Screen(),
86 ),
87
88 // Step2
89 GoRoute(
90     path: RouteGroup.Step2.path,
91     name: RouteGroup.Step2.name,
92     builder: (context, state) => const Step2Screen(),
93 ),
94
95 // Step3
96 GoRoute(
97     path: RouteGroup.Step3.path,
98     name: RouteGroup.Step3.name,
99     builder: (context, state) => const Step3Screen(),
100 ),
101
102 // Step4 - Query Parameter 사용 예시
103 GoRoute(
104     path: RouteGroup.Step4.path,
105     name: RouteGroup.Step4.name,
106     builder: (context, state) {
107         // URL에서 water, oil 값을 가져오기
108         double water = double.tryParse(state.uri.queryParameters["water"] ?? "") ??
0.0;
109         double oil = double.tryParse(state.uri.queryParameters["oil"] ?? "") ?? 0.0;
110
111         return Step4Screen(
112             water: water,
113             oil: oil,
114         );
115     },
116 ),
117
118 // Setting
119 GoRoute(
120     path: RouteGroup.Setting.path,

```

```
121         name: RouteGroup.Setting.name,  
122         builder: (context, state) => const SettingScreen(),  
123     ),  
124 ]  
125 );  
126 }
```


