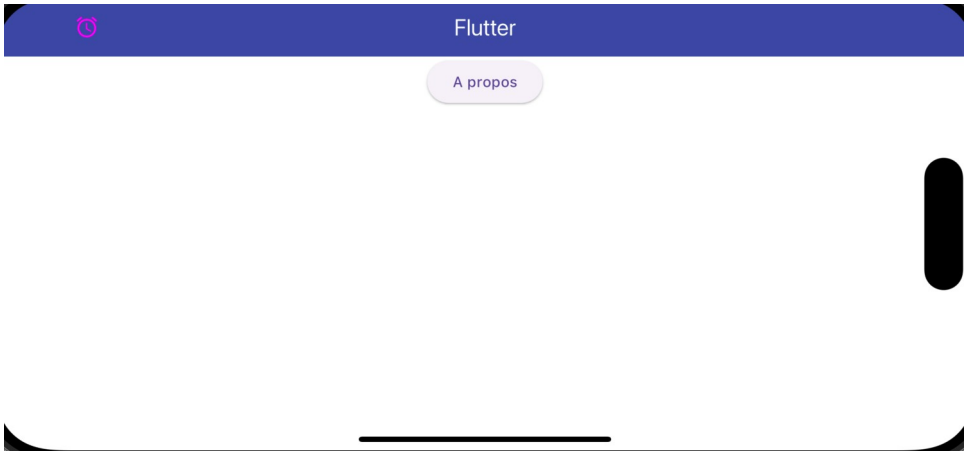
Chapitre 5 – Naviguer sur plusieurs écrans

Table des matières

Création d'un deuxième écran :.....3
Création d'un écran en lui passant un paramètre :.....4



Pour naviguer sur plusieurs écrans, nous allons structurer notre application de la façon suivante :

Dans lib/screens/home_screen.dart mettez le code suivant :

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

class HomeScreen extends StatefulWidget {
  const HomeScreen({super.key});
  @override
  State<StatefulWidget> createState() {
    return _HomeScreen();
  }
}

class _HomeScreen extends State<HomeScreen> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Flutter', style: TextStyle(color: Colors.white)),
        leading: const Icon(Icons.access_alarm, color: Color(0xFFFF00FF)),
```

```
      elevation: 10.0,
      centerTitle: true,
    ),
    backgroundColor: const Color.fromRGBO(255, 255, 255, 1),
  );
 }
}
```

Dans le fichier « main.dart », mettez le code suivant :

```
import 'package:flutter/material.dart';
import 'package:nuage/screens/about_screen.dart';
import 'package:nuage/screens/home_screen.dart';

void main() {
 runApp(const MainApp());
}

class MainApp extends StatelessWidget {
 const MainApp({super.key});

 @override
 Widget build(BuildContext context) {
  return MaterialApp(
    title: 'Flutter',
    theme: ThemeData(
      appBarTheme: const AppBarTheme(
        backgroundColor: Colors.indigo,
      ),
    ),
    debugShowCheckedModeBanner: false,
    initialRoute: '/',
    routes: {
      '/': (context) => const HomeScreen(),
    },
  );
 }
}
```

Commentaire :

initialRoute: '/', //Définit la route initiale de l'application.
 routes: { // Configure un tableau de routes, qui associe chaque chemin de route à un builder de widget.

Travail à faire :

Avant d'apprendre à naviguer sur un nouvel écran, réalisez un bouton « À propos » sur la page d'accueil (HomeScreen).

Création d'un deuxième écran :

Dans « lib/screens/about_screen.dart » :


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

class About extends StatelessWidget {
  const About({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
          title: const Text('À propos', style: TextStyle(color: Colors.white)),
        ),
        body: const Center(
          child: Text("À propos"),
        ));
  }
}
```

Compléter le fichier « main.dart » :

```
initialRoute: '/',
      routes: {
        '/': (context) => const HomeScreen(),
        '/about': (context) => const AboutScreen(),
```

Le bouton « À propos » de la page d'accueil doit maintenant pouvoir se rendre sur la page « À propos » avec le code suivant :

```
void btAbout() {
   Navigator.pushNamed(
     context,
     '/about',
   );
  }
```

Travail à faire :

1) Sur la page « À propos », ajoutez les informations suivantes :
- une image
- le nom de l'application
- la version du logiciel
- « Développée par »
- votre nom

2) Réalisez une page supplémentaire nommée « settings » et faites en sorte de pouvoir s'y rendre à partie de l'écran principal.

Création d'un écran en lui passant un paramètre :

Créer un nouvel écran « message » avec un bouton sur « HomeScreen » puis ajouter le code suivant : (n'oubliez pas de créer sa route)

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

class MessageScreen extends StatelessWidget {
  const MessageScreen({super.key});

  @override
  Widget build(BuildContext context) {
    final args = ModalRoute.of(context)?.settings.arguments as String?;
    final message = args ?? 'No message provided';

    return Scaffold(
        appBar: AppBar(
          title: const Text('Message', style: TextStyle(color: Colors.white)),
        ),
        body: Center(
          child: Text(message),
        ));
  }
}
```

Commentaire :
?.settings.arguments : settings contient la configuration de la route, y compris les arguments passés à celle-ci lors de sa création.

?. est utilisé pour éviter une exception en cas où ModalRoute.of(context) retourne null.

arguments peut contenir n'importe quelles données nécessaires passées à la route.
as String? : Cela tente de « caster » les arguments récupérés en une chaîne de caractères, tout en permettant que la valeur soit null si les arguments ne sont pas du type attendu ou s'ils ne sont pas présents.

Puis dans HomeScreen :

```
void btMessage() {
    Navigator.pushNamed(
      context,
      '/message',
      arguments: 'Message passé en paramètre',
    );
  }
```

Commentaire :
C’est ici que le paramètre est passé à l’écran « Message ».