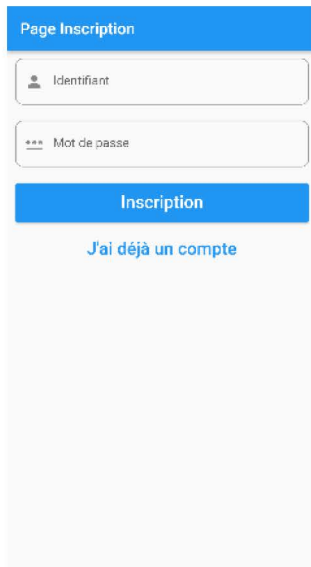


Atelier 3

Enseignant : S. Hadhri

GLID 2

L'objectif de cet atelier est de créer un système d'authentification de notre application qui est formé essentiellement par 3 pages :



Page Inscription

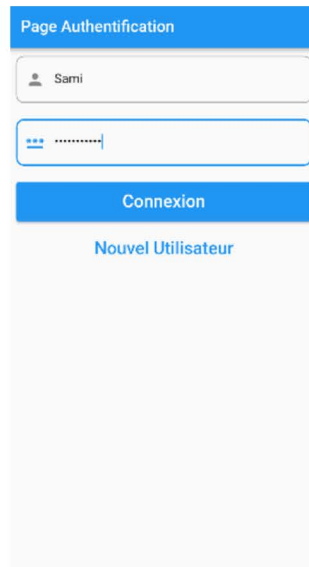
Identifiant

*** Mot de passe

Inscription

J'ai déjà un compte

Page Inscription



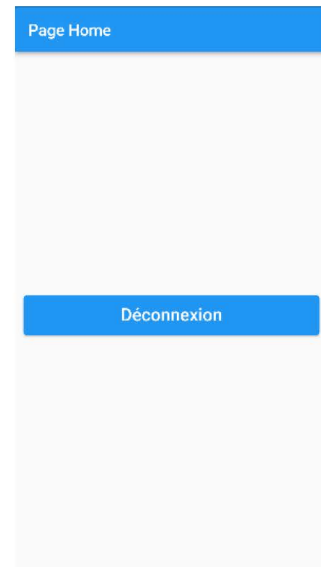
Page Authentification

Sami

Connexion

Nouvel Utilisateur

Page Authentification



Page Home

Déconnexion

Page Home (Accueil)

1. Créer un nouveau projet Flutter nommé « voyage »
2. Sous le dossier « lib », créer un sous-dossier « pages » qui contient 2 fichiers « inscription.page.dart » et « home.page.dart ».
3. Ecrire le code de ces 2 fichiers permettant d'afficher le nom de chaque page au centre. Ecrire également un code basique dans « main.dart » permettant d'afficher la page « inscription.page.dart »

Fichier inscription.page.dart

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

class InscriptionPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('Page Inscription')),
      body: Center(
        child: Text(
          'Page Inscription',
          style: TextStyle(fontSize: 22),
        ),
      ),
    );
  }
}
```

Fichier home.page.dart

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

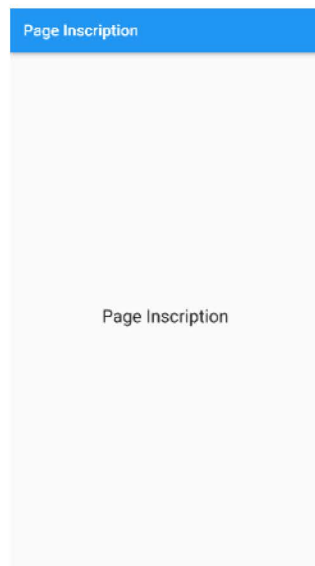
class HomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('Page Home')),
      body: Center(
        child: Text(
          'Page Home',
          style: TextStyle(fontSize: 22),
        )),
    );
  }
}
```

Fichier main.dart

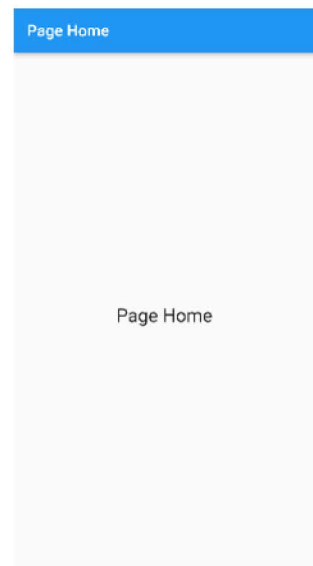
```
import 'package:flutter/material.dart';
import 'package:voyage/pages/inscription.page.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: InscriptionPage(),
    );
  }
}
```



Page Inscription

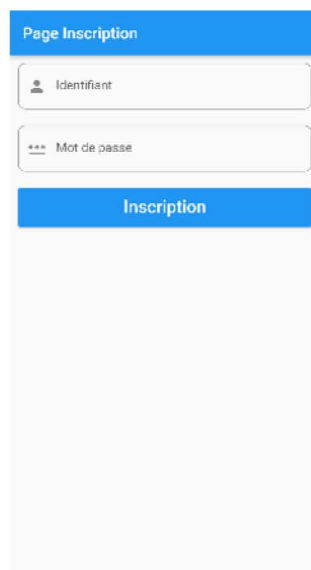


Page Home (Accueil)

4. Créer dans la page d'inscription un formulaire formé par 2 zones de texte (widget `TextFormField`) et un bouton (widget `ElevatedButton`) organisés dans une colonne.

Fichier inscription.page.dart

```
...
TextEditingController txt_login = new TextEditingController();
...
Container(
  padding: EdgeInsets.all(10),
  child: TextFormField(
    controller: txt_login,
    decoration: InputDecoration(
      prefixIcon: Icon(Icons.person),
      hintText: "Utilisateur",
      border: OutlineInputBorder(
        borderSide: BorderSide(width: 1),
        borderRadius: BorderRadius.circular(10)),
    ),
  ),
),
...
Container(
  padding: EdgeInsets.all(10),
  child: ElevatedButton(
    style: ElevatedButton.styleFrom(
      backgroundColor: Colors.blue,
      minimumSize: Size.fromHeight(50)),
    onPressed: () {},
    child: Text('Inscription', style: TextStyle(fontSize: 22))),
  ),
),
...
```



5. Ajouter un système de routage dans le fichier « *main.dart* » selon le schéma suivant :

- **'/home'** : path vers la page Home
- **'/inscription'** : path vers la page Inscription

Fichier main.dart

```
...
class MyApp extends StatelessWidget {
  final routes = {
    '/home': (context) => HomePage(),
    '/inscription': (context) => InscriptionPage(),
  };

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
```

```

        debugShowCheckedModeBanner: false,
        routes: routes,
        home: InscriptionPage(),
    );
}
}

```

6. On souhaite gérer les utilisateurs à travers les préférences partagées. Développer la méthode privée `onInscrire()` appelée au clic sur le bouton « Inscription » qui permet d'ajouter l'identifiant et le mot de passe dans les préférences partagées puis de basculer vers la page Home.

Remarque :

Pour pouvoir utiliser les préférences partagées, il faudrait déclarer la dépendance correspondante dans la partie « dependencies » du le fichier « `pubspec.yaml` » :

Fichier `inscription.page.dart`

```

...
class InscriptionPage extends StatelessWidget {
  late SharedPreferences prefs;
  ...
  onPressed: () {
    _onInscrire(context);
  }
  ...
  Future<void> _onInscrire(BuildContext context) async {
    prefs = await SharedPreferences.getInstance();
    if (!txt_login.text.isEmpty && !txt_password.text.isEmpty) {
      prefs.setString("login", txt_login.text);
      prefs.setString("password", txt_password.text);
      Navigator.pop(context);
      Navigator.pushNamed(context, '/home');
    } else {
      const snackBar = SnackBar(
        content: Text('Id ou mot de passe vides'),
      );
      ScaffoldMessenger.of(context).showSnackBar(snackBar);
    }
  }
}
...

```

Remarque :

`Future` est une classe Dart de base pour travailler avec des opérations asynchrones. Un objet `Futur` représente une valeur ou une erreur potentielle qui sera disponible à un moment donné dans le futur.

7. On souhaite maintenant gérer l'état de connexion de l'utilisateur de telle sorte qu'elle démarre automatiquement sur la page Home si l'utilisateur était connecté lors de la dernière utilisation de l'application. Modifier de nouveau la méthode `_onInscrire()` de façon à enregistrer une variable booléenne « connecte » (qui représente si l'utilisateur est connecté ou non) à `True` dans les préférences partagées.

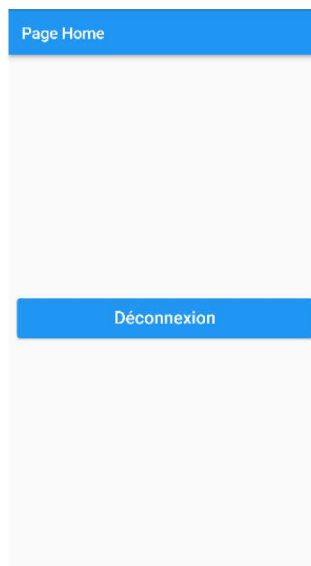
Fichier inscription.page.dart

```
Future<void> _onInscrire(BuildContext context) async {  
  ...  
  prefs.setBool("connecte", true);  
  ...  
}
```

8. Ajouter dans la page « Home » un bouton de déconnexion qui permet d'enregistrer la variable booléenne « connecte » à `False` et de basculer vers la page d'inscription.

Fichier home.page.dart

```
...  
class HomePage extends StatelessWidget {  
  late SharedPreferences prefs;  
  ...  
  body: Center(  
    child: Container(  
      padding: EdgeInsets.all(10),  
      child: ElevatedButton(  
        style: ElevatedButton.styleFrom(  
          backgroundColor: Colors.blue,  
          minimumSize: const Size.fromHeight(50),  
          onPressed: () {  
            _Deconnexion(context);  
          },  
          child: Text('Déconnexion', style: TextStyle(fontSize: 22))),  
      ),  
    ),  
  ),  
  Future<void> _Deconnexion(context) async {  
    prefs = await SharedPreferences.getInstance();  
    prefs.setBool("connecte", false);  
    Navigator.pushNamedAndRemoveUntil(context, '/inscription', (route) => false);  
  },  
}
```



9. Modifier la page « *main.dart* » pour ne pas afficher la page Inscription à chaque fois mais plutôt récupérer la variable booléenne « connecte » des préférences partagées et d'afficher :
- La page « Home » si elle vaut True
 - La page « Authentification » si elle vaut False

Fichier main.dart

```
...
home: FutureBuilder(
  future: SharedPreferences.getInstance(),
  builder: (context, snapshot)
  {
    if (snapshot.hasData) {
      bool conn = snapshot.data?.getBool('connecte') ?? false;
      if (conn)
        return HomePage();
    }
    return InscriptionPage();
  })
...
```

10. Faire toutes les étapes nécessaires pour permettre à un utilisateur inscrit de s'authentifier:
- Sous le dossier « *lib/pages* », créer un fichier « *authentification.page.dart* ».
 - Créer dans la page d'authentification un formulaire formé par 2 zones de texte et un bouton.
 - Ajouter la route de la page dans le système de routage
 - Ajouter sous les boutons « Inscription » et « Connexion » des liens (widget `TextButton`) qui permettent de basculer d'une page à l'autre.



- Développer la méthode privée `_onAuthentifier()` appelée au clic sur le bouton « Connexion » qui permet de vérifier que l'identifiant et le mot de passe saisis sont identiques à ceux enregistrés dans les préférences partagées puis de basculer vers la page Home.

