Persistance locale de données

Possiblités

- SharedPreferences (clés – valeurs)
- Fichiers
- Sqlite (Supporte Ios, Android, MacOS)
- Hive
- Drift (Base de données réactive basée sur Sqlite)

Persistance avec Drift (https://drift.simonbinder.eu/setup/#database-class)

1. Ajouter les dépendances

Dépendances nécessaires : drift (installe aussi : sqlite3_flutter_libs, path_provider, path)

```
PS C:\ateliers\Flutter\flutter_app_2024> dart pub add drift
Resolving dependencies...
Downloading packages... (4.7s)
  async 2.11.0 (2.12.0 available)
  boolean_selector 2.1.1 (2.1.2 available)
  characters 1.3.0 (1.3.1 available)
  clock 1.1.1 (1.1.2 available)
  collection 1.18.0 (1.19.1 available)
+ convert 3.1.2
+ drift 2.21.0
  fake_async 1.3.1 (1.3.2 available)
  flutter_lints 4.0.0 (5.0.0 available)
  http_parser 4.0.2 (4.1.1 available)
  leak_tracker 10.0.5 (10.0.8 available)
  leak_tracker_flutter_testing 3.0.5 (3.0.9 available)
  lints 4.0.0 (5.1.0 available)
  matcher 0.12.16+1 (0.12.17 available)
  material_color_utilities 0.11.1 (0.12.0 available)
  meta 1.15.0 (1.16.0 available)
  path 1.9.0 (1.9.1 available)
+ sqlite3 2.4.7
  stack_trace 1.11.1 (1.12.0 available)
  string_scanner 1.2.0 (1.4.0 available)
  test_api 0.7.2 (0.7.3 available)
  vm_service 14.2.5 (14.3.1 available)
Changed 3 dependencies!
19 packages have newer versions incompatible with dependency constraints.
Try `dart pub outdated` for more information.
```

```
dependencies:
  drift: ^2.21.0
  flutter:
    sdk: flutter        You, 2 weeks a
  flutter_slidable: ^3.1.1
  image_picker: ^1.1.2
  path: ^1.9.0
  path_provider: ^2.1.5
  sqlite3_flutter_libs: ^0.5.26
```

2. Ajouter les dépendances de développement

Dépendances dév : drift_dev et build_runner (Pour générer le code des requêtes)

```
C:\ateliers\Flutter\flutter_app_2024>dart pub add dev:drift_dev dev:build_runner
Resolving dependencies...
Downloading packages... (3.3s)
+ _fe_analyzer_shared 72.0.0 (76.0.0 available)
+ _macros 0.3.2 from sdk dart
+ analyzer 6.7.0 (6.11.0 available)
+ analyzer_plugin 0.11.3
+ args 2.6.0
```

```
dev_dependencies:
  flutter_test:
    sdk: flutter
  flutter_lints: ^4.0.0
  drift_dev: ^2.21.2
  build_runner: ^2.4.13
```

3. Créer un fichier base.dart (dossier : data) contenant les classes suivantes :

```dart
// La table produits
import 'package:drift/drift.dart';
part 'base.g.dart';    Target of URI hasn't been generated: 'package

@DataClassName("Produit")
class Produits extends Table {
  IntColumn get id => integer().autoIncrement()();
  TextColumn get libelle => text().withLength(min: 2, max: 120)();
  TextColumn get description => text()();
  TextColumn get photo => text()();
}

@DriftDatabase(tables: [Produits])
class ProduitsDatabase extends _$ProduitsDatabase {}
```

4. Génération du code de la base de données

Exécuter “dart run build_runner build” à partir de la racine du projet en ligne de commande

```
C:\ateliers\Flutter\flutter_app_2024>dart run build_runner build
Building package executable...
Built build_runner:build_runner.
[INFO] Generating build script completed, took 301ms
[INFO] Reading cached asset graph completed, took 370ms
[INFO] Checking for updates since last build completed, took 9.6s
[INFO] Running build completed, took 17.5s
[INFO] Caching finalized dependency graph completed, took 563ms
[INFO] Succeeded after 18.2s with 6 outputs (8 actions)
```

le fichier base.g.dart est généré dans le dossier data

```
C:\ateliers\Flutter\flutter_app_2024>dart run build_runner watch
Building package executable...
Built build_runner:build_runner.
[INFO] Generating build script completed, took 317ms
[INFO] Setting up file watchers completed, took 11ms
[INFO] Waiting for all file watchers to be ready completed, took 114ms
[INFO] Reading cached asset graph completed, took 107ms
[INFO] Checking for updates since last build completed, took 7.3s
[INFO] Running build completed, took 117ms
[INFO] Caching finalized dependency graph completed, took 494ms
[INFO] Succeeded after 689ms with 0 outputs (0 actions)
```

5. Compléter la classe ProduitsDatabase comme suit :

```dart
@DriftDatabase(tables: [Produits])
class ProduitsDatabase extends _$ProduitsDatabase {
  ProduitsDatabase() : super(_openConnection());

  @override
  int get schemaVersion => 1;
}

LazyDatabase _openConnection() {
  return LazyDatabase(() async {
    final dbFolder = await getApplicationDocumentsDirectory();
    final file = File(p.join(dbFolder.path, 'produits.db'));
    return NativeDatabase.createInBackground(file);
  }); // LazyDatabase
}
```

Les imports suivants seront nécessaires:

```dart
import 'dart:io';
import 'package:path_provider/path_provider.dart';
import 'package:drift/native.dart';
import 'package:path/path.dart' as p;
```

6. Ajouter ProduitDAO (dossier dao, fichier :produit_dao.dart)

```dart
import 'package:drift/drift.dart';
import 'package:flutter_app_2024/data/base.dart';

@DriftAccessor(tables: [Produits])
class ProduitDAO extends DatabaseAccessor<ProduitsDatabase> {
  ProduitDAO(super.attachedDatabase);

  /// Insère un nouveau produit dans la base de données.
  Future<void> insertProduit(Produit produit) =>
      into(attachedDatabase.produits).insert(produit);

  /// Récupère tous les produits de la base de données.
  Future<List<Produit>> getAllProduits() =>
      select(attachedDatabase.produits).get();

  /// Fournit un flux de tous les produits de la base de données.
  Stream<List<Produit>> getProduitsStream() =>
      select(attachedDatabase.produits).watch();

  /// Met à jour un produit existant dans la base de données.
  Future<bool> updateProduit(Produit produit) =>
      update(attachedDatabase.produits).replace(produit);

  /// Supprime un produit de la base de données par son identifiant.
  Future<int> deleteProduitBuId(int id) =>
      (delete(attachedDatabase.produits)..where((p) => p.id.equals(id))).go();

  /// Récupère un produit de la base de données par son identifiant.
  Future<Produit> getProduitById(int id) =>
      (select(attachedDatabase.produits)..where((p) => p.id.equals(id)))
          .getSingle();
}
```

7. Initialiser la base données et ProduitDAO dans la fonction main()

```dart
void main() {
  final database = ProduitsDatabase();
  final produitDAO = ProduitDAO(database);
  runApp(MainApp(
    produitDAO: produitDAO,
  ));
}
```

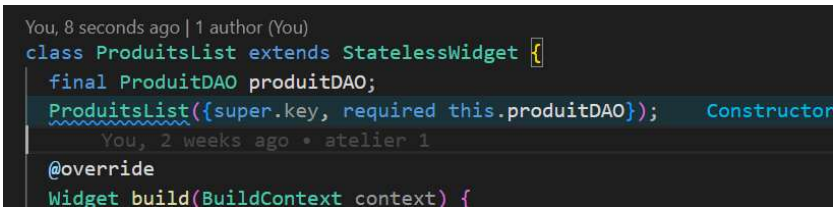
8. produitDAO doit être passé au Constructeur de ProduitList

```dart
home:    ProduitsList(produitDAO: produitDAO));
```

9. Mise à jour de ProduitList

a. Transformer le widget en statelessWidget, (il faut enlever tous les appels à setState pour que l'assistant vs code vous propose l'option de conversion vers un widget statelessWidget)

b. Supprimer tous les membres de la classe (méthodes et attributs) et ajouter un membre produitDAO



c. Encapsuler ListView par StreamBuilder<List<ProduitsTable>>

10. Ajout d'un produit (addProduitForm)

```dart
import 'dart:io';
import 'package:flutter/material.dart';
import 'package:flutter_app_2024/dao/produit_dao.dart';
import 'package:flutter_app_2024/data/base.dart';
import 'package:image_picker/image_picker.dart';

You, 21 hours ago | 1 author (You)
class AddProduitForm extends StatefulWidget {
  final ProduitDAO produitDAO;
  const AddProduitForm({super.key, required this.produitDAO});

  @override
  State<AddProduitForm> createState() => _AddProduitFormState();
}

You, 21 hours ago | 1 author (You)
class _AddProduitFormState extends State<AddProduitForm> {
  final _formKey = GlobalKey<FormState>();
  double? _prix;
  String? _libelle;
  String? _description;

  String? _pickedImagePath;
```

```dart
  Future<void> _pickImage() async {
    final ImagePicker picker = ImagePicker();
    final XFile? pickedFile =
        await picker.pickImage(source: ImageSource.gallery);
    if (pickedFile == null) return;
    setState(() {
      _pickedImagePath = pickedFile.path;
    });
  }

  void _saveProduit() {
    if (_formKey.currentState!.validate()) {
      _formKey.currentState!.save();
      final photo = _pickedImagePath;

      widget.produitDAO.insertProduit(ProduitsCompanion.insert(
          libelle: _libelle!,
          description: _description!,
          prix: _prix!,
          photo: photo!));

      Navigator.pop(context);
    }
  }
```

```dart
@override
Widget build(BuildContext context) {
  return Scaffold(
    resizeToAvoidBottomInset: false,
    appBar: AppBar(title: const Text('Ajouter un produit')),
    body: Padding(
      padding: const EdgeInsets.all(12.0),       You, 2 days ago • Fusion de la
      child: SingleChildScrollView(
        child: Form(
          key: _formKey,
          child: Column(
            children: [
              TextFormField(
                decoration: const InputDecoration(labelText: 'Libellé:'),
                validator: (value) {
                  if (value!.isEmpty) {
                    return 'Veuillez saisir le libellé';
                  }
                  return null;
                },
                onSaved: (value) => _libelle = value,
              ), // TextFormField
              TextFormField(
                decoration: const InputDecoration(labelText: 'Description'),
                maxLines: null,
                keyboardType: TextInputType.multiline,
                validator: (value) {
                  if (value == null || value.isEmpty) {
                    return 'Veuillez saisir la description';
                  }
                  return null;
                },
                onSaved: (value) => _description = value,
              ), // TextFormField
```

```dart
TextFormField(
  decoration: const InputDecoration(
    labelText: 'Email',
  ), // InputDecoration
  validator: (value) {
    if (value == null || value.isEmpty) {
      return 'Entrez l' 'email';
    }
    return null;
  },
  onSaved: (newValue) => email = newValue!,
), // TextFormField
TextFormField(
  decoration: const InputDecoration(
    labelText: 'Age',
  ), // InputDecoration
  keyboardType: TextInputType.number,
  validator: (value) {
    if (value == null || value.isEmpty || int.parse(value) < 17)
      return 'L' 'age doit être supérieur ou égal à 17 ans';
    }
    return null;
  },
  onSaved: (newValue) => age = int.parse(newValue!),
), // TextFormField
```

```dart
TextFormField(
  decoration: const InputDecoration(labelText: 'Prix'),
  keyboardType: TextInputType.number,
  validator: (value) {
    if (value != null && value.isEmpty) {
      return 'Veuillez saisir le prix';
    }
    return null;
  },
  onSaved: (value) => _prix = double.parse(value!),
), // TextFormField
```

```dart
GestureDetector(
    onTap: _pickImage,
    child: Container(
      width: 80,
      height: 80,
      decoration: BoxDecoration(
        border:
            Border.all(color: Theme.of(context).primaryColor),
        color: Theme.of(context).primaryColorLight,
        shape: BoxShape.circle,
        image: DecorationImage(
          fit: BoxFit.cover,
          image: _pickedImagePath != null
              ? FileImage(File(_pickedImagePath!))
                  as ImageProvider<Object>
              : const AssetImage(
                  'assets/images/placeholder.jpg'), // AssetImage
        ), // DecorationImage
      ), // BoxDecoration
    )), // Container // GestureDetector
```

```dart
              )); // Container // GestureDetector
            Padding(
              padding: const EdgeInsets.symmetric(vertical: 16.0),
              child: Row(
                children: [
                  TextButton.icon(
                      onPressed: () {
                        _formKey.currentState!.reset();
                        setState(() {
                          _pickedImagePath = null;
                        });
                      },
                      icon: const Icon(Icons.restore_from_trash_rounded),
                      label: const Text('Réinitialiser')), // TextButton.icon
                  ElevatedButton(
                    onPressed: () {
                      if (_formKey.currentState!.validate()) {
                        _saveProduit();
                      }
                    },
                    child: const Text('Enregistrer'),
                  ), // ElevatedButton
                ],
              ), // Row
            ), // Padding
          ],
        ), // Column
      ), // Form
    ), // SingleChildScrollView
  ), // Padding
); // Scaffold
}
```

11. ProduitDetails

```dart
import 'dart:io';
import 'package:flutter/material.dart';
import 'package:flutter_app_2024/data/base.dart';

You, 21 hours ago | 1 author (You)
class ProduitDetails extends StatelessWidget {
  final Produit produit;
  const ProduitDetails({super.key, required this.produit});
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(produit.libelle,
            style: const TextStyle(fontSize: 16, fontWeight: FontWeight.bold)), //
      ), // AppBar
      body: SingleChildScrollView(      You, 21 hours ago • Uncommitted changes
        child: Center(
          child: Padding(
            padding: const EdgeInsets.all(16.0),
            child: Column(
              crossAxisAlignment: CrossAxisAlignment.center,
              children: [
                Image.file(
                  File(produit.photo),
                  height: 160,
                  width: 160,
                ), // Image.file
```

```dart
                const SizedBox(height: 16),
                Text('Prix: ${produit.prix}',
                    style: const TextStyle(fontSize: 16)), // Text
                const SizedBox(height: 16),
                Text(
                  'Description: ${produit.description}',
                  style: const TextStyle(fontSize: 16),
                  maxLines: null,
                  overflow: TextOverflow.visible,
                ), // Text
              ],
            ), // Column
          ), // Padding
        ), // Center
      ), // SingleChildScrollView
    ); // Scaffold
  }
}
```

12. ProduitBox

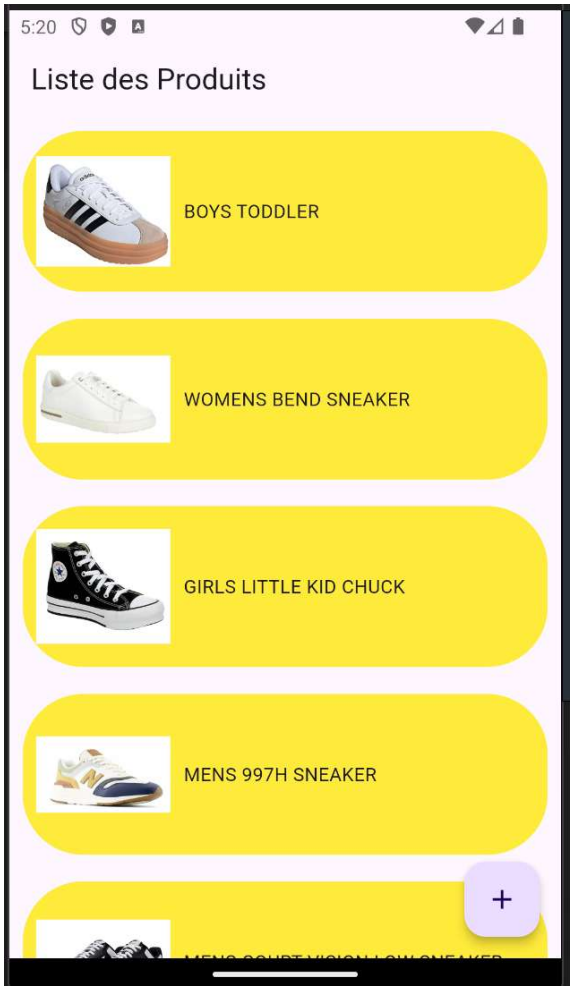
```dart
import 'dart:io';
import 'package:flutter/material.dart';
import 'package:flutter_app_2024/dao/produit_dao.dart';
import 'package:flutter_app_2024/data/base.dart';
import 'package:flutter_app_2024/produit_details.dart';
import 'package:flutter_slidable/flutter_slidable.dart';

You, 21 hours ago | 1 author (You)
class ProduitBox extends StatelessWidget {
  final Produit produit;
  final ProduitDAO produitDAO;

  const ProduitBox({
    super.key,
    required this.produit,
    required this.produitDAO,
  });

  @override
  Widget build(BuildContext context) {
    return Padding(
      padding: const EdgeInsets.all(10.0),
      child: Slidable(
        endActionPane: ActionPane(
          motion: const StretchMotion(),
          children: [
            SlidableAction(
              onPressed: (context) => produitDAO.deleteProduitBuId(produit.id),
              icon: Icons.delete,
              backgroundColor: Colors.red,
              borderRadius: BorderRadius.circular(45),
            ), // SlidableAction
          ],
        ), // ActionPane
```

```
        child: InkWell(
          onTap: () {
            Navigator.push(
                context,
                MaterialPageRoute(
                    builder: (context) => ProduitDetails(produit: produit))); // MaterialPageRoute
          },
          child: Container(
            decoration: BoxDecoration(
              color: Colors.yellow,       You, 2 days ago • Fusion de la branche atelier2 dans main
              borderRadius: BorderRadius.circular(45),
            ), // BoxDecoration
            height: 120,
            child: Row(
              children: [
                const SizedBox(width: 10),
                Center(
                  child:
                      Image.file(width: 100, height: 100, File(produit.photo)),
                ), // Center
                const SizedBox(width: 10),
                Text(produit.libelle),
              ],
            ), // Row
          ), // Container
        ), // InkWell
      ), // Slidable
    ); // Padding
  }
}
```


5:20
Liste des Produits
BOYS TODDLER
WOMENS BEND SNEAKER
GIRLS LITTLE KID CHUCK
MENS 997H SNEAKER
+