Hive

Lokalna podatkovna baza
- Flutter

Hive

- Hive je lahka in izjemno hitra **NoSql** podatkovna baza tipa **ključ-vrednost**, napisana v čistem **Dartu**.
- Močno šifrirana z uporabo **AES-256**.
- Hive ne podpira le osnovne tipe in sezname, ampak tudi **kateri koli Dart objekt**.
- Potrebno je ustvariti **adapter** za tip pred shranjevanjem objektov ❗

https://docs.hivedb.dev/#/

Primer uporabe podatkovne baze Hive

1. Dodajanje odvisnosti
2. Definiranje objekta (Hive)
3. Inicializacija baze
4. Pridobivanje iz baze

1. Dodajanje odvisnosti

- V datoteko ***pubspec.yaml*** dodamo ustrezne odvisnosti

```
dependencies:
  hive: ^2.2.3 # check for the latest version
  hive_flutter: ^1.1.0 # check for the latest version

dev_dependencies:
  hive_generator: ^2.0.1 # check for the latest version
  build_runner: ^2.4.6 # check for the latest version
```

2. Definiranje objekta (Hive)

- Obstoječi model spremenimo v **objekt Hive**

- V terminalu poženemo ukaz:

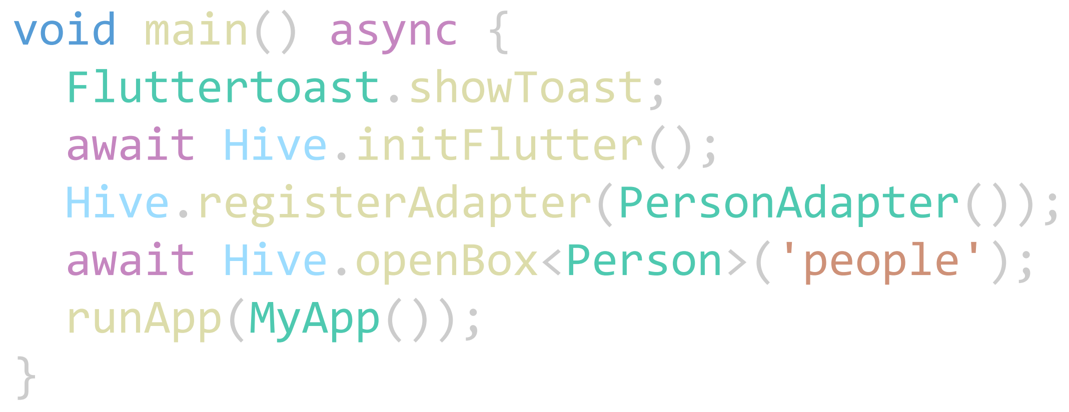
  dart *run* build_runner **build**

  ki ustvari nov **adapter**

```dart
import 'package:hive/hive.dart';

part 'person.g.dart'; // add this line

@HiveType(typeId: 0)
class Person {
  @HiveField(0)
  final String name;
  @HiveField(1)
  final String surname;
  @HiveField(2)
  final String position;
  @HiveField(3)
  final DateTime checkInDateTime;

  Person({
    required this.name,
    required this.surname,
    required this.position,
    required this.checkInDateTime,
  });
}
```

3. Inicializacija baze

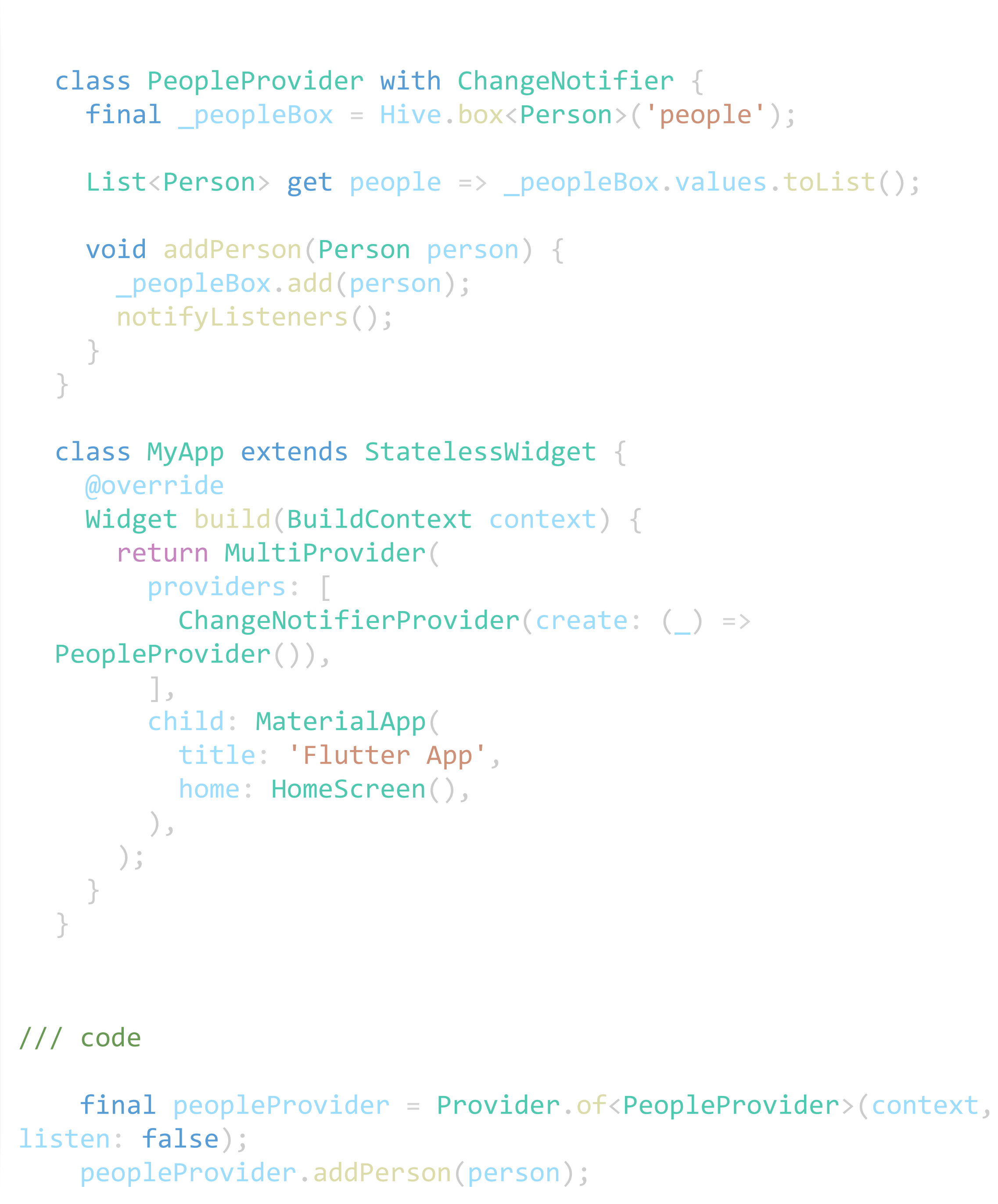
- Znotraj main metode **inicializairamo** podatkovno bazo Hive, registriramo ustvarjen adapter in uporabimo metodo ***.openBox()***



```
void main() async {
  Fluttertoast.showToast;
  await Hive.initFlutter();
  Hive.registerAdapter(PersonAdapter());
  await Hive.openBox<Person>('people');
  runApp(MyApp());
}
```

4. Pridobivanje iz baze

- Uredimo obstoječo kodo, da beremo (pišemo) podatke iz (v) baze/o



```
  class PeopleProvider with ChangeNotifier {
    final _peopleBox = Hive.box<Person>('people');

    List<Person> get people => _peopleBox.values.toList();

    void addPerson(Person person) {
      _peopleBox.add(person);
      notifyListeners();
    }
  }

  class MyApp extends StatelessWidget {
    @override
    Widget build(BuildContext context) {
      return MultiProvider(
        providers: [
          ChangeNotifierProvider(create: (_) =>
  PeopleProvider()),
        ],
        child: MaterialApp(
          title: 'Flutter App',
          home: HomeScreen(),
        ),
      );
    }
  }


/// code

    final peopleProvider = Provider.of<PeopleProvider>(context,
listen: false);
    peopleProvider.addPerson(person);
```