Dependency:

hive: ^1.4.4

hive\_flutter: ^0.3.1

Dev\_dependency:

hive\_generator: ^0.7.1

build\_runner:

Step 1). In main.dart

WidgetsFlutterBinding.*ensureInitialized*();

Directory directory = await getApplicationDocumentsDirectory();

Hive.init(directory.path);

//await Hive.close(); //close the hive

//await Hive.deleteFromDisk(); //remove all boxes

//await Hive.deleteBoxFromDisk('followed'); //remove specific box

Hive.registerAdapter(FollowedAdapter());

await Hive.openBox<Follow>('followed');

Step 2)

import 'package:hive/hive.dart';

part 'follow.g.dart';

@HiveType(typeid:1)

class Follow extends HiveObject {

@HiveField(0)

int riderId;

Follow(this.riderId);

}

Step 3)

flutter packages pub run build\_runner build

If conflict error then below line

flutter packages pub run build\_runner build --delete-conflicting-outputs

Step 4)

var box = Hive.box<Follow>('followed');

box.clear();

box.add(Follow(10));

box.add(Follow(10));

//await box.deleteAt(0);

for (int i = 0; i < box.values.length; i++)

print("${box.getAt(i).riderId}");

Step. Dummy) Practical use

if (box.values.contains(Follow(e.id))) { // if you now index use. box.add instead of put and use box.deleteAt() instead of delete.

print("already followed ${e.id}");

await box.delete(e.id);

setState(() {});

} else {

box.put(e.id, Follow(e.id));

for (int i = 0; i < box.values.length; i++)

print("${box.getAt(i).riderId}");

setState(() {});

}