**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA ĐIỆN TỬ VIỄN THÔNG**

**BÁO CÁO**

**LẬP TRÌNH ĐA NỀN TẢNG**

SharedPreferences trong Flutter – Lưu trữ dữ liệu đơn giản

**Sinh viên thực hiện:**

**01. Nguyễn Hồng Ân** **Lớp: 22KTMT1   MSSV106220208**

**02. Phan Văn Danh** **Lớp: 22KTMT1   MSSV:106220212**

**Người hướng dẫn:**

**TS. Nguyễn Duy Nhật Viễn**

**Đà Nẵng, ngày 26 tháng 10 năm 2025.**

**THUYẾT MINH**

**BÁO CÁO**

**LẬP TRÌNH ĐA NỀN TẢNG**

SharedPreferences trong Flutter – Lưu trữ dữ liệu đơn giản

BẢNG PHÂN CÔNG CÔNG VIỆC TRONG NHÓM

|  |  |  |  |
| --- | --- | --- | --- |
| STT | HỌ VÀ TÊN | NHIỆM VỤ | KHỐI LƯỢNG |
| 01 | NGUYỄN HỒNG ÂN |  | 50% |
| 02 | PHAN VĂN DANH |  | 50% |

Github:  <https://github.com/nguyenhongan15/demo_app_with_sharedpreferences>

**Mục lục**

[1.Dữ liệu cục bộ trong Flutter 4](#_Toc212486547)

[2.SharedPreferences 4](#_Toc212486548)

[3.Demo app sử dụng SharedPreferences 7](#_Toc212486549)

[4.Các phương thức phương lưu trữ phổ biến khác trong Flutter 21](#_Toc212486550)

[4.1.SQLite 21](#_Toc212486551)

[4.2.File Storage – Lưu dữ liệu trong file 22](#_Toc212486552)

[4.3.Cloud Storage / Firebase 22](#_Toc212486553)

[4.4.Bảng so sánh các phương thức lưu trử chính trong FLUTTER 23](#_Toc212486554)

[5.Xử lý lỗi và Best practices 24](#_Toc212486555)

[5.1.Các lỗi thường gặp khi sử dụng SharedPreferences 24](#_Toc212486556)

[5.2.Best practices 26](#_Toc212486557)

[6.Tổng kết 27](#_Toc212486558)

[7.Tài liệu tham khảo 28](#_Toc212486559)

**1.Dữ liệu cục bộ trong Flutter**

    Dữ liệu cục bộ (LocalData) là dữ liệu được lưu trữ trực tiếp trên thiết bị người dùng. Nói cách khác dữ liệu cục bộ là dữ liệu nằm trong bộ nhớ của chính điện thoại hoặc máy tính đang chạy app.

Ví dụ:

* Tài khoản người dùng đã đăng nhập.
* Cài đặt ứng dụng (chế độ tối/sáng).
* Danh sách yêu thích (favorite).
* Dữ liệu cache (tạm thời).

Mục đích:

* Giúp ứng dụng hoạt động ngoại tuyến (offline).
* Tăng tốc độ truy xuất dữ liệu.
* Giảm tần suất gọi API.

**2. SharedPreferences**

SharedPreferences là một trong những cách lưu trữ cơ bản nhất để lưu trữ dữ liệu cục bộ, hoạt động như một file cấu hình hoặc registry lưu dưới dạng key-value.

Cho phép lưu các giá trị nhỏ như:

* Trạng thái đăng nhập
* Chủ đề sáng/tối
* Tên người dùng, ngôn ngữ, kích thước chữ,..
* Các cài đặt cá nhân đơn giản khác

Cách sử dụng:

* Loại dữ liệu hỗ trợ: int, double, bool, String và List<String>
* Hỗ trợ các nền tảng: Android, iOS, Linux, macOS, Web, Windows
* Sử dụng SharedPreferences (truyền thống)
* Ví dụ ghi dữ liệu

A screen shot of a computer code

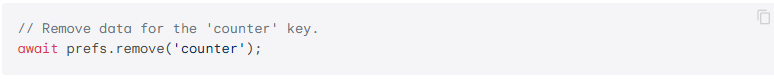
AI-generated content may be incorrect.

* Ví dụ đọc dữ liệu

A screen shot of a computer code

AI-generated content may be incorrect.

* Xoá mục



* Sử dụng SharedPreferencesAsync

A screen shot of a computer code

AI-generated content may be incorrect.

* Sử dụng SharedPreferencesWithCache

A screen shot of a computer code

AI-generated content may be incorrect.

Nguyên lý hoạt động:

* Trên Android:

Flutter gọi xuống API gốc của android cũng tên là SharedPreferences.

Android sẽ lưu dữ liệu trong 1 file XML nằm trong thư mục:

/data/data/<package\_name>/shared\_prefs/

File XML này chứa các căp key-value, ví dụ:

<map>

<string name="username">anngh</string>

<boolean name="isDarkMode">true</boolean>

<int name="fontSize">16</int>

</map>

Khi gọi hàm set, Flutter gọi plugin shared\_preferences, plugin chuyển lệnh xuống Android Layer rồi ghi giá trị vào file XML. Khi app khởi động lại, Flutter sẽ đọc từ file XML đó.

* Trên Ios: Flutter sử dụng API gốc của Ios gọi là NSUserDefaults. Dữ liệu được lưu trong plist (property list), dạng tương tự XML. Vị trí file thường nằm trong thư mục app sandbox.
* Trên Web: Flutter sử dụng window.localStorage của trình duyệt, dữ liệu lưu trong bộ nhớ web ở dạng key-value. Nếu người dùng xoá dữ liệu trình duyệt thì sẽ mất dữ liệu chương trình.

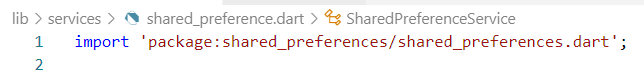
**3. Demo app sử dụng SharedPreferences**

* Thêm thư viện vào pubspec.yaml

A screen shot of a computer

AI-generated content may be incorrect.

* Để sử dụng SharedPreferences thì ta phải import package vào trong file code. Ví dụ:



* Tạo class SharedPreferences để quản lý dữ liệu, đọc, ghi và cập nhật dữ liệu như danh sách người dùng, dark mode, avatar. Lớp này hoạt động khi app chạy, init() sẽ được gọi để mở kết nối đến. Khi người dùng thay đổi cài đặt, các hàm addOrUpdateUsers, setSavedDarkMode,.. được gọi để cập nhật thay đổi. Các dữ liệu sẽ được lưu thành JSON trong bộ nhớ thiết bị nhờ class usersetting. Khi app mở lại sẽ gọi users, savedDarkMode để đọc lại dữ liệu đã lưu.

A screenshot of a computer program

AI-generated content may be incorrect.

A computer screen with text

AI-generated content may be incorrect.

A computer screen shot of a program code

AI-generated content may be incorrect.

A screen shot of a computer code

AI-generated content may be incorrect.

A white background with black text

AI-generated content may be incorrect.

A computer screen shot of a computer code

AI-generated content may be incorrect.

* Class usersetting: là lớp mô tả thông tin cài đặt của 1 người dùng, nó đại diện cho 1 người dùng cụ thể và các cài đặt cá nhân của họ như theme, font chữ, font size, …

A screenshot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer code

AI-generated content may be incorrect.

* Mối quan hệ giữa 2 lớp

A diagram of a service

AI-generated content may be incorrect.

* Trong hàm main() phải gọi await instance.init() một lần khi app chạy

A computer code with text

AI-generated content may be incorrect.

* Bắt đầu từ file get\_started\_page.dart, SharedPreferences được kết nối, lúc này chưa lưu gì mới, chỉ đọc lại dữ liệu cũ.

A computer code with text

AI-generated content may be incorrect.

* Dòng await SharedPreferenceService.instance.init():

-Mở kết nối với bộ nhớ SharedPreferences của thiết bị.

-Đây là bước khởi tạo lần đầu tiên, giống như "mở ổ đĩa" để truy cập dữ liệu đã lưu trước đó.

* Sau đó userList = SharedPreferenceService.instance.users: Đọc danh sách người dùng đã từng lưu (nếu có).

Khi người dùng nhấn vào tạo tài khoản mới , hàm \_addNewUser() được gọi để lưu người dùng mới vào SharedPreferences và nếu danh sách quá 3 thì sẽ xoá bớt và cập nhật lại danh sách đã lưu.

A screenshot of a computer code

AI-generated content may be incorrect.

Sau khi lưu xong sẽ chuyển hướng qua màn chọn Mode sáng/tối

A computer screen shot of a program

AI-generated content may be incorrect.

Hoặc nếu có user trước đo rồi thì sẽ chuyển thẳng sang màn hình của file HelloUserPage.dart, khi chuyển qua trang này tức là nó đã đọc dữ diệu từ người dùng đã cài đặt và lưu trước đó, bây giờ chỉ việc mở lại các cài đặt lúc trước.

A screen shot of a computer code

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a cell phone

AI-generated content may be incorrect.

* Trường hợp người dùng chọn tạo mới, lúc này ta sẽ đến với trang chọn mode sáng/tối, khi chọn Light hoặc Dark rồi bấm Continue, chương trình chỉ tạo ra một object mới của UserSetting có trạng thái isDarkMode. Lúc này giá trị isDarkMode mới chỉ nằm trong biến updatedUser tạm thời mà chưa được ghi vào bộ nhớ SharedPreferences

A screen shot of a computer program

AI-generated content may be incorrect.

Sau đó chuyển qua trang chọn kích cỡ chữ bằng Navigator.push

A close-up of a computer code

AI-generated content may be incorrect.

A screenshot of a cell phone

AI-generated content may be incorrect.

* Đến với màn hình chọn kích cỡ chữ, lúc này cả Dark/Light mode và fontSize đều được lưu vào SharedPreferences thông qua:



Cụ thể, khi kéo thanh trượt để chọn kích cỡ chữ theo mong muốn và nhấn nút “Continue” thì hàm updatedUser() lấy username từ trang đầu tiên, mode chọn từ trang trước và fontsize người dùng chọn từ thanh trượt ở trang hiện tại, ghi dữ liệu mới vào SharedPreferences. Hàm addOrUpdateUsers sẽ làm trung gian giúp cập nhật một user mới mà không làm mất dữ liệu của user khác, nó chèn hoặc cập nhật một user trong danh sách rồi mới gọi saveUsers

A screen shot of a computer program

AI-generated content may be incorrect.

Hàm addOrUpdateUsers (tạo trong class SharedPreferences) sẽ làm trung gian giúp cập nhật một user mới mà không làm mất dữ liệu của user khác, nó chèn hoặc cập nhật một user trong danh sách rồi mới gọi saveUsers(nằm trong class SharedPreferences)

A computer screen shot of a program code

AI-generated content may be incorrect.

A screenshot of a cell phone

AI-generated content may be incorrect.

* Kiểm tra và lưu xong , ta sẽ đến với màn hình chào người dùng được tạo bởi class HelloUserPage. Ở màn hình này ta sẽ biết được người dùng chọn mode sáng hay tối, kích cỡ chữ như thế nào và cũng là nơi xử lý lưu dữ liệu cuối cùng khi người dùng quay lại trang đầu tiên. Và khi quay lại trang khởi đầu, ứng dụng đảm bảo rằng dữ liệu người dùng hiện tại đã được lưu trong SharedPreferences.

A screen shot of a computer code

AI-generated content may be incorrect.

A screenshot of a phone

AI-generated content may be incorrect.

* Khi quay trở lại trang đầu, lúc này sẽ có thêm user mới, nếu muốn có thể chọn vào user đã được tạo trước đó hoặc có thể tạo user khác tuỳ vào nhu cầu người dùng.

A screen shot of a cell phone

AI-generated content may be incorrect.

A screen shot of a cell phone

AI-generated content may be incorrect.

**4. Các phương thức phương lưu trử phổ biến khác trong Flutter**

* 1. **. SQLite**

SQLite là một cơ sở dữ liệu quan hệ nhúng, mạnh mẽ để lưu trữ dữ liệu có cấu trúc dạng bảng.

* Lưu trữ dữ liệu có cấu trúc bằng SQL.
* Thư viện: sqflite + path\_provider.
* Sử dụng thư viện sqflite để tương tác.
* Phù hợp cho danh sách bài viết, sản phẩm, lịch sử giao dịch...
* Yêu cầu viết các câu lệnh SQL thuần túy.
  1. Khởi tạo CSDL

A screenshot of a computer code

AI-generated content may be incorrect.

**Giải thích:**

- openDatabase(...) → Mở hoặc tạo cơ sở dữ liệu mới (file .db) trong bộ nhớ ứng dụng.

- 'app\_database.db' → Tên file database sẽ được lưu cục bộ

(VD: /data/data/app\_name/databases/app\_database.db).

- version: 1 → Phiên bản database

(sẽ dùng khi cần cập nhật cấu trúc bảng sau này).

- onCreate: → Hàm này chỉ chạy **lần đầu tiên** khi database được tạo.

- db.execute('CREATE TABLE posts(...)') → Câu lệnh SQL để **tạo bảng** posts gồm:

* id → kiểu INTEGER PRIMARY KEY (tự tăng, là khóa chính)
* title → kiểu TEXT (chuỗi)
  1. Chèn dữ liệu

A black and white text

AI-generated content may be incorrect.

**Giải thích:**

* db.insert() là hàm tiện lợi của sqflite để thêm dữ liệu.
* 'posts' là tên bảng.
* {'title': 'New Post'} là map dữ liệu tương ứng với cột title.
  1. Truy cập dữ liệu (Select)G

Giải thích:

* db.query('posts') → Lấy toàn bộ dữ liệu trong bảng posts.
* Kết quả trả về là **danh sách các hàng (List)**, mỗi hàng là **Map<String, dynamic>**
  1. **File Storage – Lưu dữ liệu trong file**

**Dùng cho:** file .txt, .json, ảnh, dữ liệu tạm.

**Đặc điểm:**

- Khi muốn tự tạo file trong bộ nhớ (app directory).

- Có thể đọc/ghi như file bình thường trong máy tính.

**Dùng khi:**

* Lưu file ảnh, JSON, tài liệu.
* Tự quản lý dữ liệu → linh hoạt nhưng cần cẩn thận khi đọc/ghi.

**4.3.Cloud Storage / Firebase**

**Firebase** là nền tảng của **Google** giúp lập trình viên phát triển ứng dụng nhanh chóng mà **không cần tự xây server**.

**Firebase** cung cấp nhiều dịch vụ — trong đó **lưu trữ dữ liệu trên cloud (đám mây)** là phần quan trọng nhất.

**Firebase Cloud Storage** là dịch vụ giúp :

* Lưu file trực tuyến (ảnh, PDF, âm thanh, video…)
* Dễ dàng tải lên (upload) và tải xuống (download)
* Tự động bảo mật bằng Authentication của Firebase

**Ví dụ:**

* Lưu ảnh đại diện người dùng.
* Lưu ảnh sản phẩm.
* Lưu file PDF tài liệu

**4.4. Bảng so sánh các phương thức lưu trử chính trong FLUTTER**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tiêu chí | SharedPreferences | File Storage | SQLite/sqflite | Cloud Storage  /Firebase |
| Kiểu dữ liệu lưu được | Key-Value đơn giản(int,double,bool, String,List<String>) | Bất kỳ(tập tin văn bản JSON,hình ảnh,...) | Dữ liệu có cấu trúc  (bảng,quan hệ) | Dữ liệu mọi loại  (tập tin,JSON,ảnh,...) |
| Cấu trúc lưu trữ | Key-Value  (dictionary nhỏ) | File trong bộ nhớ thiết bị | Cơ sở dữ liệu  SQL | Máy chủ  (Cloud) |
| Dung  lượng | Nhỏ(~ vài MB) | Vừa(phụ thuộc bộ nhớ máy) | Lớn (tối ưu cho hàng ngàn bản ghi) | Không giới hạn   (theo gói Clooud) |
| Tốc độ  truy cập | Nhanh | Trung bình | Nhanh | Phụ thuộc mạng |
| Phạmvi  sử dụng | Lưu cấu hình cài đặt người dùng | Lưu file,tài liệu,JSON confis | Lưu danh sách  sản phẩm,người  dùng.. | Lưu dữ liệu chia sẻ nhiều thiết bị |
| Bảo  mật | Thấp (file có thể đọc nếu root máy) | Trung bình  (tùy mã hóa) | Khá cao | Rất cao |
| Yêu cầu mạng | Không cần | Không cần | Không cần | Cần Internet |
| Thư  viện | Shared\_preferences | Path\_provider,  Dart:io | Spflite,drift | Firebase\_storage,  Cloud\_firestore |
| Dùng  khi | Lưu cài đặt,token   đăng nhập,theme,.. | Lưu file,JSON  cấu hình | Lưu dữ liệu ứng dụng lớn | Lưu dữ liệu đồng bộ, chia sẻ cloud |

Kết luận:

 Dùng SharedPreferences khi ghi nhớ trạng thái giao diện(Dark Mode, ngôn ngữ, font size) hoặc cần lưu những token đăng nhập tạm với

dung lượng ít và nhỏ.

**5.Xử lý lỗi và Best practices**

**5.1.Các lỗi thường gặp khi sử dụng SharedPreferences**

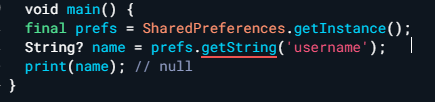
TH1: Null khi đọc dữ liệu

Nguyên nhân:

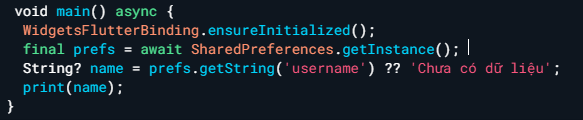
 - Đọc dữ liệu trước khi SharedPreferences khởi tạo.

-   Hoặc khóa (key) chưa tồn tại.

Ví dụ lỗi:



Cách khắc phục:



TH2: Không cập nhật dữ liệu mới

Nguyên nhân:

* Gọi prefs.set...() nhưng quên await
* Gọi trước khi Flutter binding được khởi tạo.

Ví dụ lỗi:

A computer code with colorful text

AI-generated content may be incorrect.

Cách khắc phục:

A screen shot of a computer code

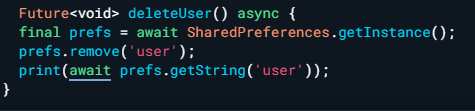
AI-generated content may be incorrect.

* Luôn dùng await prefs.setString(...)

TH3: Không xóa được dữ liệu

Nguyên nhân:

* Dùng sai key hoặc chưa gọi await prefs.remove(key)



Cách khắc phục:

A computer screen shot of text

AI-generated content may be incorrect.

TH4: Không hoạt động trên Web / Desktop

Nguyên nhân:

* SharedPreferences không hỗ trợ tất cả nền tảng như trên mobile

Cách khắc phục:

* Kiểm tra package tương thích (shared\_preferences\_web)

TH5: Lưu JSON dài bằng SharedPreferences

Hiện tượng:

* App lag, freeze hoặc crash trên thiết bị yếu.
* Lúc đọc lại: Có thể null hoặc bị cắt cụt

Cách khác phục:

* Dùng hive, sqflite, hoặc objectbox để lưu cấu trúc dữ liệu phức tạp.

**5.2.Best practices**

* *Best Practices*: Cách làm đúng, hiệu quả và bền vững nhất.:
* Giảm lỗi
* Dễ bảo trì
* Dễ mở rộng ứng dụng
* Giúp code chạy nhanh, an toàn và rõ ràng hơn
* *Best Practices khi dùng SharedPreferences*

|  |  |
| --- | --- |
| Best Practices | Giải thích/ lợi ích |
| Dùng Singleton hoặc lớp helper | Tránh lặp lại code khởi tạo SharedPreferences ở nhiều nơi. |
| Luôn dung await khi đọc/ghi/xóa dữ liệu | Vì các hàm của SharedPreferences là async, nên nếu không await  có thể xảy ra lỗi hoặc dữ liệu cũ. |
| Xác định rõ kiểu dữ liệu khi lưu  (bool,int,double,string,stringList) | Tránh lỗi kiểu dữ liệu không khớp. |
| Đặt Key có cấu trúc rõ ràng | Ví dụ: “user\_token”,”theme\_mode”,”is\_logged\_in” |
| Không lưu dữ liệu nhạy cảm  (mật khẩu, token bảo mật cao) | Vì SharedPreferences không mã hóa. Hãy dùng Flutter\_secure\_storage cho dữ liệu nhạy cảm. |
| Kiểm tra null khi đọc dữ liệu | Tránh crash khi key chưa tồn tại. |
| Dọn dẹp (remove/clear) khi đăng xuất hoặc reset app | Giúp tránh lỗi lưu dữ liệu cũ của người dung trước. |
| Không lạm dụng SharedPreferences để lưu những dữ liệu lớn hay danh sách phức tạp | Chỉ dung cho cài đặt cấu hình không dung như database. |
| Test kỹ luồng khởi tạo  (intiState/FutureBuilder) | Vì SharedPreferences cần thời gian load trước khi dung. |

**6.Tổng kết**

   SharedPreferences là lựa chọn tối ưu nhất cho việc lưu trữ dữ liệu nhẹ và cấu hình người dùng trong Flutter. Khi kết hợp với cách viết code chuẩn và tuân thủ

Best Practices, ứng dụng sẽ: Hoạt động ổn định hơn,dễ mở rộng và bảo trì, mang

lại trải nghiệm người dùng nhất quán qua nhiều lần sử dụng.

**7.TÀI LIỆU THAM KHẢO**

[1]. SharedPreferences Flutter : https://pub.dev/packages/shared\_preferences <https://pub.dev/packages/shared_preferences>

[2]. Youtube: How to Store and Retrieve Data Using Shared Preference in Flutter Application? - Full Tutorial <https://www.youtube.com/watch?v=hiZcVbyukBo>

[3]. Github: afonsocraposo/save-sharedpreferences <https://github.com/afonsocraposo/save-sharedpreferences>

[4]. Github: JohannesMilke/user\_profile\_shared\_preferences\_example <https://github.com/JohannesMilke/user_profile_shared_preferences_example>

[5]. Github: nirmalnyure1/sharedPreferences-in-flutter <https://github.com/nirmalnyure1/sharedPreferences-in-flutter>