



TRƯỜNG ĐẠI HỌC BÁCH KHOA - ĐẠI HỌC ĐÀ NẴNG
KHOA ĐIỆN TỬ - VIỄN THÔNG

LẬP TRÌNH ĐA NỀN TẢNG
SHARED PREFERENCES



Đà Nẵng, tháng 10 năm 2025

Giảng viên hướng dẫn

TS Nguyễn Duy Nhật Viễn

Sinh viên thực hiện

- | | |
|---------------------------|---------|
| 1. Lê Thị Hải Yến | 22KTMT2 |
| 2. Nguyễn Thị Uyên Phương | 22KTMT1 |



Nội dung

- 1 GIỚI THIỆU
- 2 NGUYÊN LÝ HOẠT ĐỘNG
- 3 DEMO ỨNG DỤNG
- 4 SO SÁNH VỚI CÁC PHƯƠNG THỨC KHÁC
- 5 XỬ LÝ LỖI VÀ BEST PRACTICES
- 6 KẾT LUẬN



1. GIỚI THIỆU

Tổng quan lưu trữ trong Flutter:

Flutter có nhiều cách lưu trữ:

- File (JSON, text)
- SQLite (CSDL)
- SharedPreferences (key–value): Nhẹ, nhanh, dễ dùng
- SecureStorage (bảo mật cao)



SharedPreferences

1. GIỚI THIỆU

SharedPreferences là gì?

- Plugin lưu dữ liệu dạng key–value
- Dữ liệu nhỏ (int, bool, string, double, List)
- Lưu cục bộ – không bị mất khi tắt app
- Phù hợp cho cài đặt người dùng, token đăng nhập



SharedPreferences

2. NGUYÊN LÝ HOẠT ĐỘNG

Dữ liệu lưu dưới dạng:

- Key → tên biến
- Value → giá trị được lưu

Ví dụ:

"isDarkMode" : true

"fontSize" : 18.0



2. NGUYÊN LÝ HOẠT ĐỘNG

Gói sẽ tự động biên dịch các lệnh gọi chung thành các API lưu trữ cụ thể của từng nền tảng

Nền tảng	Cách lưu	Định dạng
Android	SharedPreferences API	XML
iOS	NSUserDefaults	plist
Web	LocalStorage	JSON
Windows	LocalSettings	JSON



2. NGUYÊN LÝ HOẠT ĐỘNG

Đặc điểm

- Truy cập nhanh (dữ liệu cache trong RAM)
- Dễ dùng – không cần cấu trúc phức tạp
- Không cần khởi tạo database
- Không phù hợp lưu dữ liệu lớn hoặc bảo mật



2. NGUYÊN LÝ HOẠT ĐỘNG

Các kiểu dữ liệu hỗ trợ:

- bool – giá trị đúng/sai
- int, double – số nguyên, số thực
- String – chuỗi ký tự
- List<String> – danh sách chuỗi



2. NGUYÊN LÝ HOẠT ĐỘNG

1. Lấy instance: `SharedPreferences.getInstance()`
2. Ghi dữ liệu: `setBool()`, `setString()`...
3. Đọc dữ liệu: `getBool()`, `getString()`...
4. Xóa dữ liệu: `remove()` hoặc `clear()`



2. NGUYÊN LÝ HOẠT ĐỘNG



```
1 Future<void> _setTheme(bool isDark) async {  
2   final prefs = await SharedPreferences.getInstance();  
3   await prefs.setBool('isDarkMode', isDark);  
4   setState(() => _isDarkMode = isDark);  
5   ThemeManager.of(context).setTheme(isDark);  
6   _showSnackBar(isDark ? "Đã bật chế độ tối" : "Đã tắt chế độ tối");  
7 }
```

Lấy instance
Ghi dữ liệu



```
1 Future<void> _loadSettings() async {  
2   final prefs = await SharedPreferences.getInstance();  
3   setState(() {  
4     _isDarkMode = prefs.getBool('isDarkMode') ?? false;  
5     _selectedLanguage = prefs.getString('language') ?? _availableLanguages.first;  
6     _fontSize = prefs.getDouble('fontSize') ?? 16.0;  
7   });  
8 }  
9
```

Đọc dữ liệu



SharedPreferences

2. NGUYÊN LÝ HOẠT ĐỘNG



```
1 Center(  
2   child: OutlinedButton.icon(  
3     onPressed: () async {  
4       final prefs = await SharedPreferences.getInstance();  
5       await prefs.clear();  
6       _loadSettings();  
7       _showSnackBar("Đã đặt lại tất cả cài đặt");  
8     },),  
9   ),
```

→ Xóa dữ liệu



SharedPreferences

3. DEMO ỨNG DỤNG

Chuẩn bị:

- Cài package:
flutter pub add shared_preferences

```
1 dependencies:  
2   flutter:  
3     sdk: flutter  
4   shared_preferences: ^2.0.0
```

- Import:
import 'package:shared_preferences/shared_preferences.dart';



SharedPreferences

3. DEMO ỨNG DỤNG

Kết quả demo:

- Người dùng đăng nhập, chọn Dark Mode, Font Size và Language
- Đóng app → mở lại → vẫn giữ nguyên trạng thái đăng nhập và các thiết lập
- SharedPreferences hoạt động nên → không mất dữ liệu



SharedPreferences

4. SO SÁNH VỚI CÁC PHƯƠNG THỨC KHÁC

Phương thức	Dữ liệu	Quy mô	Bảo mật	Dùng cho
SharedPref.	Key-value	Nhỏ	Thấp	Cài đặt
File	JSON, text	TB	TB	Cache
SQLite	CSDL	Lớn	TB	Dữ liệu phức tạp
SecureStorage	Key-value	Nhỏ	Cao	Token, mật khẩu



4. SO SÁNH VỚI CÁC PHƯƠNG THỨC KHÁC

Khi nào dùng sharedPreferences?

- Cài đặt giao diện
- Ngôn ngữ, font chữ
- Lưu cờ trạng thái (đã đăng nhập)

Vd: + App cài đặt Dark Mode (on/off)

+ Ghi nhớ người dùng từng chọn “đăng nhập tự động”

+ Lưu ngôn ngữ “vi/en” để hiển thị khi khởi động

- Không dùng cho danh sách dữ liệu lớn
- Không lưu thông tin nhạy cảm



SharedPreferences

5. XỬ LÝ LỖI VÀ BEST PRACTICES

Các lỗi phổ biến:

- Dùng sai mục đích (lưu dữ liệu nhạy cảm)
- Xử lý null sai cách
- Gọi sai thời điểm khởi tạo
- Sai kiểu dữ liệu
- Vấn đề cache và đồng bộ dữ liệu



5. XỬ LÝ LỖI VÀ BEST PRACTICES

- Khởi tạo trong initState()

```
void initState() {  
  super.initState();  
  loadPrefs();  
}
```

- Mỗi key nên duy nhất, tránh ghi đè, quản lý key tập trung

```
class PrefKeys {  
  static const theme = 'isDarkMode';  
  static const fontSize = 'fontSize';  
}
```



5. XỬ LÝ LỖI VÀ BEST PRACTICES

- Tải dữ liệu trước khi chạy App → Tránh hiện tượng nháy theme khi mở app
`WidgetsFlutterBinding.ensureInitialized();`
`prefs = await SharedPreferences.getInstance();`
`runApp(MyApp());`
- Không xóa toàn bộ dữ liệu khi reset
→ Chỉ xóa các key cần thiết



5. XỬ LÝ LỖI VÀ BEST PRACTICES

- Tạo lớp quản lý riêng

```
class SettingsService {  
    final prefs = await SharedPreferences.getInstance();  
    Future<void> setTheme(bool value) => prefs.setBool('isDarkMode', value);  
}
```



5. XỬ LÝ LỖI VÀ BEST PRACTICES

- Không lưu dữ liệu nhạy cảm
Dùng flutter_secure_storage thay vì SharedPreferences
- Đặt giá trị mặc định → Tránh lỗi null khi key chưa tồn tại
`prefs.getBool('isDarkMode') ?? false;`



5. XỬ LÝ LỖI VÀ BEST PRACTICES

- Giới hạn tần suất ghi
 - Không gọi `set...()` liên tục trong vòng lặp
 - Ghi khi người dùng thực sự thay đổi
- Xử lý lỗi bằng try-catch

```
try {  
    await prefs.setString('user', 'abc');  
} catch(e) {  
    print('Save error: $e');  
}
```



5. XỬ LÝ LỖI VÀ BEST PRACTICES

- Đối tượng phức tạp → lưu dưới dạng JSON string

Quy trình:

- + toJson() → jsonEncode() → prefs.setString()
- + prefs.getString() → jsonDecode() → fromJson()

Ví dụ: Lưu thông tin user, cấu hình, danh sách thú cưng.



6. KẾT LUẬN

SharedPreferences: giải pháp lưu trữ nhỏ gọn, dễ dùng
Phù hợp cho:

- Cài đặt người dùng
- Trạng thái giao diện

Không phù hợp cho dữ liệu lớn hoặc bảo mật

Kết hợp state management → App mượt, ổn định hơn



SharedPreferences

The background features a vibrant, abstract design. It consists of numerous overlapping, wavy lines in shades of light blue, teal, and pale yellow, creating a sense of movement and depth. Superimposed on these are several sharp, geometric shapes in darker blue and grey tones, which appear to be floating or layered on top of the wavy patterns. The overall effect is a modern, artistic composition.

Thank You