



OTUS

ОНЛАЙН-ОБРАЗОВАНИЕ

# Онлайн-образование

О С WWW.SW.HELP - ПРИСОЕДИ

Проверить, идет ли запись!





# Меня хорошо видно && слышно?

Ставьте  , если все хорошо  
Напишите в чат, если есть проблемы

# Маршрут вебинара

**File**



**Shared Preference**



**Hive**



**SQLite**

# Цели вебинара | После занятия вы сможете

1

На практике освоить работу с такими инструментами как File, SharedPreferences, Hive, SQLite

2

Понять для каких целей, какой инструмент больше подходит

# Данные приложения

Размещение данных приложения

Локально

Удаленно

# Сравнение

	File	SharedPreferences	Hive	SQLite
Инициализация				
Добавление ключ/значение				
Работа с коллекциями				
Поиск				



# File



# File

## Основные методы

### Чтение

`openRead`

Побайтовое чтение файла через Stream

`readAsBytes / readAsBytesSync`

Возвращает все содержимое файла в виде списка байтов

`readAsString / readAsStringSync`

Возвращает все содержимое файла в виде текста с использованием заданной кодировки. (По умолчанию utf8)

`readAsLines / readAsLinesSync`

Возвращает все содержимое файла в виде строк текста с использованием заданной кодировки. (По умолчанию utf8)

# File

## Основные методы

### Запись

`openWrite`

Создает новый независимый [IOSink] для файла.

`writeAsBytes/writeAsBytesSync`

Записывает список байтов в файл.

`writeAsString/writeAsStringSync`

Записывает текст в файл с использованием заданной кодировки.  
(По умолчанию utf8)

# File

## Параметры записи

mode

FileMode.append /writeOnlyAppend - добавляет содержимое в конец текущих данных  
FileMode.write /writeOnly - перезаписывает содержимое поверх текущих данных

# File

## Файловая система

Для работы с файловой системой устройства удобно использовать плагин **path\_provider**

[https://pub.dev/packages/path\\_provider](https://pub.dev/packages/path_provider)

# path\_provider

## Основные методы

getTemporaryDirectory()

getApplicationSupportDirectory()

getApplicationDocumentsDirectory()

getLibraryDirectory()

getExternalCacheDirectories()

getExternalStorageDirectory()

getExternalStorageDirectories()

getDownloadsDirectory()

# path\_provider

Только Android

Android/iOS

Только IOS

# path\_provider

Только Android

Android/iOS

Только IOS

getLibraryDirectory

# path\_provider

Только Android

Android/iOS

Только IOS

getExternalStorageDirectory

getExternalStorageDirectories

getExternalCacheDirectories

# path\_provider

Только Android

Android/iOS

Только IOS

getExternalStorageDirectory

getApplicationDocumentsDirectory

getLibraryDirectory

getExternalStorageDirectories

getTemporaryDirectory

getExternalCacheDirectories

getApplicationSupportDirectory

# path\_provider

getTemporaryDirectory

Путь к временному каталогу на устройстве, для которого не выполняется резервное копирование и которое подходит для хранения кэшей загруженных файлов.

В **iOS** это использует API `NSCachesDirectory`.

В **Android** это использует API `getCacheDir`.

# path\_provider

```
getApplicationDocumentsDirectory
```

Путь к каталогу, в котором приложение может размещать данные, созданные пользователем.

В **iOS** здесь используется API **NSDocumentDirectory**.

В **Android** это использует API `getDataDirectory`

# path\_provider

`getApplicationSupportDirectory`

Путь к каталогу, в котором приложение может разместить файлы для поддержки приложения.

Используйте это для файлов, которые вы не хотите показывать пользователю. Ваше приложение не должно использовать этот каталог для файлов пользовательских данных.

В **iOS** здесь используется API **NSApplicationSupportDirectory**. Если этот каталог не существует, он создается автоматически.

В **Android** эта функция использует API `getFilesDir`.

# path\_provider

Вопросы:

1. Какие платформы поддерживает метод `getExternalCacheDirectories()`. (Android/iOS/обе).
2. Какой метод вернет раздел, который больше всего подходит для хранения пользовательских данных?
  - a) `getTemporaryDirectory`
  - б) `getApplicationDocumentsDirectory`
  - г) `getApplicationSupportDirectory`

An aerial view of a city skyline, likely New York City, with a blue and green color overlay. The image is split horizontally into two sections. The top section shows a dense cluster of skyscrapers, and the bottom section shows a similar view from a slightly different angle. A central horizontal band features a network of glowing blue and green lines connecting various points, creating a digital or data network aesthetic. The text 'flutter\_secure\_storage' is centered in this band in a white, bold, sans-serif font.

**flutter\_secure\_storage**

# flutter\_secure\_storage

[https://pub.dev/packages/flutter\\_secure\\_storage](https://pub.dev/packages/flutter_secure_storage)

Плагин Flutter для хранения данных в безопасном хранилище::

- [Keychain](#) используется для iOS
- Для Android используется шифрование AES. Секретный ключ AES зашифрован с помощью RSA, а ключ RSA хранится в KeyStore

*Note* KeyStore was introduced in Android 4.3 (API level 18). The plugin wouldn't work for earlier versions.

The image features a central horizontal band with a blue-to-green gradient background. This band is overlaid with a white network diagram consisting of interconnected nodes and lines. The text "shared\_preference" is centered within this band in a white, bold, sans-serif font. The top and bottom portions of the image show an aerial view of a city skyline, with numerous skyscrapers and buildings, all rendered in a monochromatic blue and green color scheme that matches the central band's background.

**shared\_preference**

# shared\_preference

[https://pub.dev/packages/shared\\_preferences](https://pub.dev/packages/shared_preferences)

Плагин для работы с постоянным хранилищем простых данных (key/value) (NSUserDefaults в iOS и macOS, SharedPreferences в Android и.т. д).

Данные могут сохраняться на диске асинхронно, и нет гарантии, что записи будут сохраняться на диск после возврата, **поэтому этот плагин нельзя использовать для хранения критических данных!**

# shared\_preferences. Основные методы

## Запись

setString

setBool

setDouble

setInt

setStringList

## Получение

get

getKeys

getString

getBool

getInt

getDouble

getStringList



# Hive









# Hive

<https://pub.dev/packages/hive>

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

Hive - это легкая и невероятно быстрая база данных ключ-значение, написанная на чистом Dart.

-  Cross platform: mobile, desktop, browser
-  Great performance (see [benchmark](#))
-  Simple, powerful, & intuitive API
-  Strong encryption built in
-  NO native dependencies
-  Batteries included

# Hive

## Вох

Все данные, хранящиеся в Hive, организованы с помощью vox.

Вох можно сравнить с таблицей в SQL, но он не имеет структуры и может содержать что угодно.

Для небольшого приложения может быть достаточно одного Вох.

Вох может быть зашифрован

# Hive

## Box

name

Указывает место хранения и используется для проверки, наличия box. **Регистр не учитывается.**

encryptionKey

Ключ должен быть массивом байтов длиной 32 и использоваться для шифрования и дешифрования всех значений в поле.

keyComparator

По умолчанию ключи отсортированы лексикографически. Этот параметр позволяет указать собственный порядок сортировки.

compactionStrategy

Позволяет задать свои собственные правила автоматического уплотнения.

crashRecovery

Если приложение убито во время операции записи, последняя запись может быть повреждена. Эта запись удаляется автоматически при запуске приложения.

path

По умолчанию ящики хранятся в каталоге, указанном для Hive.init (). С помощью этого параметра вы можете указать место, где должен храниться ящик.

bytes

Вместо использования файла в качестве бэкэнда вы можете предоставить поле в двоичной форме и открыть окно в памяти.

# Hive

Box

Vs

LazyBox

- Хранит все данные в памяти

+ Не хранит данные в памяти

+ Имеет синхронный доступ к элементам

- Элементы можно получить только асинхронно

# Hive

## TypeAdapter

Hive поддерживает все примитивные типы: List, Map, DateTime и Uint8List.  
Если вы хотите хранить другие объекты, нужно зарегистрировать TypeAdapter, который преобразует объект из двоичной формы и обратно.

# Hive. TypeAdapter

## Самостоятельно

```
class UserAdapter extends TypeAdapter<User> {  
    @override  
    final typeId = 0;  
  
    @override  
    User read(BinaryReader reader) {  
        return User(reader.read());  
    }  
  
    @override  
    void write(BinaryWriter writer, User obj) {  
        writer.write(obj.name);  
    }  
}
```

## Кодогенерация

Используем

[hive generator](#)

```
@HiveType(typeId : 1)  
class Person {  
    @HiveField(0)  
    String name;  
  
    @HiveField(1)  
    int age;  
  
    @HiveField(2)  
    List<Person> friends;  
}
```



# SQLite



# sqflite

<https://pub.dev/packages/sqflite>

<https://github.com/tekartik/sqflite/blob/master/sqflite/doc/sql.md>

Плагин SQLite для Flutter. Поддерживает iOS, Android и MacOS.

Поддержка **transactions** и **batches**

Автоматическое управление версиями во время открытия

Помощники для запросов вставки / запроса / обновления / удаления

Операция БД выполняется в фоновом потоке на iOS и Android

# sqflite

## Поддерживаемые типы данных

### **INTEGER**

- Dart type: int
- Supported values: from  $-2^{63}$  to  $2^{63} - 1$

### **REAL**

- Dart type: num

### **TEXT**

- Dart type: String

### **BLOB**

- Dart type: Uint8List

# sqlite

## Создание базы данных

```
Database database = await openDatabase(path, version: 1,  
    onCreate: (Database db, int version) async {  
    // создаем таблицы  
    await db.execute(  
        'CREATE TABLE Test (id INTEGER PRIMARY KEY, name TEXT, value INTEGER, num REAL)');  
    });
```

# sqlite

Добавление значений

```
int id1 = await txn.rawQuery(  
    'INSERT INTO Test(name, value, num) VALUES("some name", 1234, 456.789)');
```

Или

```
await db.insert(tableTodo, todo.toMap());
```

# sqlite

## Обновление значений

```
await database.rawQuery(  
    'UPDATE Test SET name = ?, value = ? WHERE name = ?',  
    ['updated name', '9876', 'some name']);
```

## Или

```
await db.update(tableTodo, todo.toMap(),  
    where: '$columnId = ?', whereArgs: [todo.id])
```

# sqlite

Получение значений


```
await database.rawQuery('SELECT * FROM Test')
```

Или

```
await db.query(tableTodo,  
  columns: [columnId, columnDone, columnTitle],  
  where: '$columnId = ?',  
  whereArgs: [id])
```

# Сравнение

	File	SharedPreferences	Hive	SQLite
Инициализация	+	+	-	-
ключ/значение	-	+	+	+
Работа с коллекциями	-	-	+/-	+
Поиск	-	-	+/-	+

The image features a blue-tinted aerial view of a city skyline, likely New York City, with numerous skyscrapers. A semi-transparent blue band with a white network pattern of dots and lines runs horizontally across the middle of the image. The text is centered within this band.

Заполните, пожалуйста,  
опрос о занятии по ссылке в чате