



# Сохранение **данных**

# Зачем хранить данные?



- Поддержка offline mode
- Сохранение настроек и пользовательских данных
- Сохранение данных и файлов на временной основе

# Offline mode



- Возможность функционирования приложения без интернета
- Локальное сохранение нужных данных до появления интернета
- Лучше всего для этого использовать БД





# Пользовательские данные

Что важно учесть:

- Миграция между устройствами в рамках Apple ID
- Резервные копии
- Возможность удалить данные
- Безопасность чувствительных данных

# Cache



- Оптимизация скорости работы приложения
- Снижения нагрузки на backend
- Экономия батареи и трафика пользователя
- Данные могут в любой момент быть удалены системой
- Не должны быть жизненно важными для приложения

# Инструменты



- CoreData
- БД
- UserDefaults
- KeyChain
- FileManager

# CoreData



- не база данных
- фреймворк
- управляет графом объектов
- внутри может работать с SQLite
- not thread safe

# SQLite



- база данных
- позволяет иметь одну реализацию на разных платформах iOS и Android



# UserDefaults



- интерфейс для хранения
- ключ - значение
- NSKeyArchiver/NSKeyUnarchiver
- Позволяет передавать данные между app extensions и app
- thread safe

# KeyChain



- Часть фреймворка Security
- Зашифрованная база данных
- Имеет не очень удобное API

# Безопасность



Все инструменты обладают шифрованием данных при хранении. Но для обеспечения полной защиты данных стоит пользоваться KeyChain.

# UserDefaults



## Getting the Standard User Defaults Object

```
class var standard: UserDefaults
```

Returns the shared defaults object.

## Creating User Defaults Objects

```
init()
```

Creates a user defaults object initialized with the defaults for the app and current user.

```
init?(suiteName: String?)
```

Creates a user defaults object initialized with the defaults for the specified database name.

## Getting Default Values

```
func object(forKey: String) -> Any?
```

Returns the object associated with the specified key.

```
func url(forKey: String) -> URL?
```

Returns the URL associated with the specified key.

```
func array(forKey: String) -> [Any]?
```

Returns the array associated with the specified key.

```
func dictionary(forKey: String) -> [String : Any]?
```

Returns the dictionary object associated with the specified key.

```
func string(forKey: String) -> String?
```

Returns the string associated with the specified key.

```
func stringArray(forKey: String) -> [String]?
```

Returns the array of strings associated with the specified key.

```
func data(forKey: String) -> Data?
```

Returns the data object associated with the specified key.

```
func bool(forKey: String) -> Bool
```

Returns the Boolean value associated with the specified key.

# Работаем с UserDefaults и KeyChain



- Вместе сохраняем типизированные данные
- Сохраняем любой класс через Data Codable
- Сохраняем данные в KeyChain
- Задание - Сохраняете через NSSecureCoding

# Файловая система



- Bundle приложения
- Container/data
- Свой sandbox
- FileManager - интерфейс работы с ней

# FileManager



- Позволяет создавать, удалять, копировать и перемещать файлы и директории
- Разные директории имеют разное поведение

# FileManager



- Documents - для хранения сопроводительных файлов в приложении (persistent)
- Application Support - для хранения все файлов для работы приложения (persistent)
- Cache - для хранения любых файлов (not persistent)



# FileManager



21:33

Apps Manager

<

7061C303-5BEF-4ACD-9A9A-3901...

Edit

Name

Date

Size

<

.com.apple.mob...r.metadata.plist

381 bytes

i

PLIST

Mar 21, 2021 19:58

Library

i

>

Documents

i

>

tmp

i

>

StoreKit

i

>

SystemData

i

>

/dev/disk0s1s2

38.0 GB

6 item(s)

21:33

Apps Manager

< Back

Library

Edit

Name

Date

Size

<

com.amplitude.database

61 KB

i

Apr 15, 2021 10:29

com.amplitude.plist

239 bytes

i

Mar 24, 2021 17:55

Caches

i

>

Application Support

i

>

Saved Application State

i

>

WebKit

i

>

SplashBoard

i

>

Preferences

i

>

Cookies

i

>

/dev/disk0s1s2

38.0 GB

9 item(s)



# Работаем с FileManager

- Вместе смотрим на интерфейс и пробуем сохранять файлы
- Работа через `Data.write()`, `string.write()`
- Задание - перемещать и копировать файлы