ackee edu

We know how
We know how
We know how

OK

BI-IOS

Lukáš Hromadník, Jakub Olejník, Igor Rosocha

Lecture 10 - Persistence

Persistence

storing data permanently, after closing the app / releasing memory

there are numerous ways to do so

we’ll go over some of them

ackee edu

Persistence options

- ❏ `FileManager`
- ❏ `UserDefaults`
- ❏ `Core Data`
- ❏ `iCloud` / `CloudKit`
- ❏ Third-party options (`Realm`, `SQLite`, and many more..)

File System

application sees iOS file system like a UNIX filesystem

there are file protections - you can’t see anything

you can only read and write in your application’s ”sandbox”

Why sandbox?

Security - no one else can damage your app’s data

Privacy - no other app can view your app’s data

Cleanup - when you delete the app, sandbox is cleared

Backup - certain parts of sandbox are backed up during device backup

Sandbox

Application directory - app’s executable (not writable)

Documents directory - Permanent storage always visible to the user

Application support directory - Permanent storage not seen directly by the user

Caches directory - Temporary files (not backed up)

whole lot of more directories…

FileManager

Used to browse (read/write) file system, along with `URL`

Shared singleton `default` available

Provides utility operations (`fileExits`, `isExecutableFile` etc.)

Thread safe

Let’s code! 🤓

UserDefaults

lightweight storage for user preferences

”ancient” API - predates Swift

can only store a `Property List`

powerful way is to take advantage of `Codable`, as `Data` is a `Property List`

UserDefaults

has a lot of uses of type `Any` (which basically means ”untyped”)

Swift is a strongly-typed language, bud supports `Any` for backwards compatibility

Let’s code! 🤓

Core Data

object-oriented database

based on SQL

app interacts with data in an entirely object-oriented way

we can perform custom predicates and sorting

plugs beautifully into SwiftUI

Core Data

the heart of `Core Data` is creating a map between objects and ”tables and rows” of a relational databse

Xcode has a built-in graphical editor for this map

also lets us graphically create ”relationships” that point to other objects

Core Data

Xcode will generate classes for objects we specified in the map

these objects serve as VMs for our UI

we can add custom extensions and computed vars

Let’s code! 🤓

CloudKit

storing data into the cloud database

data are synced to all user’s iCloud devices

plays nicely with `Core Data`

simple to use, but required some thoughtful async programming

CloudKit

enabling `CloudKit` capability is required

has a web-based UI dashboard, which shows all the records

`CKContainer`, `CKRecord`, `CKReference`

Questions?

2. úkol

❗ **Deadline (13.12.2022 23:59:59)**

Thank you very much,
see you next week! 🙌