4th Day - iOS Swift Training in Barcelona

Accessible places

The .app is what you ship
The app is a bundle.
From within the app you can access the bundle itself, Documents, Library, tmp.
How to access the 4 folders? For the bundle
Import Foundation and
Bundle.main.resourcePath property.
You will need to create an url. In iOS url are paths

File Manager

FileManager is the class to manipulate files in the filesystem
FileManager.default

Data structure represents buffers (e.g. for streaming video, ..)
String structure allows you to copy a text file into a string, for example
In iOS platforms you can use also NSString class in Swift (there are a lot of legacy methods)

To convert a Swift 2.0 project in Swift 3 there is a tool. However It doesn't change NSURL into URL automatically.

Preferences

Are persistent stored informations to configure something.
UserDefaults class (NSUserDefaults in Obj-c) is a singleton to manage small amount of informations. It is a small database.
You can access it from any point of the app.
You have two options:

- make your settings embedded into the app
- Make your settings accessible from the apple System settings

let defaults = UserDefault.standard

If you delete the app and reinstall it quickly you will find the same settings again. If you wait too long and then reinstall you will lose the settings.

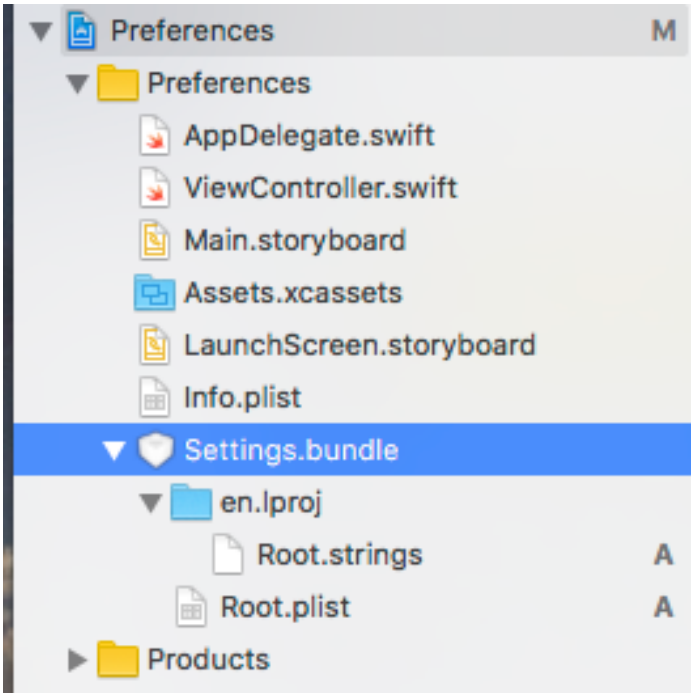
If two app have the same bundle identifier they can share the same UserDefaults settings.

System Settings

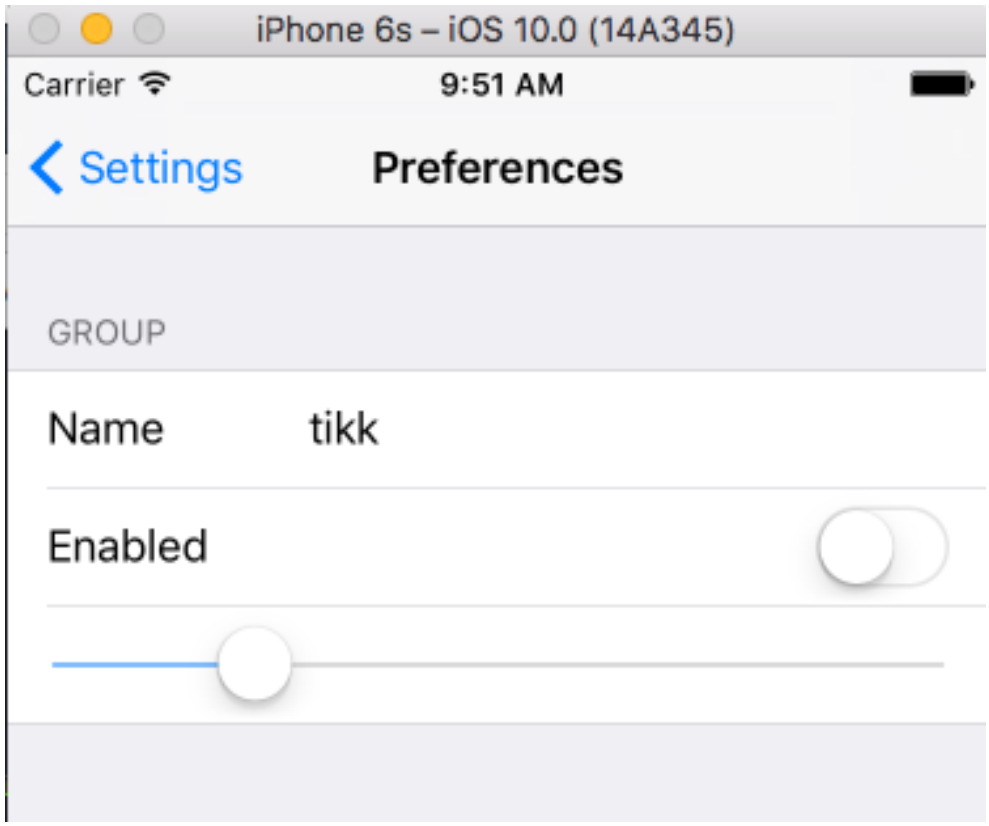
How to access the info from within the System Settings?
Add a special file to the app. It is a bundle containing the Root.plist xml file which is a table view.
You can create groups of settings.

go to Preferences app..

Ctrl + click on Preferences folder and add a file



There is a bug in Xcode 8, wait for 8.1



Localization

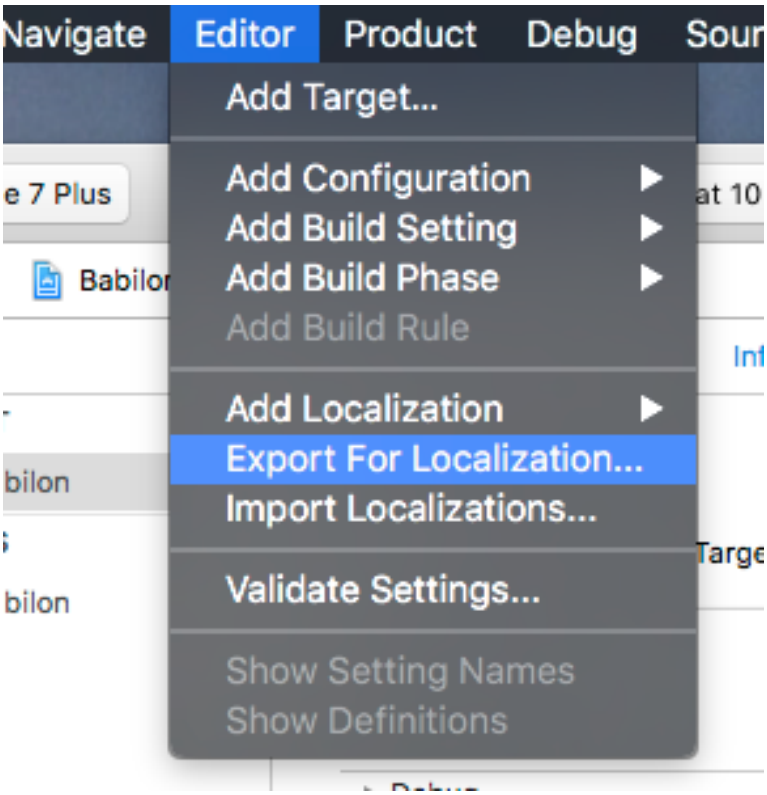
An app may be distributed in many different countries.
For nib file there is no problem.
For strings that you put in the UI you should use use NSLocalizedDescription.

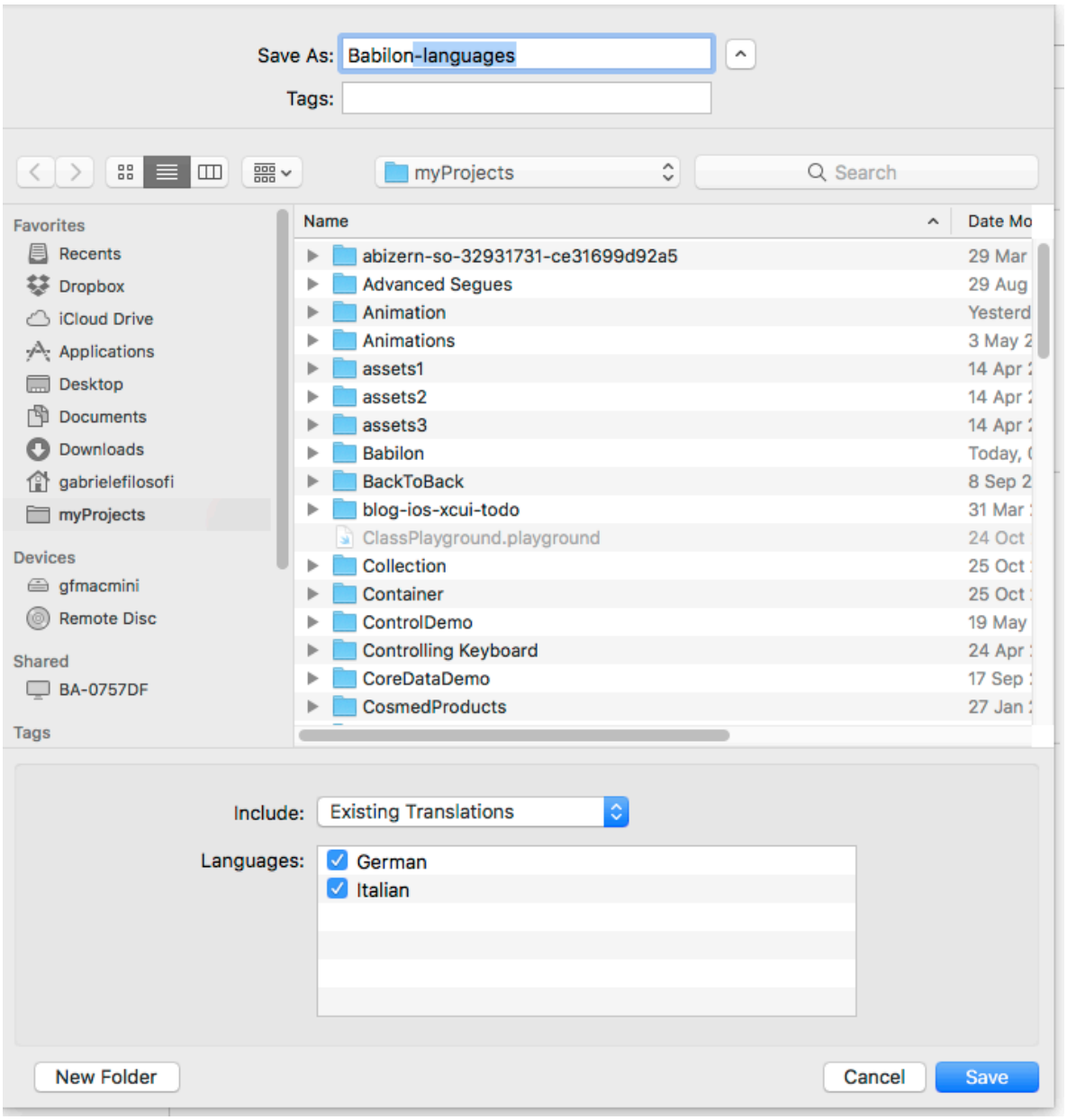
There is a Locale class (NSLocale in obj-c) that allows you to manipulate plural, singular, country code, currency, UM, ... and more related to localization.

In iOS 10 there is also Measurements class.

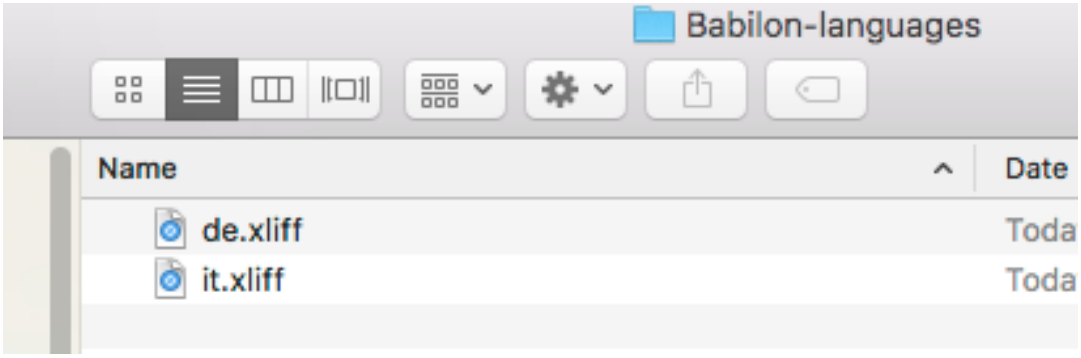
As regards the translation, Autolayout is very important, because strings have different lengths in different languages.

go to Babylon app..

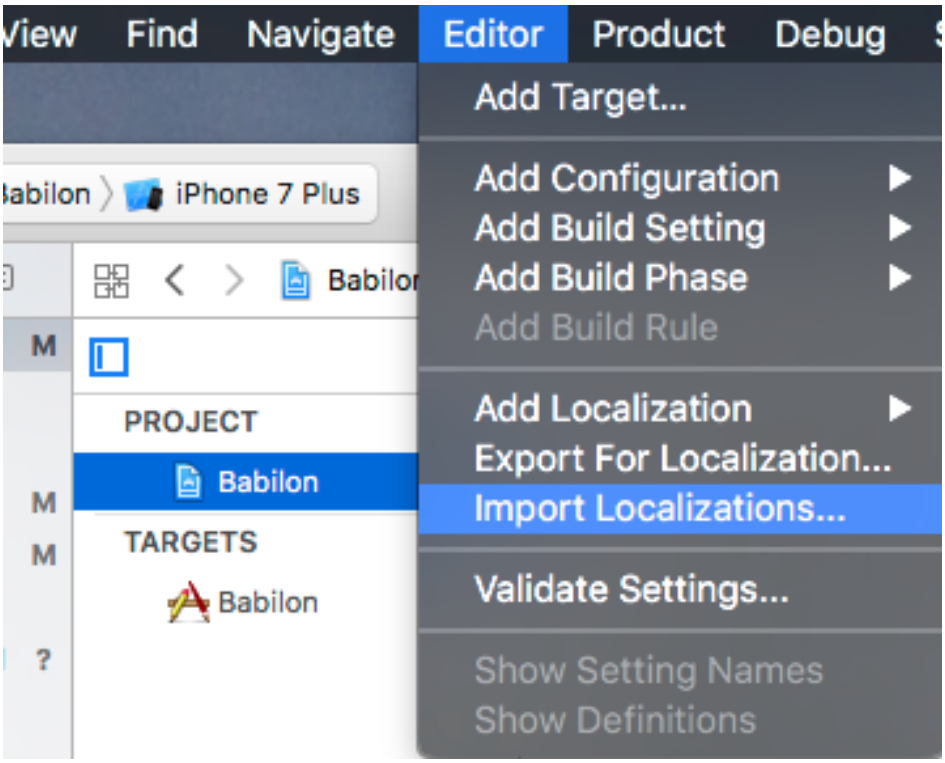

Navigate
Editor
Product
Debug
Sour
Add Target...
Add Configuration
Add Build Setting
Add Build Phase
Add Build Rule
Add Localization
Export For Localization...
Import Localizations...
Validate Settings...
Show Setting Names
Show Definitions
e 7 Plus
at 10
Babilor
Inf
bilon
bilon
Targe

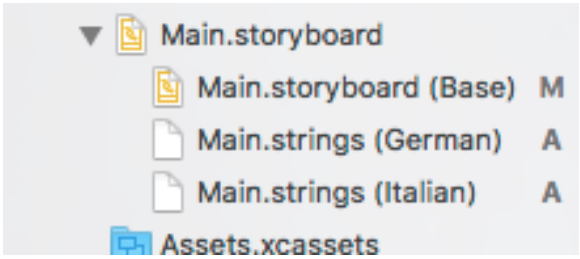


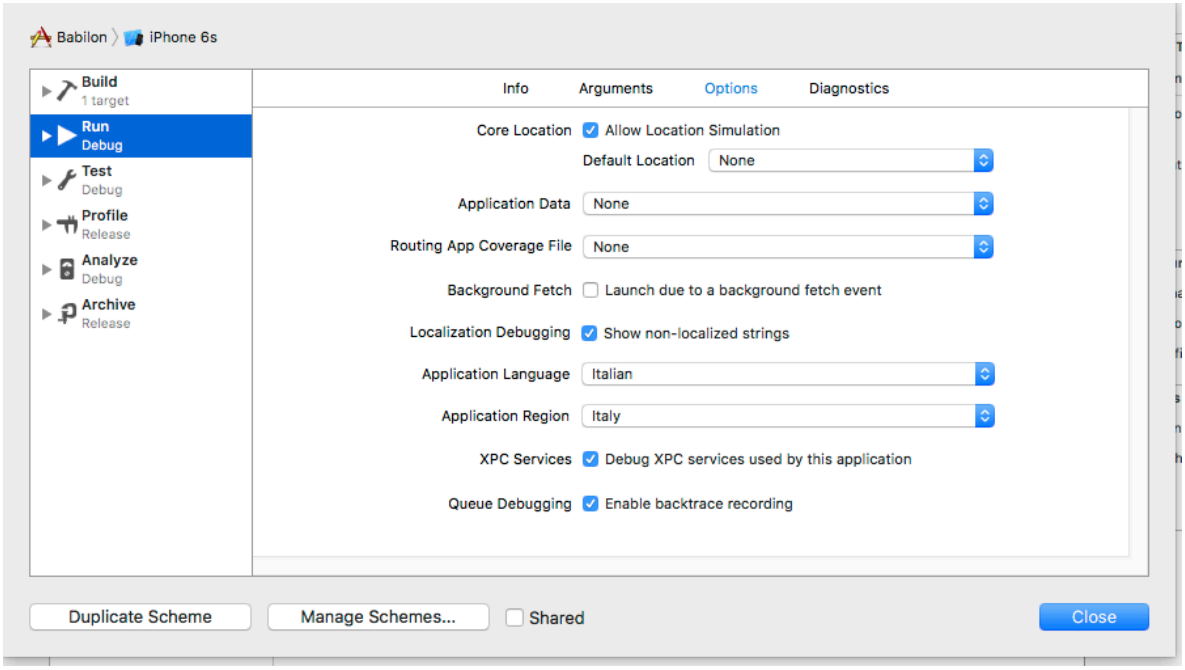
The Counterparts app can open the xliff files.



It allows you to edit and translate expressions.
Then you will import the edited xliff files into Xcode.

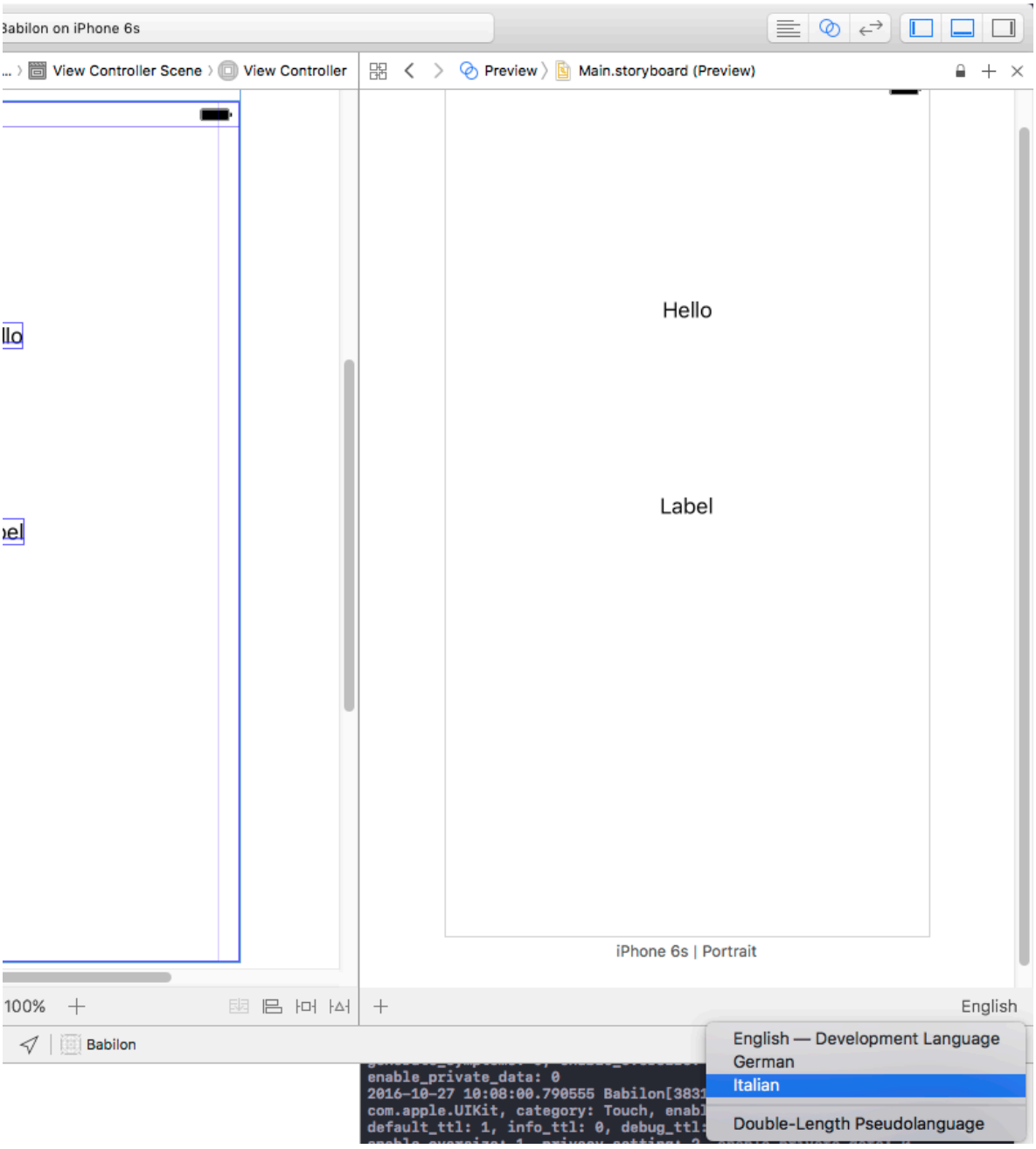






To test the translations, compile, go to preview and select the language from

the bottom right panel



You can do the translation at any given moment in the app development.

Core Graphics exercise:

go to Paint app ..

In this app we are going to mix Core Graphics and Core Animation.

Tip: The same holds in tvOS and Mac. In Mac the view origin is on bottom left corner. In iOS and tvOS the origin is on top left.

Networking

ATS (App Transport Security)

Apple is imposing some new protocols to increase the security.
Starting from iOS 8 you can use IPv6. In iOS 10 it is mandatory!
Apple enforces HTTPS.

ATS configuration:
Into the .plist file add and edit the NSAppTransportSecurity dictionary.
You can also import additional cyphers.

Tip: If the app doesn't connect to the server may be there is a special configuration of that server that you need to include in the ATS configuration.

ATS Diagnostics:
add CF_NETWORK_DIAGNOSTIC = 1

URL connections

NSURLSession and CFNetwork are already IPv6
You can make the mac do the DNS service.

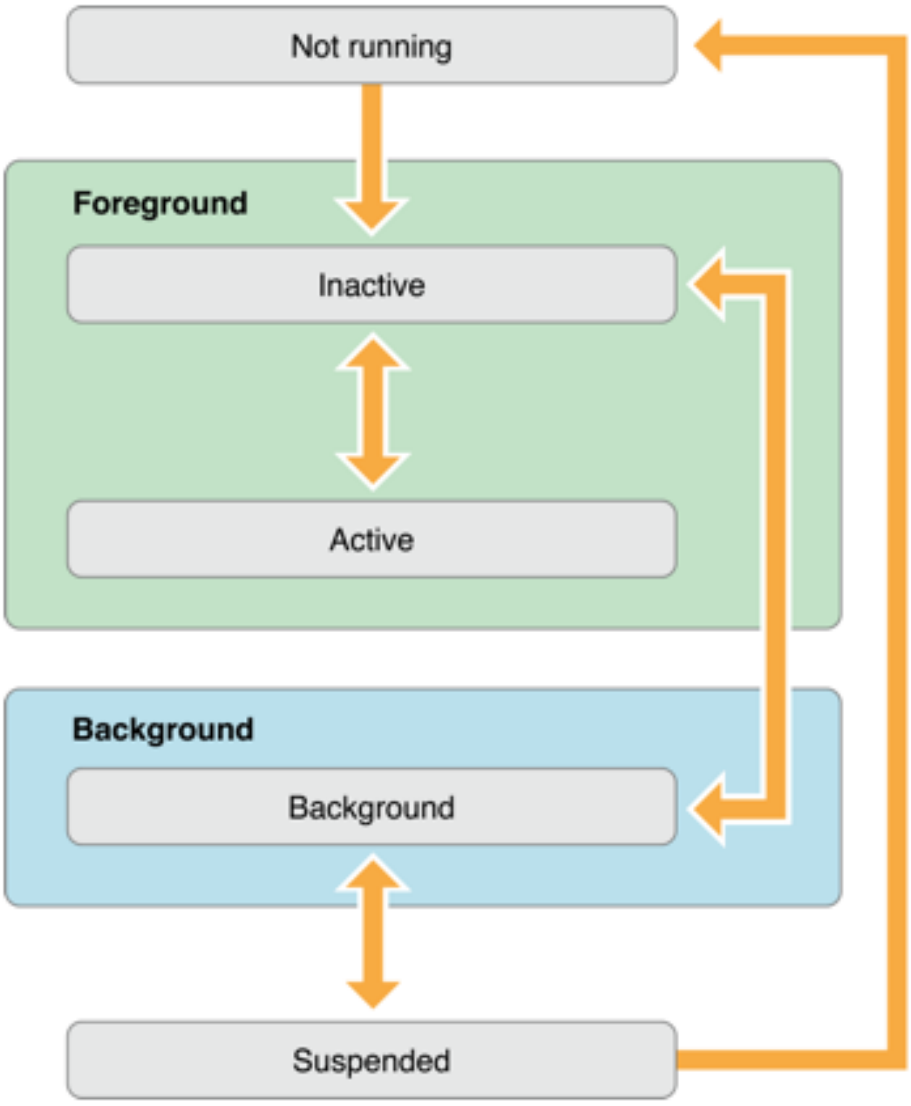
NSURLSession (URLSession in Swift 3.0)
the oldest version NSURLConnection is bad. Do not use it

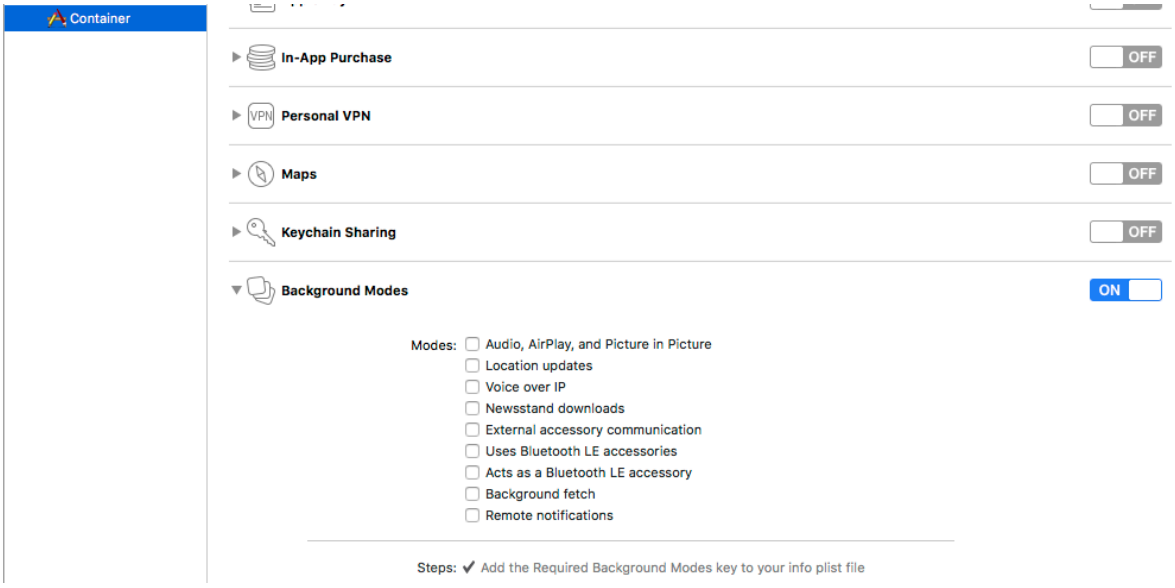
Type of session. 3 types
- Default. the disk is the cache
- Ephemeral. the memory is the cache
- Background. allows background upload and download of files

Set the task
- Data
- Download
- Upload

Background Transfers

Tip: make the Inactive state transition very fast.


Not running
Foreground
Inactive
Active
Background
Background
Suspended


Container
In-App Purchase
OFF
Personal VPN
OFF
Maps
OFF
Keychain Sharing
OFF
Background Modes
ON
Modes:
Audio, AirPlay, and Picture in Picture
Location updates
Voice over IP
Newsstand downloads
External accessory communication
Uses Bluetooth LE accessories
Acts as a Bluetooth LE accessory
Background fetch
Remote notifications
Steps:
Add the Required Background Modes key to your info plist file

JSON Parser

JSONSerialization class
NSArray is not an array.

Convert DATA in JSON

go to Earthquakes app.. (enable Core Data)

The app has 3 VC
a TableView where we use all earthquake
The 2nd VC is a map. We seta pin collout with a button that go to
web view with info abut the earthquake
We need a navigation controller
In the VC1 we put a button to update the list using URLSession and download JSON data

Core Data

it is designed for the UI
NSManagedObject
NSManagedObjectContext (moc. You can have more of them)
NSPersistentStoreCoordinator (do not touch it)
NSManagedObjectModel (In Xcode there is a special tool to fill it)
NSFetchRequest (it is like a query for a database)

you can add multiple persistent store (sqLite files, binary files, memory)
It is very common to have two, a sqLite one and a memory one

In iOS 10 we work with

NSPersistentContainer
In the AppDelegate.swift you will find
    lazy var persistentContainer: NSPersistentContainer = {..

Fetching Data

How to request object?
Create a NSFetchRequest

Data is stored in Entities, tables with columns called Attributes.

Before iOS 10:
- first define the entity (NSEntityDescription)

- You can filter the results by Predicates (NSPredicate)
- You can sort the results by attribute with Sorting (NSSortDescription)
- Then you Fetch (NSFetchRequest)

In iOS 10 and Swift 3.0 it is more simple to define the entity.

Saving data

.save()

Fault

there are pointer objects not filled with real data, in order to save RAM. Only

Batching

If your Table View shows only 12 rows in a screen, Core Data will fetch the faults 12 by 12

Aggregates

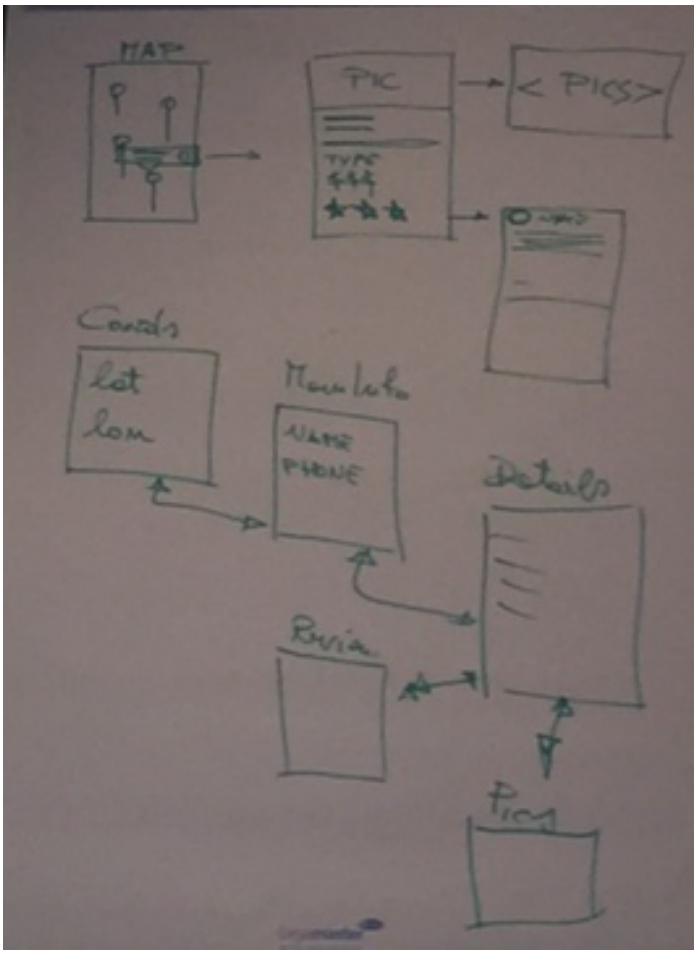
to do calculation directly in sqlLite

Debugging

set some special settings in the Xcode scheme

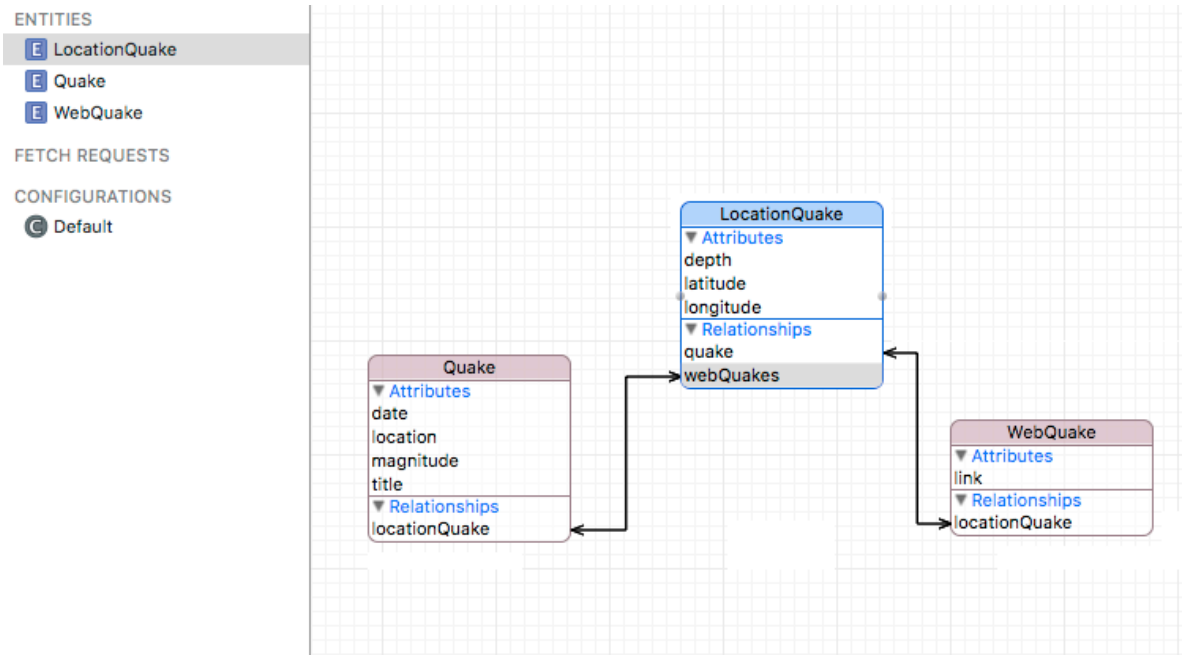
How to structure the data model ?

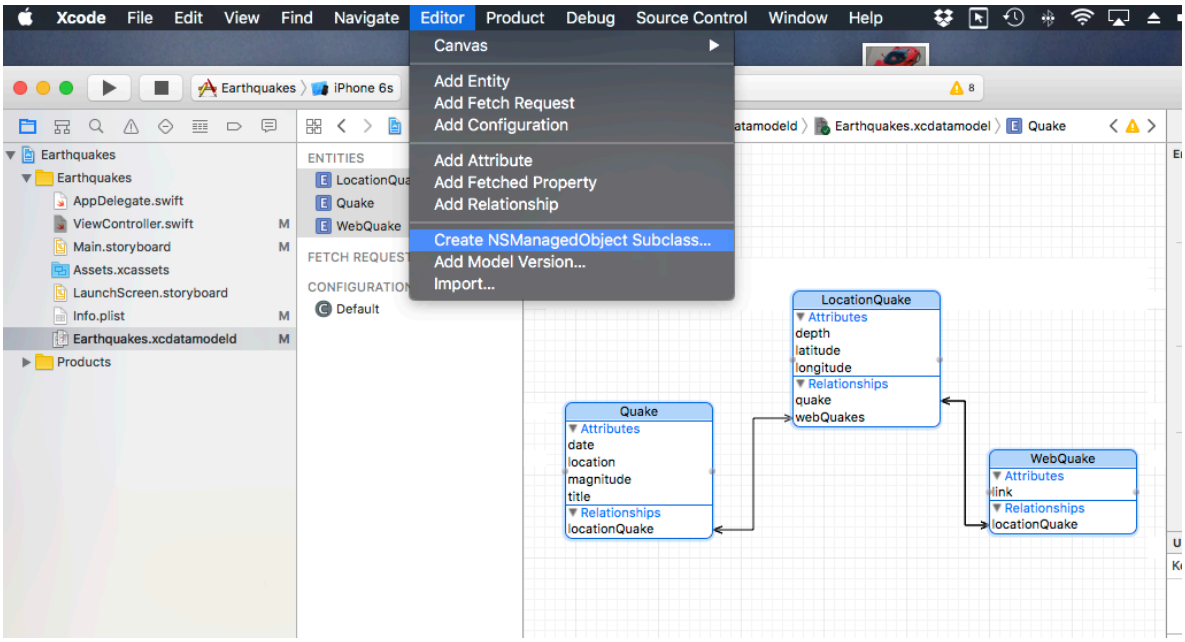
Tip: THE DATA MODEL MUST FOLLOW THE UI!!!
Example: restaurant in Barcelona.
The first view is a map with a pin for each restaurant
then I create a table with coordinates only
When user tap the pin he gets the name and phone numbers
the I create a table with name and phone
Then we create a relation between the two tables
Then the user may tap the button and open a details view ..
If you change the UI you have to remap the data model.



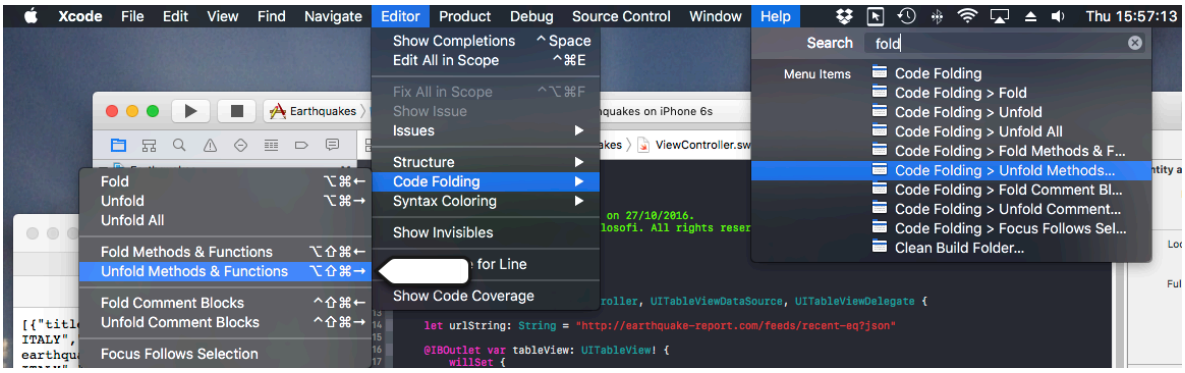
go to the Earthquake app ..

Create and edit attributes in entities and create relationships





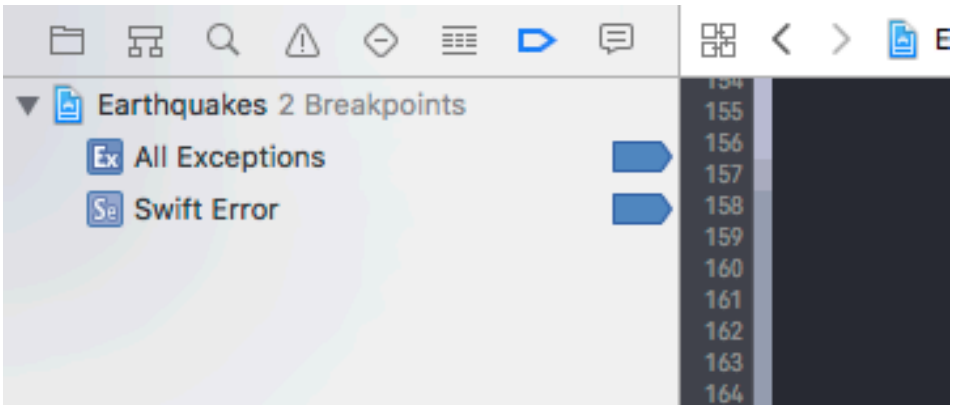
Tip: Cmd+Option+ left/right collapse/expand a method
Tip: If you look for a specific menu in Xcode, type “fold” in Help->Search

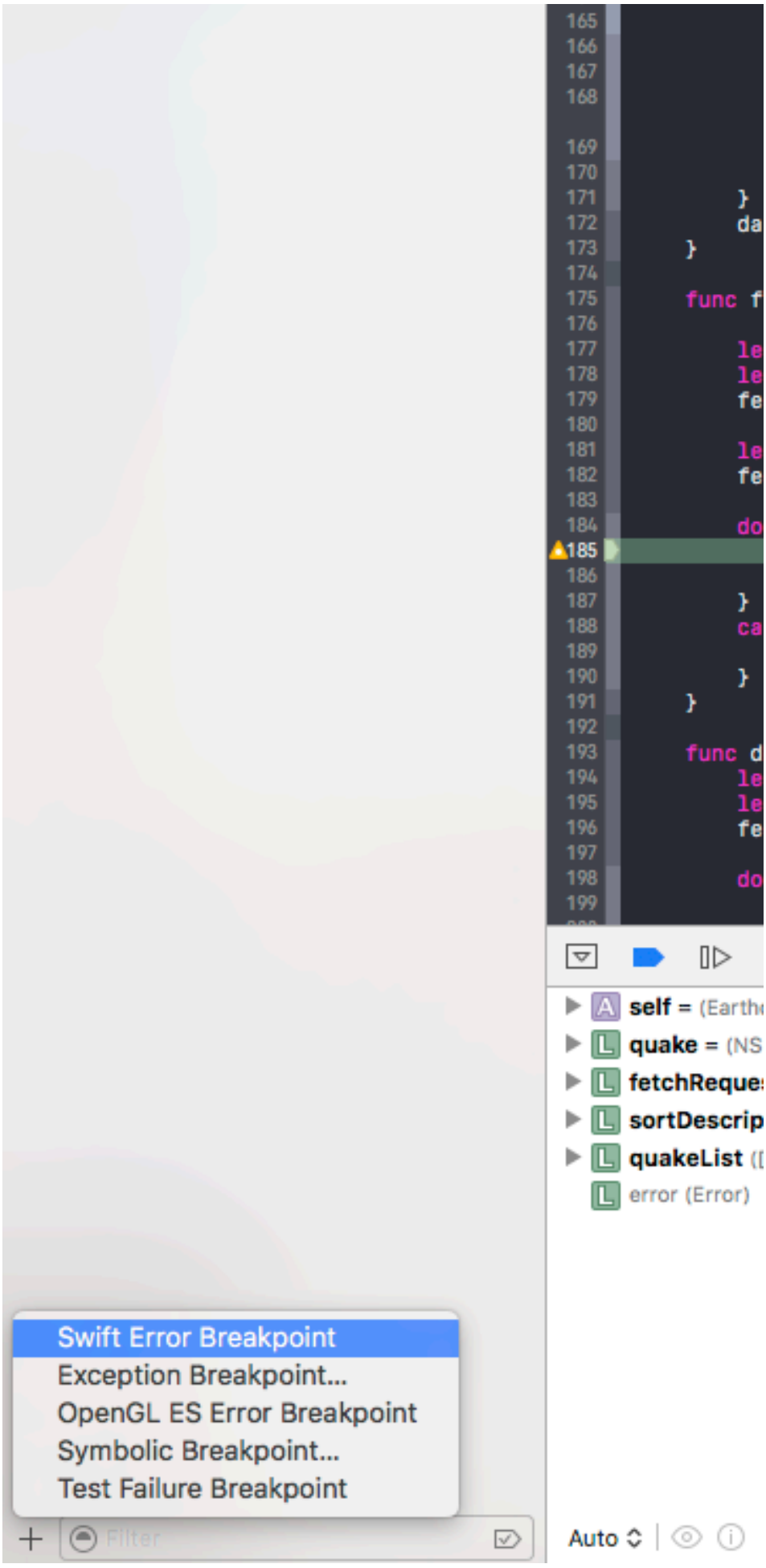


You can create classes and methods with generic type

Tip: It is a good practice to split a gigantic VC in multiple VCs. Then split each VC in multiple extensions.

Tip: when you have a crash enable this two breakpoints in the debugger perspective. Run again and you will catch the exact point of the crash




Swift Error Breakpoint
Exception Breakpoint...
OpenGL ES Error Breakpoint
Symbolic Breakpoint...
Test Failure Breakpoint
Filter
Auto
self = (Earth
quake = (NS
fetchReque
sortDescrip
quakeList ([
error (Error)

Tip: Xcode caches a lot of thing to speedup compilation. Press Build Clean to clean the cache and then rebuild the project.
Tip: in Xcode Before to Run the project, click the Stop button. Otherwise multiple execution could run

Multithreading issue in the app

It is not good to run everything in the view contextthe view context deals with the UI (it runs in the main queue)
Let’s create a global context that will run in a secondary queue.
You can create as many global context as you want.

Migration

If you change the data model after the app has been shipped to the customer, the new version of the app may be not compatible with the information already stored in the user device. The the new app will crash. Migrate to a new version of the Core Data data model is very critical.
Core Data offers three methods to do migration.
The Lightweight migration is the easiest way.

Tip: When you create the model there are some tricks to make things easy in a future migration
Tip: For better performance is better to assign a picture to a different entity