



Piscine iOS Swift - Day 08

CoreData

Maxime LEMORT mlemort@student.42.fr

PE LIEB plieb@student.42.fr

42 Staff pedago@42.fr

*Summary: This document contains the subject for Day 08 for the „Piscine iOS Swift”
from [42](#)*

Contents

I	Foreword	2
II	General Instructions	3
III	Introduction	5
IV	Exercice 00 : Pod creation	6
V	Exercice 01 : Podspec	7
VI	Exercice 02 : xcdatamodeld	8
VII	Exercice 03 : Class Article	9
VIII	Exercice 04 : Class ArticleManager	10
IX	Exercice 05 : ViewController	11

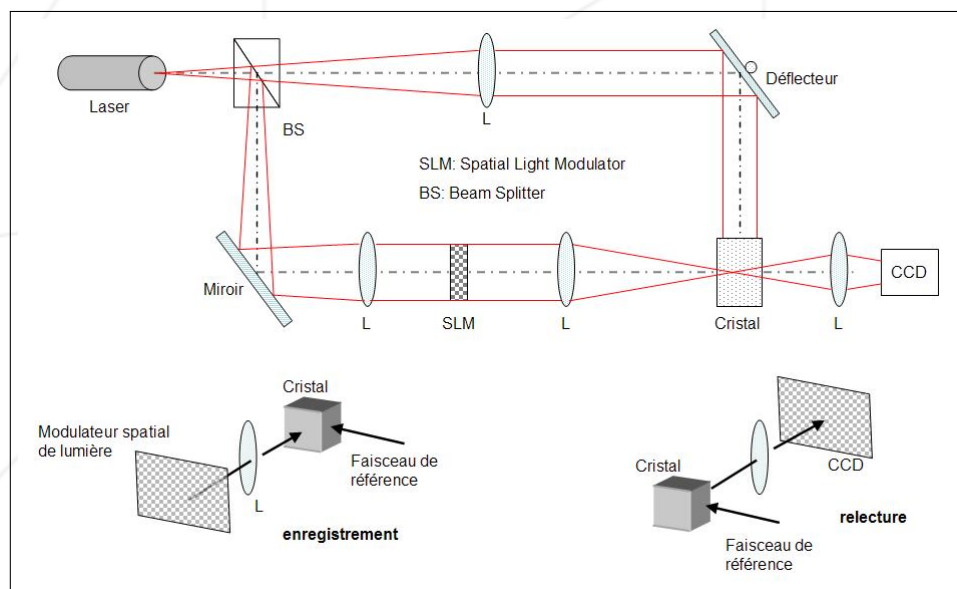
Chapter I

Foreword

Here is what the wikipedia page says about Holographic data storage:

Holographic data storage is a potential technology in the area of high-capacity data storage currently dominated by magnetic data storage and conventional optical data storage. Magnetic and optical data storage devices rely on individual bits being stored as distinct magnetic or optical changes on the surface of the recording medium. Holographic data storage records information throughout the volume of the medium and is capable of recording multiple images in the same area utilizing light at different angles.

Additionally, whereas magnetic and optical data storage records information a bit at a time in a linear fashion, holographic storage is capable of recording and reading millions of bits in parallel, enabling data transfer rates greater than those attained by traditional optical storage.



Chapter II

General Instructions

- Only this document will serve as reference. Do not trust rumors.
- Read carefully the whole subject before beginning.
- Watch out! This document could potentially change up to an hour before submission.
- This project will be corrected by humans only.
- The document can be relied upon, do not blindly trust the demos which can contain unrequired additions.
- You will have to submit one app every day (except for Day 01) on your git repository, submit the folder of the Xcode project.
- Here it is the official manual of [Swift](#) and of [Swift Standard Library](#)
- It is forbidden to use other libraries, packages, pods, etc. before Day 07
- Got a question ? Ask your peer on the right. Otherwise, try your peer on the left.
- You can discuss on the Piscine forum of your Intra !
- By Odin, by Thor ! Use your brain !!!



The videos on Intra were produced before Swift 3. Remove the prefix "NS" which you see in front of the class/struct/function in the code in the videos in order to use them in Swift 3.



Intra indicates the date and the hour of closing for your repositories. This date and hour also corresponds to the beginning of the peer-evaluation period for the corresponding piscine day. This peer-evaluation period lasts exactly 24h. After 24h passed, your missing peer grades will be completed with 0.

Chapter III

Introduction

Now that you discover about pods, let's learn how to create one. Still using [Cocoapods](#) !

If you already forgot what a pod is, consider it is a package that will be generated by a **package manager Cocoapods** here.

Today's objective will be to create a package that will use the **CoreData** framework to learn how to use data persistence and models. The aim being to create an article manager that will serve as interface for the D09.


An important point: We are talking about integrating **CoreData** inside a pod and that's the challenging part. To continue with the next day you will have to finish this one so take your time to properly finish the Day 08!

Here are the documentation you will require to finish today's project:

- [Doc Cocoapods](#)
- [CoreData](#)

Chapter IV

Exercice 00 : Pod creation


	Exercice : 00
Pod creation	
Files to turn in : .xcodeproj and all the necessary files	
Allowed functions : Aucune	
Notes : n/a	

You have to create a **Cocoapods** pod to begin with.

Go take a look on the following [Cocoapods](#) website and follow the procedure. You will find a lot of tutorials on the Internet. Your pod must be coded in **Swift**, contain an **example**, you don't need to implement **tests**. The name of your pod will be your **username followed by the current year** Faites un tour sur le site de [Cocoapods](#) et suivez la procédure. Vous trouverez aussi plein de tutoriels sur internet. Votre pod doit etre en **Swift**, contenir un **example**, vous n'avez pas besoin de **tests**. Le nom de votre pod sera votre **login suivi de l'année courante** (ex: **mlemort2016**).

Chapter V

Exercice 01 : Podspec

	Exercice : 01
Podspec	
Files to turn in : .xcodeproj and all the necessary files	
Allowed functions : Aucune	
Notes : n/a	

Now that your pod is created you have to take care about its **podspec** file.

Your **podspec** file must contain:

- A **Description**
- A **Summary**
- A **CoreData Framework**


The only thing that you don't have to do is the github project url.



`pod lib lint YOUR_POD !`

Chapter VI

Exercice 02 : xcdatamodeld


	Exercice : 02
xcdatamodeld	
Files to turn in : .xcodeproj and all the necessary files	
Allowed functions : Aucune	
Notes : n/a	

The time has come to create a data model for **CoreData** in your pod. Add a `article.xcdatamodeld` file and add:

- A Title
- A Content
- A Language
- An Image
- A Creation Date
- A Modification Date

Chapter VII

Exercice 03 : Class Article


	Exercice : 03
Class Article	
Files to turn in : .xcodeproj and all the necessary files	
Allowed functions : Swift Standard Library, CoreData	
Notes : n/a	

Create now your **Article** class that extend **NSManagedObject**.
Your class must contain the following attributes:

- A **Title** of type **String**?
- A **Content** of type **String**?
- A **Language** of type **String**?
- An **Image** of type **NSData**?
- A **Creation Date** of type **NSDate**?
- A **Modification Date** of type **NSDate**?
- An **Override of Description**

Chapter VIII

Exercise 04 : Class ArticleManager

	Exercise : 04
Class ArticleManager	
Files to turn in : .xcodeproj and all the necessary files	
Allowed functions : Swift Standard Library, CoreData	
Notes : n/a	

Now that all of this is done you will now be able to create the **ArticleManager** class. This class must contain the following methods:


- **newArticle** that allow us to create a new article and returns it.
- **getAllArticles** that returns every stored articles.
- **getArticles(withLang lang: String)** that returns every stored articles in a specific language.
- **getArticles(containString str: String)** that returns every articles containing the following string give as a parameter.
- **removeArticle(article : Article)** that removes an article.
- **save** that saves every modification.



When creating your `NSManagedObjectContext` don't forget to use `NSBundle(forClass: AnyClass)` to change the right Bundle!

Chapter IX

Exercise 05 : ViewController

	Exercise : 05
ViewController	
Files to turn in : .xcodeproj and all the necessary files	
Allowed functions : Swift Standard Library, UIKit, CoreData	
Notes : n/a	

To conclude create several articles and the **ViewDidLoad** of your **ViewController** and display them in the debug at the launch of your app.



When you launch the app several times old articles must persist.