



Piscine iOS Swift - Day 08

CoreData

Summary: This document contains the subject of Day 08 for the iOS Swift piscine of [42](#)

Contents

I	Preamble	2
II	Consignes	3
III	Introduction	4
IV	Exercise 00: Pod's creation	5
V	Exercise 01: Podspec	6
VI	Exercise 02: xcdatamodeld	7
VII	Exercise 03: Class Article	8
VIII	Exercise 04: Class ArticleManager	9
IX	Exercise 05: ViewController	10

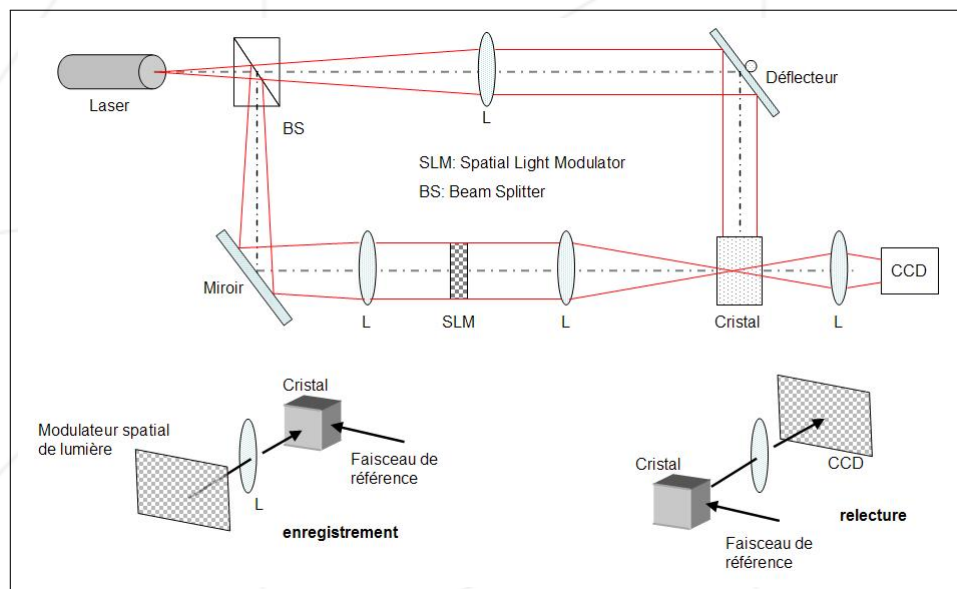
Chapter I

Preamble

Here is what Wikipedia has to say about holographic data storage:

Holographic data storage is a potential technology in the area of high-capacity data storage. While 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.

Holographic data storage contains information using an optical interference pattern within a thick, photosensitive optical material. Light from a single laser beam is divided into two, or more, separate optical patterns of dark and light pixels. By adjusting the reference beam angle, wavelength, or media position, a multitude of holograms (theoretically, several thousands) can be stored on a single volume.



Chapter II

Consignes

Sauf contradiction explicite, les consignes suivantes seront valables pour tous les jours de cette Piscine.

- Seul ce sujet sert de référence : ne vous fiez pas aux bruits de couloir.
- Le sujet peut changer jusqu'à une heure avant le rendu.
- Les exercices sont très précisément ordonnés du plus simple au plus complexe. En aucun cas nous ne porterons attention ni ne prendrons en compte un exercice complexe si un exercice plus simple n'est pas parfaitement réussi.
- Attention aux droits de vos fichiers et de vos répertoires.
- Vous devez suivre la procédure de rendu pour tous vos exercices. L'url de votre dépôt GIT pour cette journée est disponible sur votre intranet.
- Vos exercices seront évalués par vos camarades de Piscine.
- En plus de vos camarades, vous pouvez être évalués par un programme appelé la Moulinette. La Moulinette est très stricte dans sa notation car elle est totalement automatisée. Il est donc impossible de discuter de sa note avec elle. Soyez d'une rigueur irréprochable pour éviter les mauvaises surprises.
- Les exercices shell doivent s'exécuter avec `/bin/sh`.
- Vous ne devez laisser aucun autre fichier que ceux explicitement spécifiés par les énoncés des exercices dans votre dépôt de rendu.
- Vous avez une question ? Demandez à votre voisin de droite. Sinon, essayez avec votre voisin de gauche.
- Toutes les réponses à vos questions techniques se trouvent dans les `man` ou sur Internet.
- Pensez à discuter sur le forum Piscine de votre Intra et sur Slack !
- Lisez attentivement les exemples car ils peuvent vous permettre d'identifier un travail à réaliser qui n'est pas précisé dans le sujet à première vue.
- Réfléchissez. Par pitié, par Thor, par Odin !

Chapter III

Introduction

Now you have discovered pods, we're gonna learn how to create one. And unsurprisingly, you will use...à [Cocoapods](#) !

If you have short term memory problems, we'll remind you a pod is a **package** managed by a **package manager**. Here, it will be **Cocoapods**.

Today's goal will be to create one that uses the **CoreData** framework to learn how to use the data and model persistence. You will have to create a article manager that will become an interface for D09.


One important point: you're gonna have to integrate **CoreData** within a pod. This is the real challenge, today. To get to the next day, you will need to complete this one. So you should take your time to achieve this D08!

Here is the documentation you will need today:

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

Chapter IV

Exercise 00: Pod's creation


	Exercice : 00
Pod's creation	
Files to turn in : None	
Authorised functions : n/a	
Notes : n/a	

To begin, you will have to create a **Cocoapods** pod.

Go check the [Cocoapods](#) website and follow the procedure. You will also find a lot of tutorials on the net. Your pod must be in **Swift**, contain an **example**, but you don't need any **tests**. You will name your pod after your **login followed by the current year**) (ex: **mlemort2016**).

Chapter V

Exercise 01: Podspec

	Exercise : 01
Podspec	
Files to turn in : Aucune	
Authorised functions : n/a	
Notes : n/a	

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

Your **podspec** file must contain:

- A **Description**
- An **Abstract**
- A CoreData **Framework**


The only thing we will spare you is the project's URL on github.



```
pod lib lint YOUR_POD!
```

Chapter VI

Exercise 02: xcdatamodeld


	Exercise : 02
xcdatamodeld	
Files to turn in : None	
Authorised functions : n/a	
Notes : n/a	

It's time to create your data model for **CoreData** in your pod. Add a `article.xcdatamodeld` file and add to it:

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

Chapter VII

Exercise 03: Class Article


	Exercice : 03
Class Article	
Files to turn in : Swift Standard Library, CoreData	
Authorised functions : n/a	
Notes : n/a	

Now, create your **Article** class that extends from **NSManagedObject**. Your class must contain the following attributes:

- A **String?** type **Title**
- A **String?** type **Content**
- A **String?** type **Language**
- A **String?** type **Image**
- A **NSDate?** type **Creation date**
- A **NSDate?** type **Modification date**
- A **Description Override**

Chapter VIII

Exercise 04: Class ArticleManager

	Exercise : 04
Class ArticleManager	
Files to turn in : Swift Standard Library, CoreData	
Authorised functions : n/a	
Notes : n/a	

Now it's all done, we're gonna create our **ArticleManager** class. This class must contain the following methods:


- **newArticle** that helps create a new article and sends it back.
- **getAllArticles** that sends back all the stored articles.
- **getArticles(withLang lang: String)** that sends back all the stored articles with the given language.
- **getArticles(containString str: String)** that sends back all the stored articles containing the string passed in parameter.
- **removeArticle(article: Article)** that deletes the article.
- **save** that saves all the modifications.



When creating your `NSManagedObjectContext` you should use `NSBundle(forClass: AnyClass)` to load the right **Bundle**!

Chapter IX

Exercise 05: ViewController

	Exercise : 05
ViewController	
Files to turn in : Swift Standard Library, UIKit, CoreData	
Authorised functions : n/a	
Notes : n/a	

Now, create several articles in the **ViewDidLoad** of your **ViewController** and display them in the debug when launching the app.



When you launch the application several times, previous articles must persist.