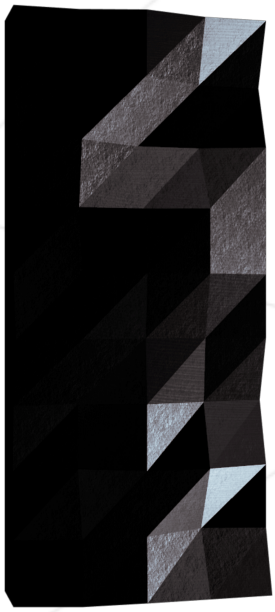


Piscine iOS Swift - Day 08

CoreData

Summary: This document contains the subject of Day 08 for the iOS Swift piscine of 42

Contents

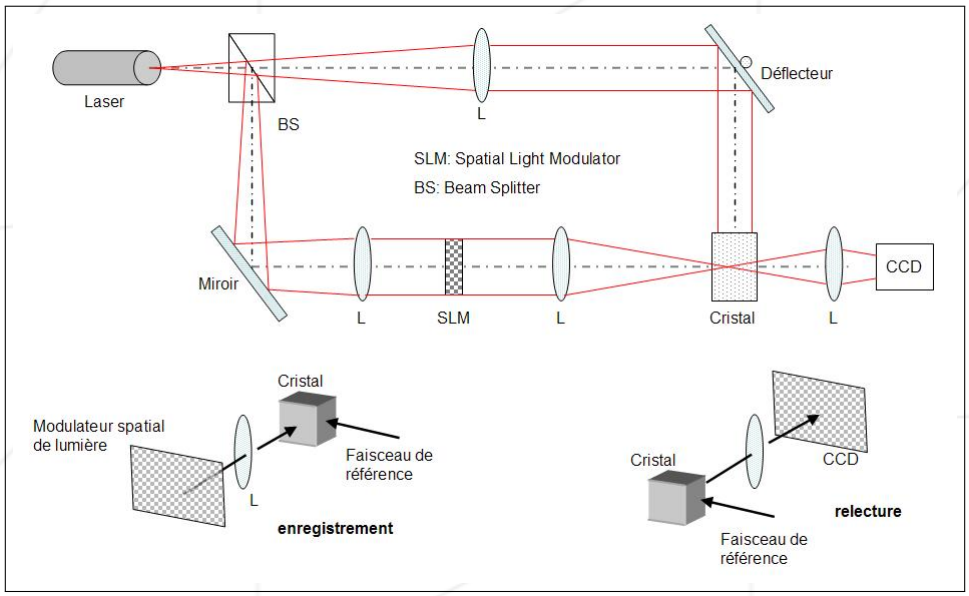
Chapter I

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

Instructions

- Only this page will serve as reference. Do not trust rumors.
- Read attentively the whole document before beginning.
- Your exercises will be corrected by your piscine colleagues.
- The document can be relied upon, do not blindly trust the demos which can contain not required additions.
- You will have to deliver an app every day (except for Day 01) on your git repository, where you deliver the file 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...before Day 07
- Got a question? Ask your peer on the right. Otherwise, try your peer on the left.
- Think about discussing on the forum Piscine of your Intra!
- 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 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 a 0.

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 on 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 tuorials 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

	Exercice : 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 it **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

	Exercice : 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 etends 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? Modification date**
- A **Description Override**

Chapter VIII

Exercise 04: Class ArticleManager

	Exercice : 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

	Exercice : 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.`