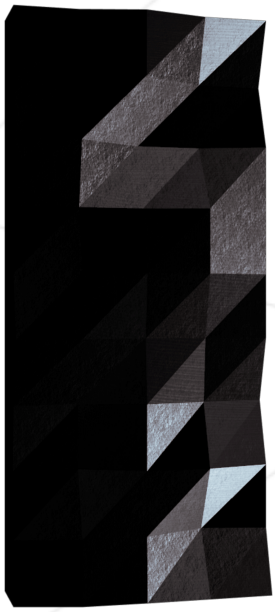

Piscine iOS Swift - Day 07

Siri

Summary: This document contains the Day 07 for the iOS Swift piscine of 42

Contents

Chapter I

Préambule

Here is what Wikipedia has to say about Artificial Intelligence:

In computer science, artificial intelligence (AI), sometimes called machine intelligence, is intelligence demonstrated by machines, in contrast to the natural intelligence displayed by humans and animals. Leading AI textbooks define the field as the study of "intelligent agents": any device that perceives its environment and takes actions that maximize its chance of successfully achieving its goals. Colloquially, the term "artificial intelligence" is often used to describe machines (or computers) that mimic "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving".

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

You probably already know that developers are mostly lazy people and there's a good reason for this: there's not need to reinvent the wheel everyday it already rolls perfectly. Today, you're going to meet the pods thanks to Cocoapods !

But what is a pod? A pod is neither nor The Breeders. A pod is a package. Thus, it is managed by a **package manager**. Here, it will be **Cocoapods**.

Today, you will learn how to use pods, that is: install them, and then use them. Today's goal will be to create a robot, more commonly known as **bot** that will cast the weather for a given city.

To do so, you will use 2 API provided by 2 different services you will have to register to.

- Recast.AI: an API will provide you the **Artificial Intelligence** brick for your bot.
- Dark Sky : an open source API that will help you recover the weather for a given latitude and longitude. (Formerly Forecast.IO)

Here is the documentation you will need:

- Doc Cocoapods
- Pod Recast.AI
- Pod Dark Sky
- Pod JSQMessagesViewController that will help you display your conversation in a messenger format.

Chapter IV

Exercise 00 : Installing Cocoapods

	Exercice : 00
Installing Cocoapods	
Files to turn in : `Swift Standard Library, UIKit, Cocoapods`	
Authorised functions : `n/a`	
Notes : `n/a`	

To begin, you're gonna have to install **Cocoapods**.

Go check the Cocoapods site and follow the procedure. Make sure Cocoapods is properly installed typing **'pod'** in the console:

```
>pod
Usage:

    $ pod COMMAND

      CocoaPods, the Cocoa library package manager.

Commands:

    + cache      Manipulate the CocoaPods cache
    + init       Generate a Podfile for the current directory.
    + install    Install project dependencies to Podfile.lock versions
    + ipc        Inter-process communication
    + lib        Develop pods
    + list       List pods
    + outdated   Show outdated project dependencies
    + plugins    Show available CocoaPods plugins
    + repo       Manage spec-repositories
    + search     Search for pods.
    + setup      Setup the CocoaPods environment
    + spec       Manage pod specs
    + trunk      Interact with the CocoaPods API (e.g. publishing new specs)
    + try        Try a Pod!
    + update     Update outdated project dependencies and create new Podfile.lock

Options:

    --silent     Show nothing
    --version    Show the version of the tool
    --verbose    Show more debugging information
    --no-ansi    Show output without ANSI codes
    --help       Show help banner of specified command
```

Chapter V

Exercise 01 : FirstViewController

	Exercice : 01
FirstViewController	
Files to turn in : `Swift Standard Library, UIKit, Cocoapods`	
Authorised functions : `n/a`	
Notes : `n/a`	

Today, you will need a controller allowing you to make requests to a `Recast` and `Dark Sky`.

Create a controller featuring:

- A **Button** to make a request to Recast.
- A **TextField** to write the text you will send.
- A **Label** to display the answer.

`Think about the 'Autolayout!`

Chapter VI

Exercise 02: 'RecastAI' pod

	Exercice : 02
pod 'RecastAI'	
Files to turn in : `Swift Standard Library, UIKit, Cocoapods`	
Authorised functions : `n/a`	
Notes : `n/a`	

It's time to create an account on Recast.AI. Install the '`RecastAI`' pod to be able to

make requests and use a Slackbot bot in the community with the weather intention then make a request with the token via the button passing **TextField** in parameter.

The **label** must display the intention returned by **Recast** or "Error" if no intention is returned.

Chapter VII

Exercise 03: 'Dark Sky' pod

	Exercice : 03
pod 'Dark Sky'	
Files to turn in : `Swift Standard Library, UIKit, Cocoapods`	
Authorised functions : `n/a`	
Notes : `n/a`	

Now, we're gonna display the weather according to the returned location. To do so, create an account on Dark Sky to get a token. Install the **'Dark Sky'** pod and make a request to Dark Sky once **Recast** request is over. The label must display the weather returned by the Dark Sky API according to the location returned by the **Recast** API.

Chapter VIII

Exercise 04: pod 'JSQMessagesViewController'

	Exercice : 04
pod 'JSQMessagesViewController'	
Files to turn in : `Swift Standard Library, UIKit, Cocoapods`	
Authorised functions : `n/a`	
Notes : `n/a`	

Now it's all done, we're gonna be able to refine our interface. To do so, we're gonna use **'JSQMessagesViewController'** pod to display our requests as a conversation with our bot.

You will also add a button to make vocal requests.