

# Formation Mobile Hybride Day 5

Cordova, Ionic, Firebase

# Agenda

- 
1. Correction travaux pratiques

---

  2. Builder sur le téléphone

---

  3. Statut des Webviews aujourd’hui

---

  4. Publication sur Apple et Android

---

  5. Questions travaux personnels

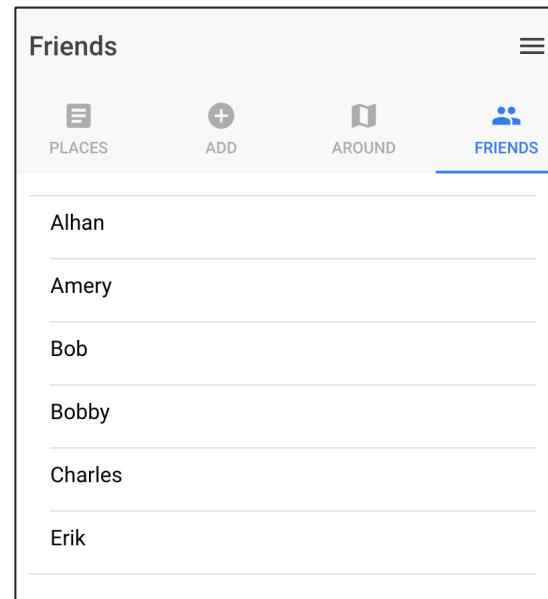
---

---

---

# Pipe

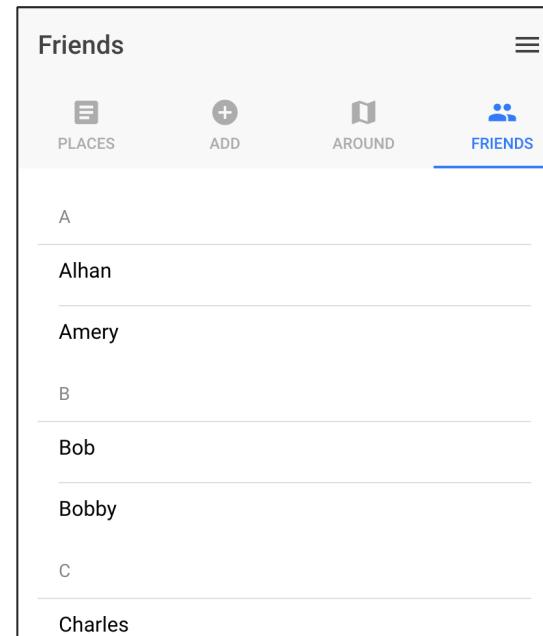
## Order friends By Name



```
@Pipe({
  name: 'sortAsc',
  pure: false,
})
@Injectable()
export class SortAsc {
  /*
    Takes an array and sort in alphabetically order
  */
  transform(value: Array<any>, property:string="name") {
    return value.sort((a, b)=>{
      if(a[property] < b[property]) return -1;
      if(a[property] > b[property]) return 1;
      return 0;
    })
  }
}
```

# Pipe

## GroupBy First Letter



```
@Pipe({
  name: 'groupBy',
  pure: false,
})
@Injectable()
export class GroupBy {
  /*
    Takes a value and makes it lowercase.
  */
  transform(value: Array<any>, property: string="name") {

    let results = []
    Observable.from(value).groupBy(
      (item:any)=> { return item[property].substring(0,1); },
      (item:any)=> { return item })
      .flatMap(group => group.toArray())
      .subscribe((data)=>{
        results.push({value:data[0][property].substring(0,1),list:data});
      })
    return results;
  }
}
```

# Builder sur le téléphone

# Environnement de développement Mobile



- Seulement sous Mac
- Xcode + **simulateur IOS**
- Debug avec **Safari**



- Sous **Windows/Mac/Linux**
- Utilisation de **Android studio**
- Debug avec chrome sur android>4.4



**Windows Phone**

- Seulement sous **Windows**
- Environnement **Visual Studio**
- **Documentation cordova obsolète**



**Android:**  
setup  
environnement de  
build et de test 1)

## 1) installer android studio

<https://developer.android.com/studio/install.html?pkg=studio>

## 2) installer le AVD et le HAXM

Android Virtual Devices allow you to run Android apps on your computer without the need for an actual Android phone or tablet. Choose Custom installation when running Android Studio for the first time. Make sure the boxes next to all of the following are checked:

- Android SDK
- Android SDK Platform
- Performance (Intel ® HAXM)
- Android Virtual Device

Then, click "Next" to install all of these components.

## 3) Install the Android 6.0 (Marshmallow) SDK

Android Studio installs the most recent Android SDK by default. Cordova, however, supports up to Android 6.0 (Marshmallow) SDK. To install it, launch the SDK Manager, click on "Configure" > "SDK Manager" in the "Welcome to Android Studio" screen.

*The SDK Manager can also be found within the Android Studio "Preferences" menu, under Appearance & Behavior → System Settings → Android SDK.*

Select the "SDK Platforms" tab from within the SDK Manager, then check the box next to "Show Package Details" in the bottom right corner. Look for and expand the Android 6.0 (Marshmallow) entry, then make sure the following items are all checked:

- Google APIs
- Android SDK Platform 23
- Intel x86 Atom\_64 System Image
- Google APIs Intel x86 Atom\_64 System Image

Next, select the "SDK Tools" tab and check the box next to "Show Package Details" here as well. Look for and expand the "Android SDK Build Tools" entry, then make sure that Android SDK Build-Tools 23.0.1 is selected.

Finally, click "Apply" to download and install the Android SDK and related build tools.

# Android

setup  
environnement de build et de test 2)

## 4) Set up the ANDROID\_HOME environment variable

The ionic/cordova command line interface requires the ANDROID\_HOME environment variable to be set up.

Add the following lines to your `~/.profile` (or equivalent) config file:

```
export ANDROID_HOME=${HOME}/Library/Android/sdk
export PATH=${PATH}: ${ANDROID_HOME}/tools
export PATH=${PATH}: ${ANDROID_HOME}/platform-tools
Type source ~/.profile to load the config into your current shell.
```

*Please make sure you export the correct path for ANDROID\_HOME.*

*reload your bash profile*  
`$ source ~/.bash_profile`

## 5) Démarrer l'outil Android Virtual Device

ouvrir android  
menu File > open the directory  
**`yourprojectPath/platforms/android`**

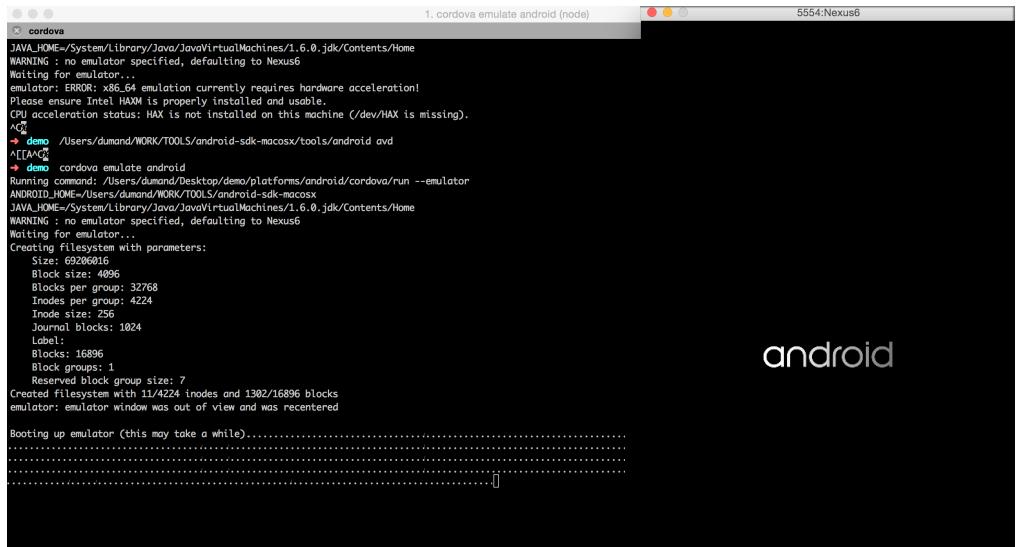
menu Tools > Android > AVD Manager

Le AVD Manager permet de créer des émulateurs android avec une config hardware paramétrable

## 6) Lancer un émulateur

Dans le AVD manager, double clicker sur une config  
L'émulateur se lance, une fois que le device android est chargé  
lancer la commande  
`$ ionic run android`  
et l'application va se lancer dans l'émulateur

# Android: emulateur



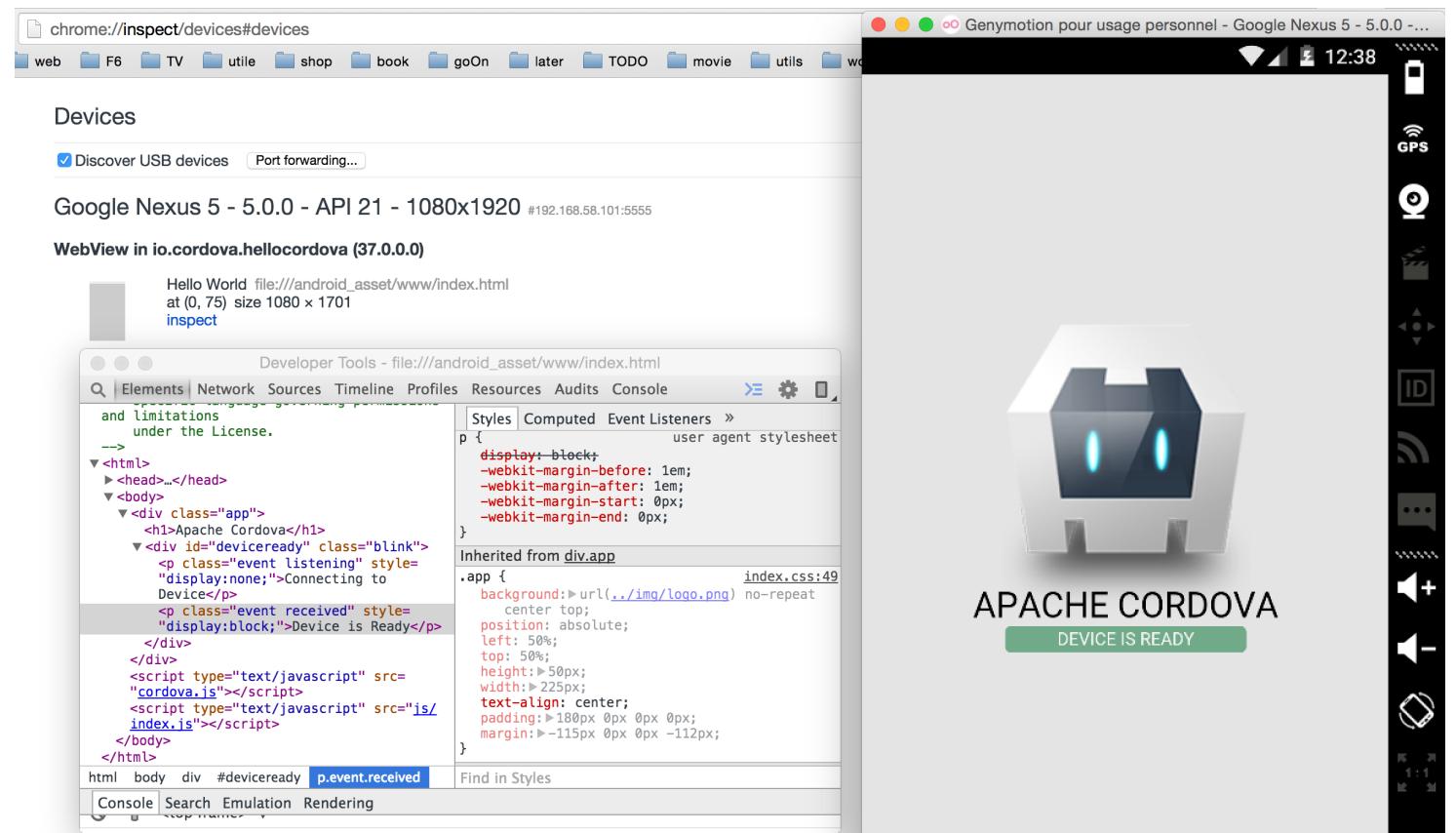
```
cordova
JAVA_HOME=/System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home
WARNING : no emulator specified, defaulting to Nexus6
Waiting for emulator...
emulator: ERROR: x86_64 emulation currently requires hardware acceleration!
Please ensure Intel HAXM is properly installed and usable.
CPU acceleration status: HAX is not installed on this machine (/dev/HAX is missing).
^C
→ demo /Users/dumond/WORK/TOOLS/android-sdk-macosx/tools/android avd
←[TAQ
→ demo cordova emulate android
Running command: /Users/dumond/Desktop/demo/platforms/android/cordova/run --emulator
ANDROID_HOME=/Users/dumond/WORK/TOOLS/android-sdk-macosx
JAVA_HOME=/System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home
WARNING : no emulator specified, defaulting to Nexus6
Waiting for emulator...
Creating filesystem with parameters:
size: 69206216
Block size: 4096
Blocks per group: 32768
Inodes per group: 4224
Inode size: 256
Journal blocks: 1024
Label:
Blocks: 16896
Block groups: 1
Reserved block group size: 7
Created filesystem with 11/4224 inodes and 1302/16896 blocks
emulator: emulator window was out of view and was recentered

Booting up emulator (this may take a while).....
```

android

# Chrome debug

- <https://docs.ionic.io/tools/developer/>





**IOS** : lancer le simulateur en ligne de commande

- Installation du simulateur

```
sudo npm install -g ios-sim
```

- Gestion des devices



To find out what are the simulator images available you can use to list them

74



```
$ ./platforms/ios/cordova/lib/list-emulator-images
iPhone-4s
iPhone-5
iPhone-5s
iPhone-6-Plus
iPhone-6
iPad-2
iPad-Retina
iPad-Air
Resizable-iPhone
Resizable-iPad
```

Then use one of the simulator names in the --target parameter:

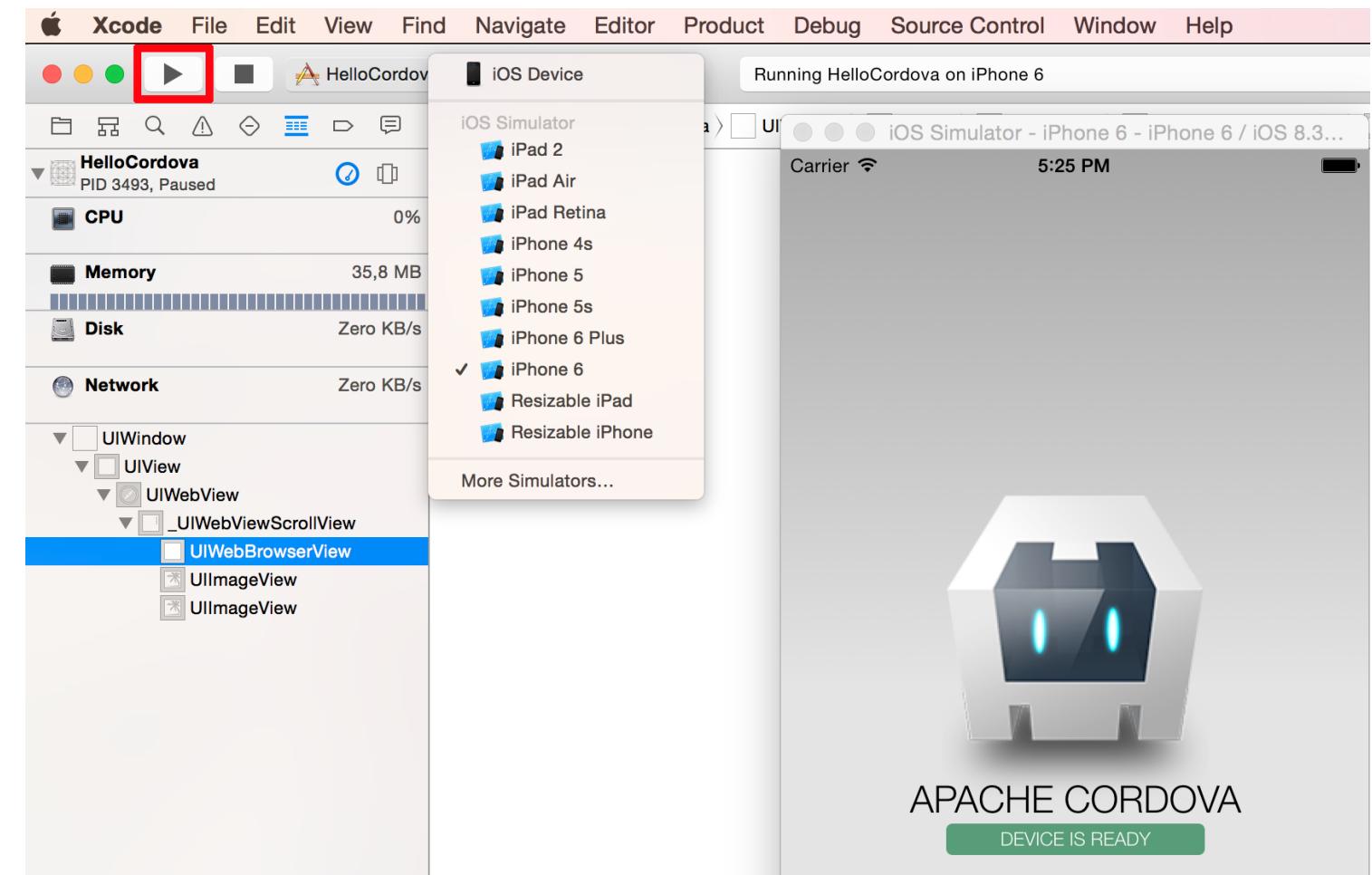
```
cordova emulate ios --target="iPhone-4s"
cordova emulate ios --target="iPad-Air"
cordova emulate ios --target="iPhone-6"
cordova emulate ios --target="iPhone-6-Plus"
```

Take into account that you may need to quit the iOS simulator via menu to switch from 3.5 to 4 inch iPhone.

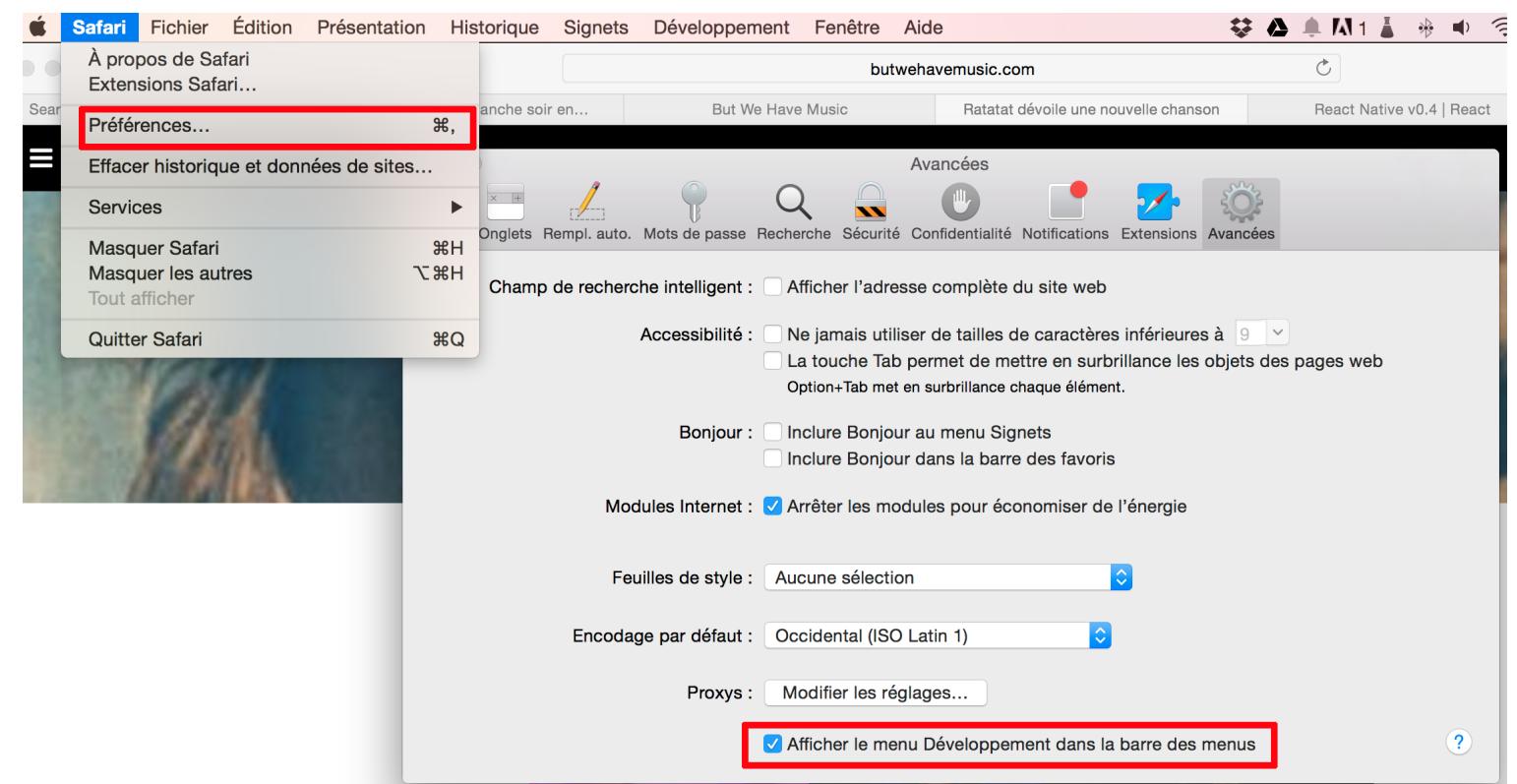
dynamic list is available in platforms/ios/cordova/lib/list-emulator-images

**IOS** : lancer le simulateur depuis Xcode

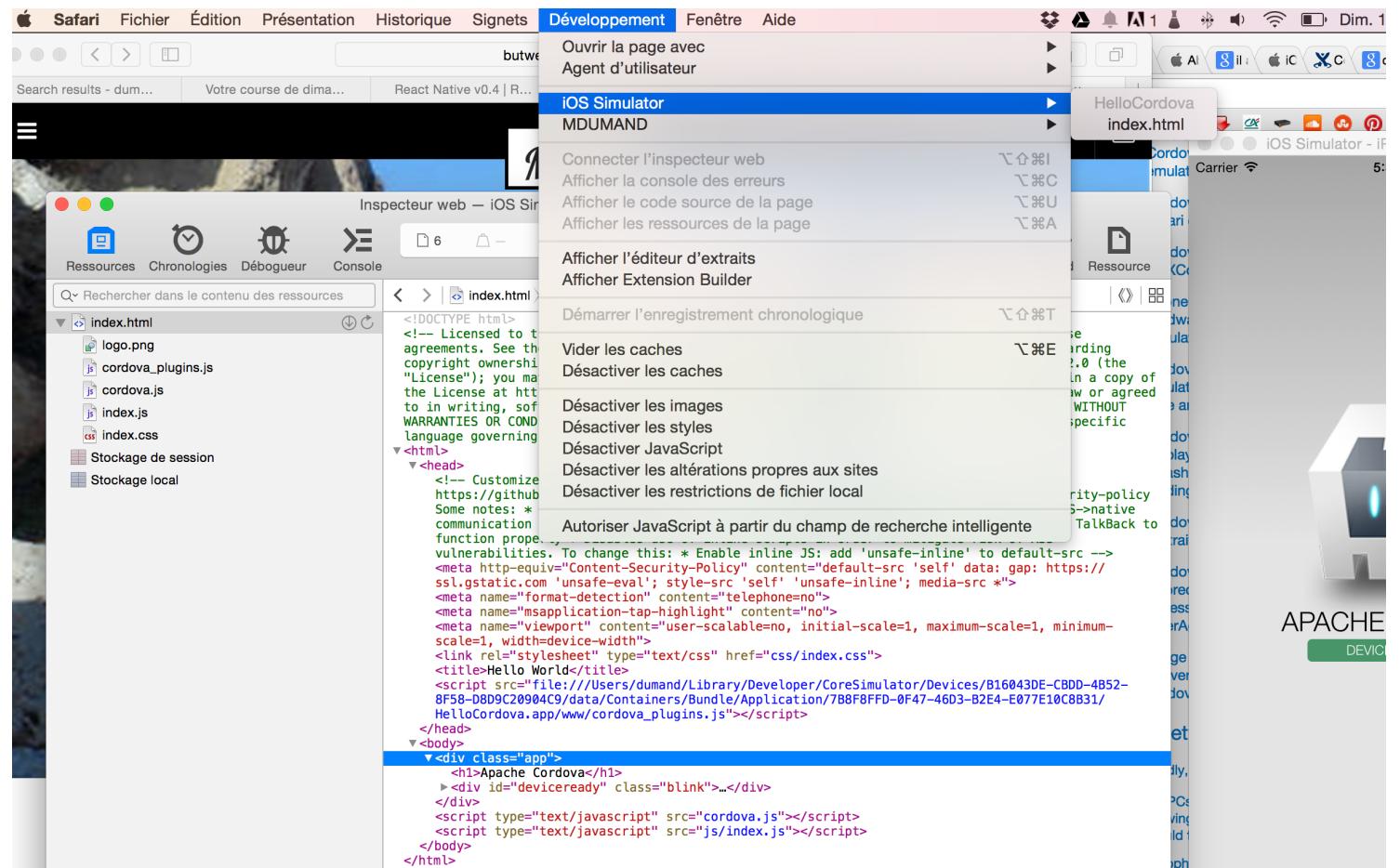
## Ouvrir platforms/ios/HelloCordova.xcodeproj



# IOS : Afficher le menu développement Safari



# IOS : Inspecter une WebView dans Safari



# Statut des WebViews aujourd’hui

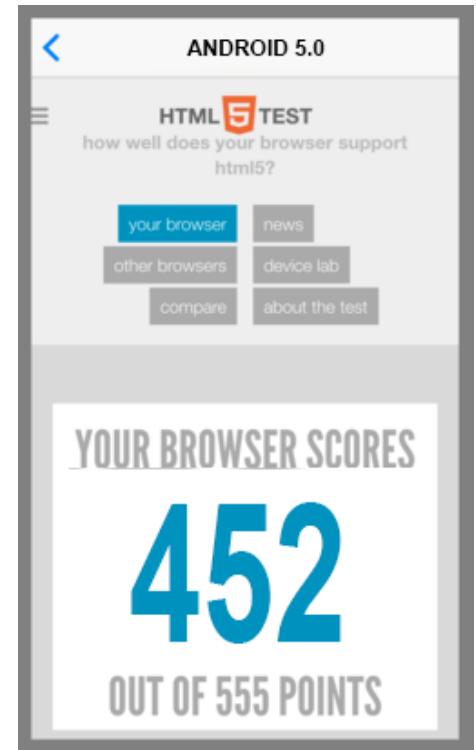
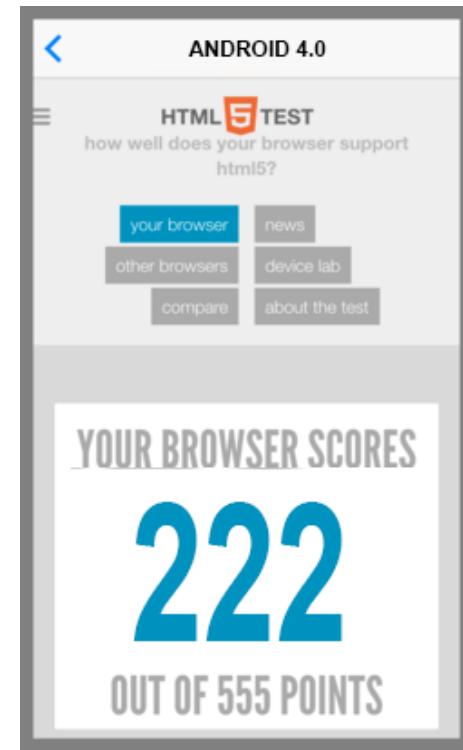
Sur IOS:  
Bon support HTML5 et  
bonne performance



The image shows a screenshot of the HTML5 Test application running within a UIWebView on an iOS device. The title bar says "UIWebView". The main content area displays the "HTML5 TEST" logo and the question "how well does your browser support html5?". Below this are several buttons: "your browser" (highlighted in blue), "other browsers", "compare", "news", "device lab", and "about the test". A large section at the bottom is titled "YOUR BROWSER SCORES" and displays a prominent blue "427". Below this, it says "OUT OF 555 POINTS".

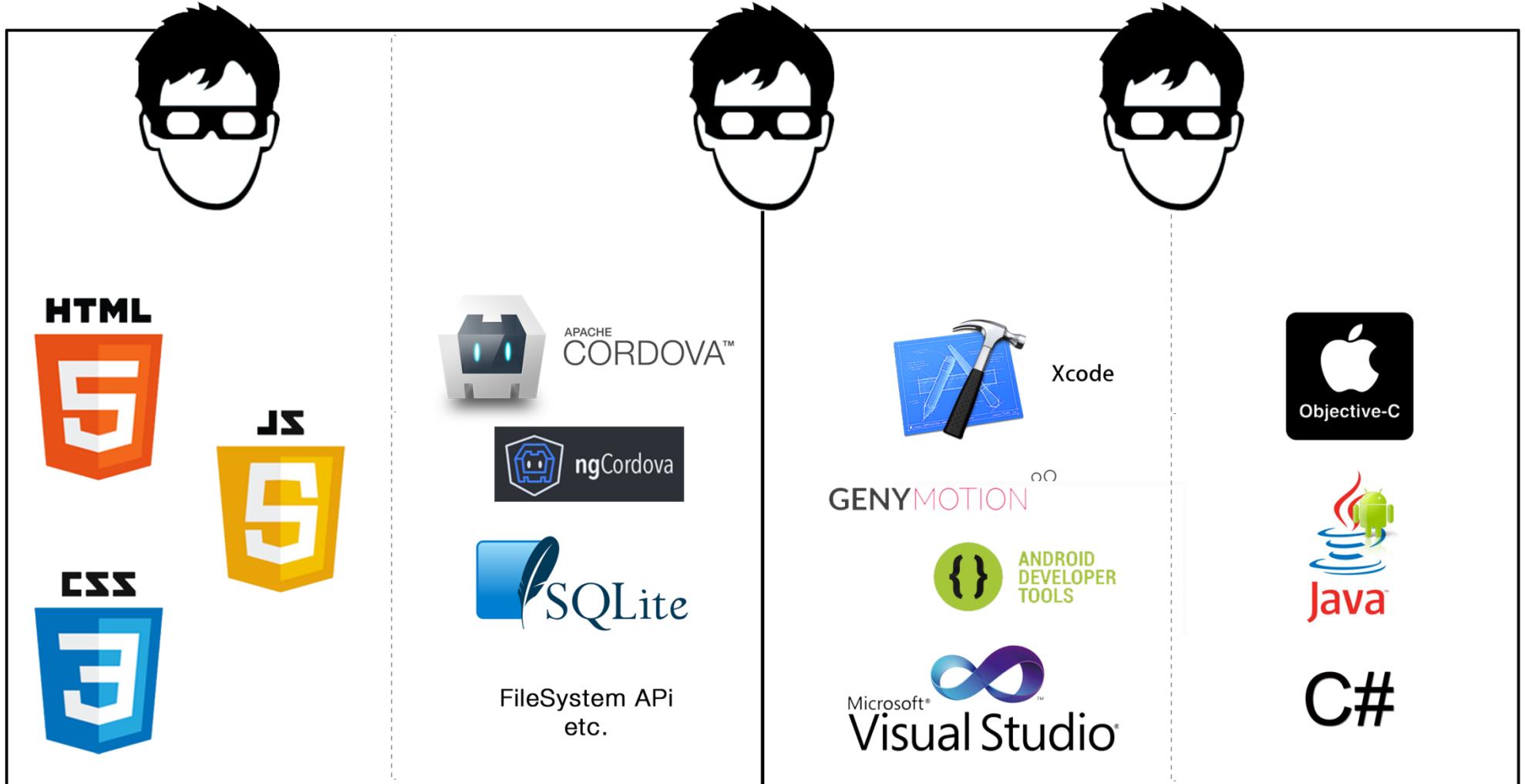
## Sur Android:

Les premières versions de WebView étaient lentes mais elles se sont bien améliorées



# Une chose est sûre !

Développeur **web** != développeur **hybride** != développeur **natif**



# Publier son application

# Préparer les assets

Prod Mode : true

```
ionicBootstrap(MyApp, [], { prodMode: true });
```

---

Générer icons et splashscreen

- **Ecraser /resources/icon.png**
  - **Ecraser /resources/splash.png**
  - Lancer \$ ionic resources
- 

Personnaliser le config.xml

- **Id : io.ionic.starter**
- **Version : 1.0.0**

# Signer une application iOS

# Signer une application IOS

## Certificats

- **.p12 = identifier le développeur**
  - **.mobileprovision = identifier l'application**
- 

Login avec votre compte apple

<https://developer.apple.com/account/>



« Enroll » IOS developer program

- <https://developer.apple.com/programs/>
- 99\$ /an pour les certificats de publication

# Enroll Program developer

## Apple Developer Program Enrollment

Profile mickael Dumand | Sign out

### Apple ID Information

The information on this Apple ID account will be used for verification and legal agreements, so please make sure your legal name and country are correct. To edit your account information, [contact us](#).

Email mickael.dumand@laposte.net  
Name mickael Dumand  
Country France

---

### Entity Type

I develop apps as  Select  Individual / Sole Proprietor / Single Person Business  Company / Organization  Government Organization

[Cancel](#) [Continue](#)

## Apple Developer Program Enrollment

Your Information mickael Dumand | Sign out

### Contact Information

The name on this Apple ID account will be used as your seller name, and for contracts and banking, if your app is selected for distribution by Apple. Make sure that it is your legal name and that your country is correct. To edit your account information, [contact us](#).

Legal Name mickael Dumand  
Phone  ext

Country Code  Phone Number  Extension

Enter your home or business address in your local language.

Address Line 1   
Address Line 2 optional   
Town / City   
Postal Code optional   
Country France

## Apple Developer Program Enrollment

Summary for Review mickael Dumand | Sign out

### Apple ID Information

The information on this Apple ID account will be used for verification and membership contracts, so please make sure your legal name and country are correct. To edit your account information, [contact us](#).

Email mickael.dumand@laposte.net  
Name mickael Dumand  
Country France

---

### Entity Type

I develop apps as a  Individual / Sole Proprietor / Single Person Business

---

### Contact Information

## Complete Your Purchase

Once you complete your purchase, you will receive a purchase acknowledgement and a membership confirmation email.

Membership Apple Developer Program  
Cost € 99  
Duration 1 year  
Enrollment ID Q72CD9HVL7

---

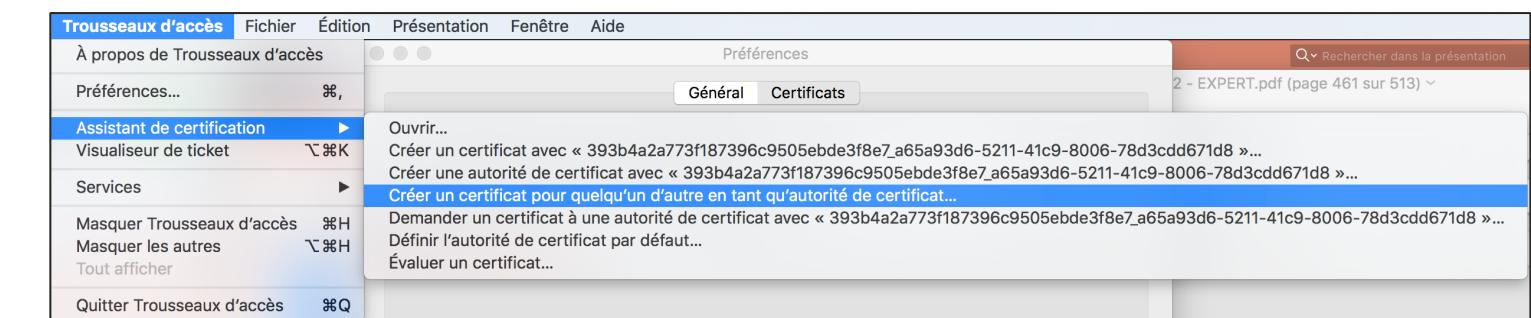
### Automatic Renewal

Automatically renewing your annual membership ensures that your apps remain available and that you maintain access to membership benefits. You can change this setting any time in your [account](#).

By checking this box and clicking the Submit button, I give permission to Apple to automatically charge the default credit/debit card associated with my Apple ID to renew my membership, subject to the following:

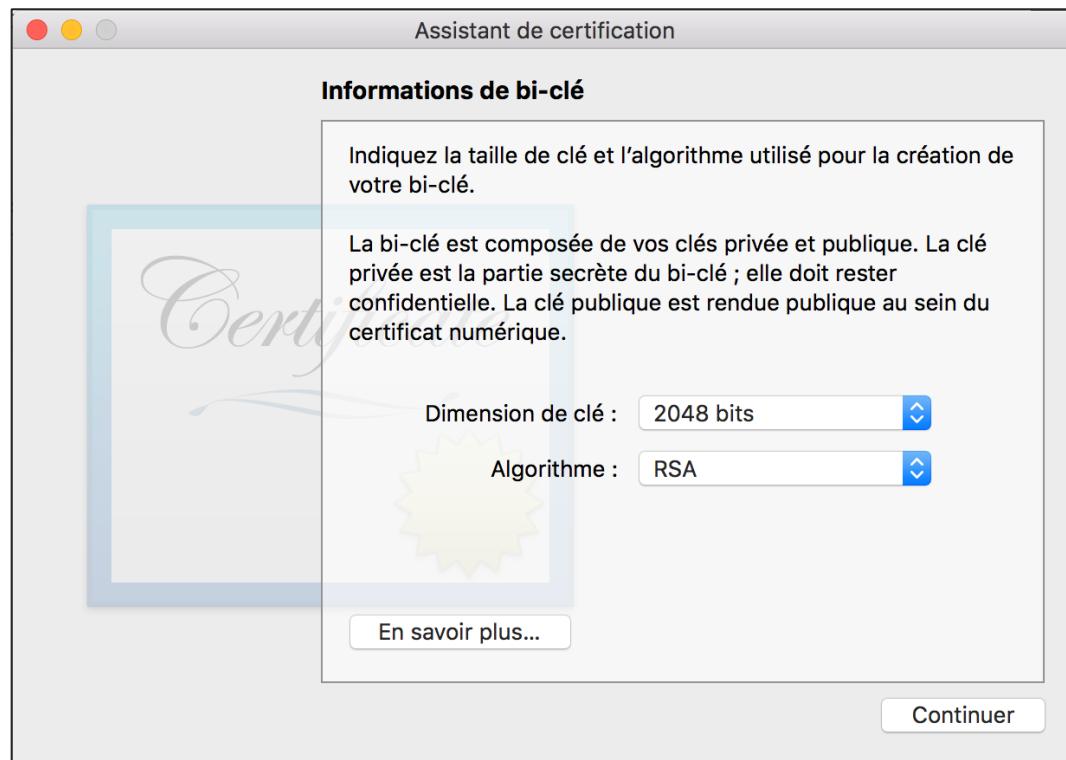
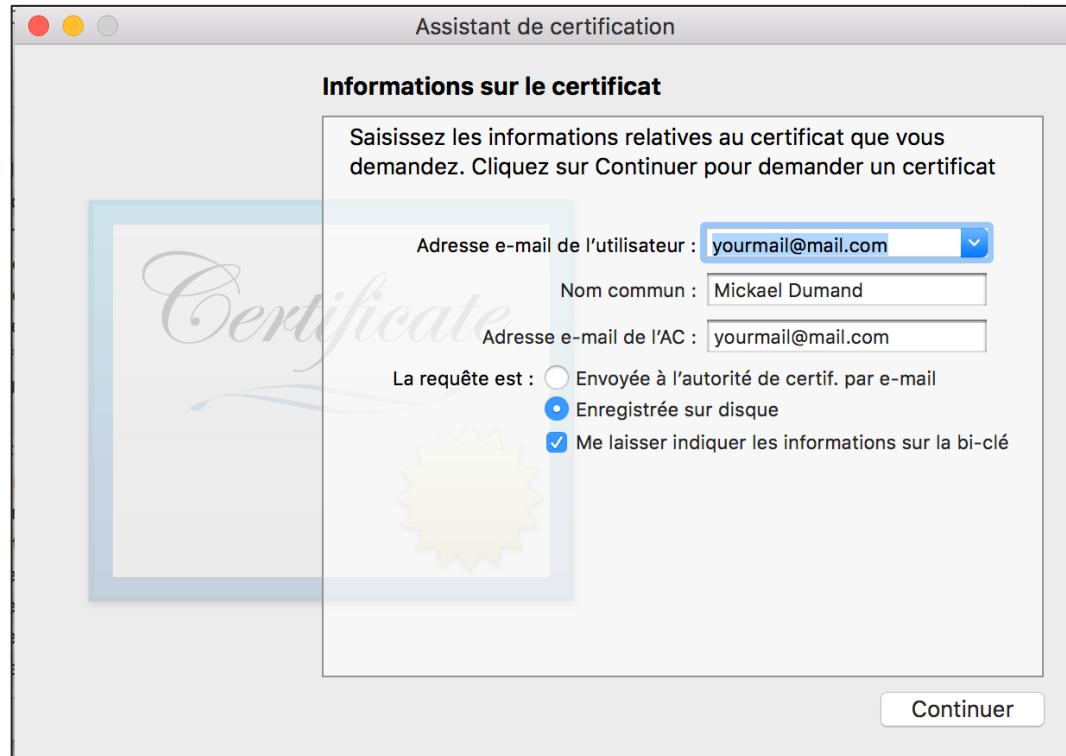
- I will be billed on an annual basis for € 99. Taxes may apply.
- I can cancel automatic renewal at any time up to 24 hours before my renewal date by unchecking the "Auto-Renew Membership" box in the Membership section of your account.
- Apple will notify me of the upcoming charge before processing my automatic renewal.
- If Apple changes the membership price, I will be notified of the new price before I am charged and can opt-out by unchecking the "Auto-Renew Membership" box in the Membership section of your account.
- If the payment for my membership cannot be transacted for any reason, Apple will not process my automatic renewal and will notify me with instructions on how to manually renew my membership.

# Le certificat développeur .p12



## Trousseau d'accès (Keychain access)

# Le certificat développeur .p12



# Le certificat développeur .p12

Trousseaux d'accès

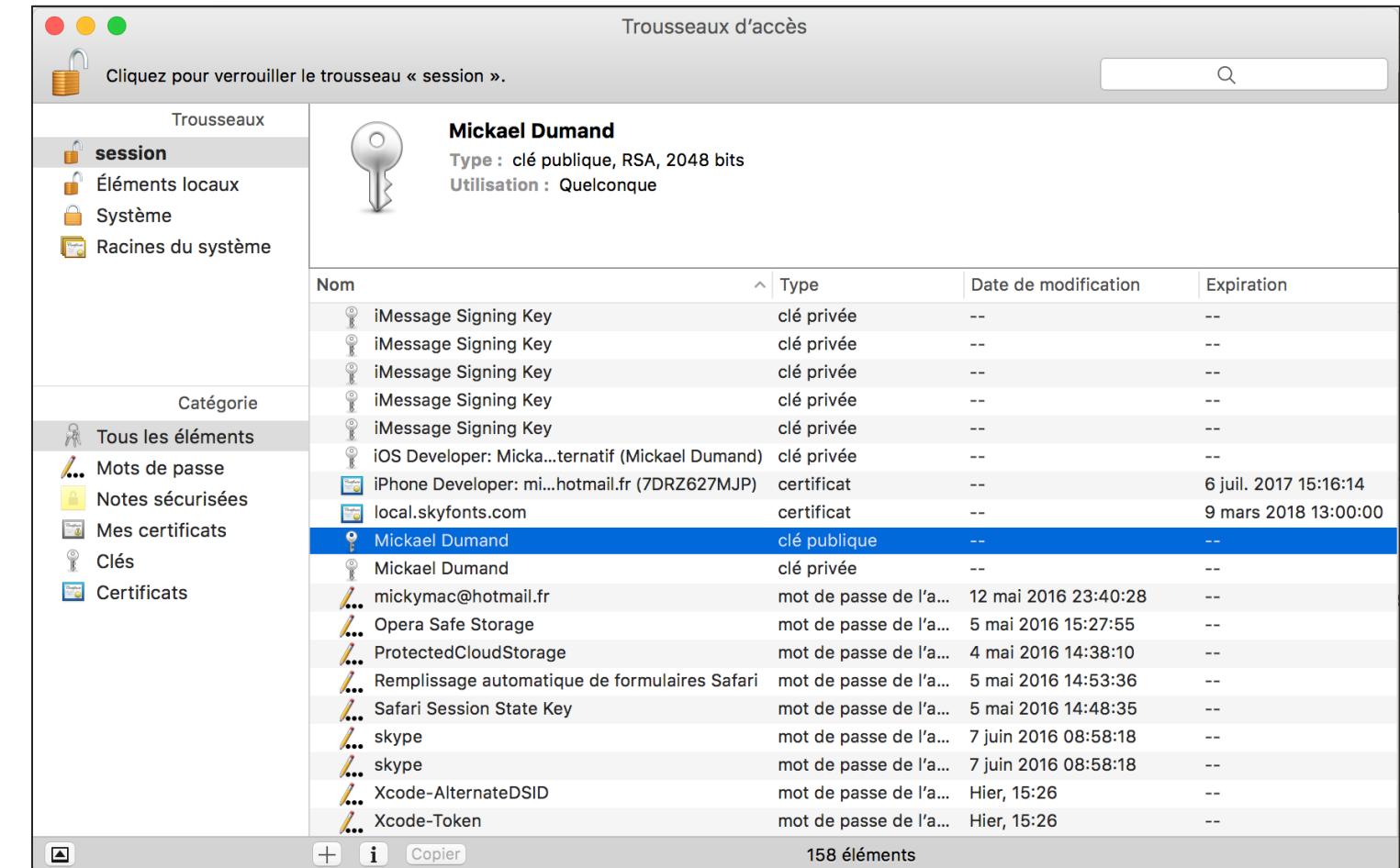
Cliquez pour verrouiller le trousseau « session ».

Mickael Dumand

Type : clé publique, RSA, 2048 bits  
Utilisation : Quelconque

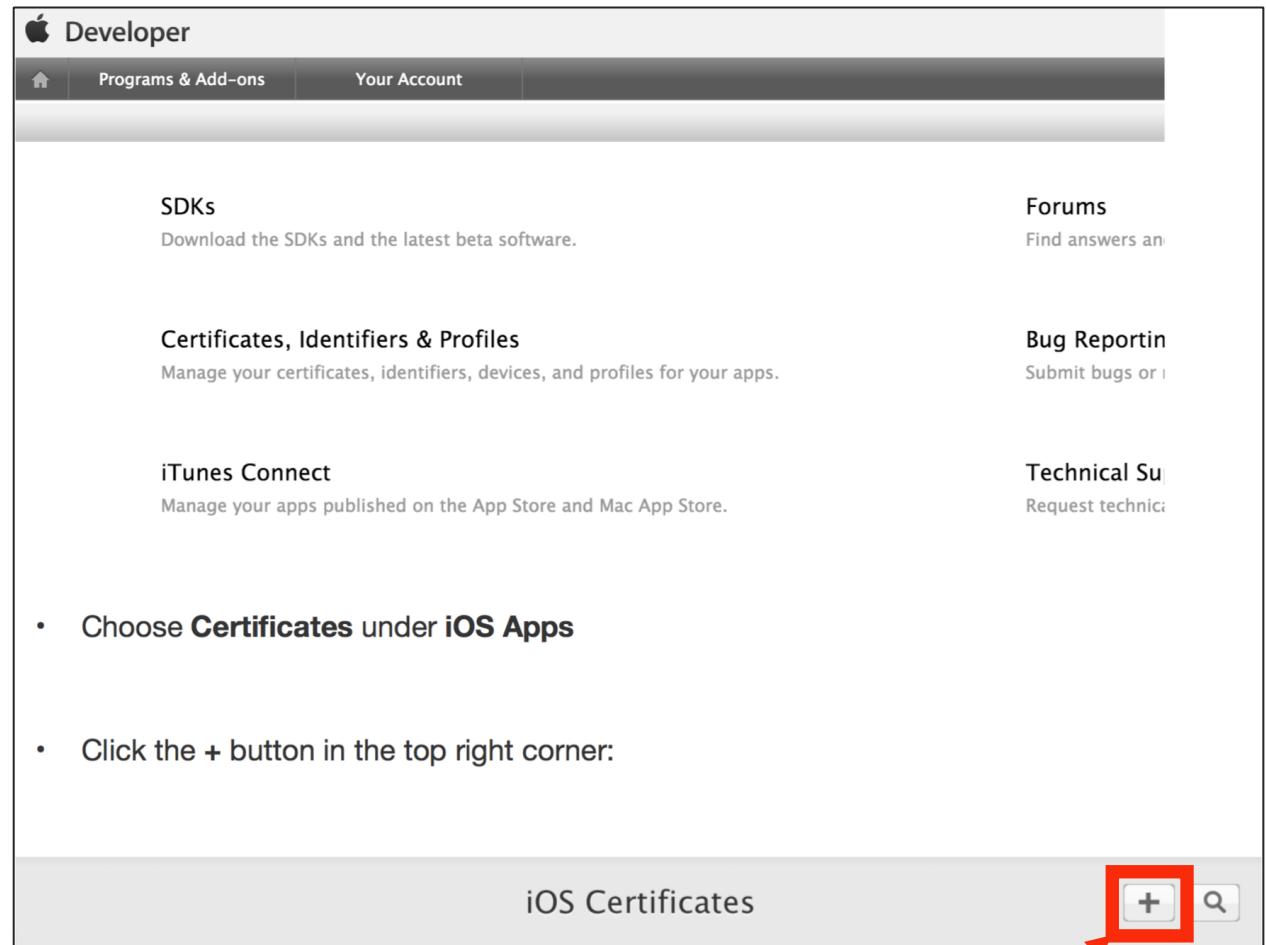
Nom	Type	Date de modification	Expiration
iMessage Signing Key	clé privée	--	--
iMessage Signing Key	clé privée	--	--
iMessage Signing Key	clé privée	--	--
iMessage Signing Key	clé privée	--	--
iMessage Signing Key	clé privée	--	--
iOS Developer: Micka...ternatif (Mickael Dumand)	clé privée	--	--
iPhone Developer: mi...hotmail.fr (7DRZ627MJP)	certificat	--	6 juil. 2017 15:16:14
local.skyfonts.com	certificat	--	9 mars 2018 13:00:00
<b>Mickael Dumand</b>	<b>clé publique</b>	--	--
Mickael Dumand	clé privée	--	--
mickymac@hotmail.fr	mot de passe de l'a...	12 mai 2016 23:40:28	--
Opera Safe Storage	mot de passe de l'a...	5 mai 2016 15:27:55	--
ProtectedCloudStorage	mot de passe de l'a...	4 mai 2016 14:38:10	--
Remplissage automatique de formulaires Safari	mot de passe de l'a...	5 mai 2016 14:53:36	--
Safari Session State Key	mot de passe de l'a...	5 mai 2016 14:48:35	--
skype	mot de passe de l'a...	7 juin 2016 08:58:18	--
skype	mot de passe de l'a...	7 juin 2016 08:58:18	--
Xcode-AlternateDSID	mot de passe de l'a...	Hier, 15:26	--
Xcode-Token	mot de passe de l'a...	Hier, 15:26	--

158 éléments



# Le certificat développeur .p12

<https://developer.apple.com/account/>



The screenshot shows the Apple Developer dashboard. At the top, there is a navigation bar with links for "Home", "Programs & Add-ons", and "Your Account". Below the navigation bar, there are several sections:

- SDKs**: A section for downloading software development kits.
- Certificates, Identifiers & Profiles**: A section for managing certificates, identifiers, devices, and profiles for apps. It includes a sub-section titled "iTunes Connect" for managing published apps.
- Forums**: A section for finding answers and submitting bugs.
- Bug Reporting**: A section for submitting bugs or issues.
- Technical Support**: A section for requesting technical support.

In the center of the dashboard, there is a list of steps:

- Choose **Certificates** under **iOS Apps**
- Click the **+** button in the top right corner:

At the bottom of the dashboard, there is a footer bar with the text "iOS Certificates" and a red arrow pointing to the "+" button in the top right corner of the footer bar.

# Les certificats

Select Type   Request   Generate   Download



**What type of certificate do you need?**

---

**Development**

**iOS App Development**  
Sign development versions of your iOS app.

**Apple Push Notification service SSL (Sandbox)**  
Establish connectivity between your notification server and the Apple Push Notification service sandbox environment. A separate certificate is required for each app you develop.

---

**Production**

**App Store and Ad Hoc**  
Sign your iOS app for submission to the App Store or for Ad Hoc distribution.

- It will now ask you for the certificate signing request you just created with Keychain Access, click continue and then upload the signing request. Once you have selected the .certSigningRequest file click Generate.

You will now be able to download your certificate.  
Download it to somewhere safe and then open it to install it.

# Create an Identifier

If you're managing your app using XCode then you don't necessarily need to worry about these step as XCode can use a Wildcard App ID, but you can also create your own App ID

manually (the Bundle ID created here will match what is in your config.xml file). Here's how you do it:

- Click on App IDs and then click the + icon

The screenshot shows the 'iOS App IDs' section of the Xcode Provisioning Profiles configuration. At the top, there are two buttons: a '+' button for adding new App IDs and a search icon. Below the title, a bullet point instructs the user to 'Fill in the App ID Description and then supply an Explicit App ID like the following:'. A section titled 'App ID Suffix' contains two options: 'Explicit App ID' (selected) and 'Wildcard App ID'. The 'Explicit App ID' section includes a note about using it for services like Game Center and In-App Purchase, and a field where 'com.example.yourproject' is entered. A note below the field recommends using a reverse-domain name style string. The 'Wildcard App ID' section includes a note about matching multiple apps and a field where an asterisk (\*) is entered.

iOS App IDs

- Fill in the App ID Description and then supply an Explicit App ID like the following:

App ID Suffix

**Explicit App ID**

If you plan to incorporate app services such as Game Center, In-App Purchase, Data Protection, and iCloud, or want a provisioning profile unique to a single app, you must register an explicit App ID for your app.

To create an explicit App ID, enter a unique string in the Bundle ID field. This string should match the Bundle ID of your app.

Bundle ID: com.example.yourproject

We recommend using a reverse-domain name style string (i.e., com.domainname.appname). It cannot contain an asterisk (\*).

**Wildcard App ID**

This allows you to use a single App ID to match multiple apps. To create a wildcard App ID, enter an asterisk (\*) as the last digit in the Bundle ID field.

Bundle ID: \*

Example: com.domainname.\*

# Provisioning Profile

- Click on Devices and then click the + button in the top right:



- Supply the name and UDID (you can find this for your device in iTunes) of the device you want to register and then click Continue. Follow the prompts to finish.

- Go to the provisioning profiles screen and click the + icon in the top right



- Choose **iOS App Development** if you are creating a provisioning profile for testing, or **App Store** if you are creating a provisioning profile for distribution. Click **Continue**.



**Select App ID.**



**Select certificates.**

# Provisioning Profile



**Name this profile and generate.**

The name you provide will be used to identify the profile in the portal.

Profile Name:

Type: **iOS Development**

App ID: **Test App(9YABKUX5J7.com**

Certificates: **1 Included**

Devices: **11 Included**

You should now be able to download your Provisioning Profile.

## Générer le fichier .p12

- Highlight both the private key and the certificate, right click and choose Export 2 items...:
  - You will then be asked where you would like to save the file and you may choose to enter a password for the .p12 file or leave it blank. You may also be asked to enter in the Admin password for your computer.
  - Congrats! You now have your .p12 file.

# Signer une application Android

# Generate a keystore

Run the following command to generate a keystore file with an alias of alias\_name (you should change this)

```
keytool -genkey -v -keystore my-release-key.keystore -alias alias_name -keyalg RSA -keysize 2048 -validity 10000
```

```
→ ~ keytool -genkey -v -keystore my-release-key.keystore -alias alias_mickymac -keyalg RSA -keysize 2048 -validity 10000
Entrez le mot de passe du fichier de clés :
Ressaisissez le nouveau mot de passe :
Quels sont vos nom et prénom ?
[Unknown]: Mickael Dumand
Quel est le nom de votre unité organisationnelle ?
[Unknown]:
Quel est le nom de votre entreprise ?
[Unknown]:
Quel est le nom de votre ville de résidence ?
[Unknown]:
Quel est le nom de votre état ou province ?
[Unknown]:
Quel est le code pays à deux lettres pour cette unité ?
[Unknown]:
Est-ce CN=Mickael Dumand, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown ?
[non]: oui

Génération d'une paire de clés RSA de 2 048 bits et d'un certificat auto-signé (SHA256withRSA) d'une validité de 10 000 jours
pour : CN=Mickael Dumand, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown
Entrez le mot de passe de la clé pour <alias_mickymac>
    (appuyez sur Entrée s'il s'agit du mot de passe du fichier de clés) :
Ressaisissez le nouveau mot de passe :
[Stockage de my-release-key.keystore]
→ ~ ls
AndroidStudioProjects  Downloads          Library        Public
Applications           Dropbox            Movies         VirtualBox VMs
Desktop                Google Drive       Music          my-release-key.keystore
Documents              InVision          Pictures       work
```

**IMPORTANT:** In order to update your application on the app store later you will need both the keystore file, along with the alias name and password.

If you lose any of these you will not be able to update the application.

## Generating a Key Hash

If you are using Facebook functionality that requires creating a Facebook application then you will need to create a Key Hash from your keystore file to supply to Facebook for Android.

```
keytool -exportcert -alias alias_name -keystore my-release-key.keystore | openssl sha1 -binary |  
openssl base64
```

Make sure to replace **alias\_name** with your own keystore files alias name and **my-release-key.keystore** with the path to your keystore file.

Once you have done this your Key Hash will be output to the terminal, And then you can simply copy it over to the Android platform settings in your Facebook application.

# Soumettre à l'Apple store

# App Store Review Guidelines

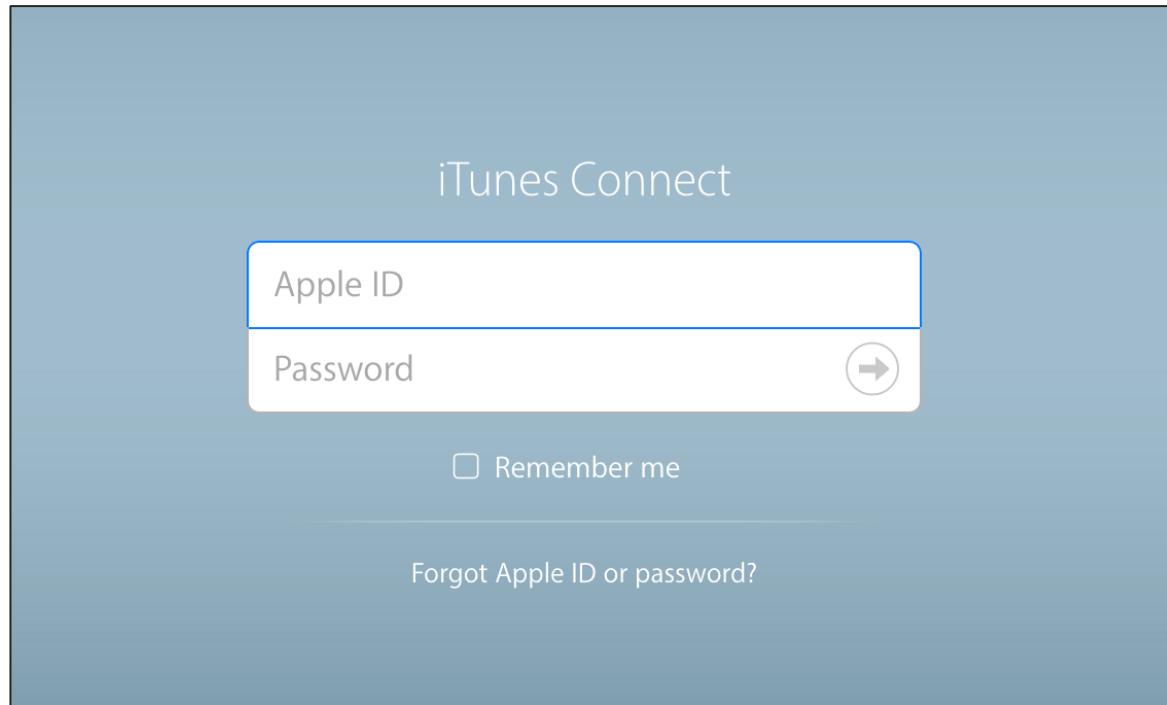
## Introduction

Apps are changing the world, enriching people's lives, and enabling developers like you to innovate like never before. As a result, the App Store has grown into an exciting and vibrant ecosystem for millions of developers and more than a billion users. Whether you are a first time developer or a large team of experienced programmers, we are excited that you are creating apps for the App Store and want to help you understand our guidelines so you can be confident your app will get through the review process quickly.

The guiding principle of the App Store is simple - we want to provide a safe experience for users to get apps and a great opportunity for all developers to be successful. We have updated the App Review Guidelines with that principle in mind. The guidelines themselves haven't changed, but they are better organized and provide more context. On the following pages you will find guidelines arranged into five clear sections: Safety, Performance, Business, Design, and Legal. A few other points to keep in mind:

- We have lots of kids downloading lots of apps. Parental controls work great to protect kids, but you have to do your part too. So know that we're keeping an eye out for the kids.
- If your app looks like it was cobbled together in a few days, or you're trying to get your first practice app into the store to impress your friends, please brace yourself for rejection. We have lots of serious developers who don't want their quality apps to be surrounded by amateur hour.

# Creating an App Store Listing



Login to [itunesconnect.apple.com](https://itunesconnect.apple.com)

The image shows the "New iOS App" setup screen in the iTunes Connect developer portal. It contains several input fields: "Name" (Snapaday), "Version" (1.0), "Primary Language" (Australian English), "SKU" (SD1), "Bundle ID" (Xcode iOS Wildcard App ID - \*), and "Bundle ID Suffix" (com.yourname.yourproject). A note at the bottom left says "Register a new bundle ID on the [Developer Portal](#)". At the bottom right are "Cancel" and "Create" buttons. A small page number "<#>" is located in the bottom right corner of the slide.

# Creating an App Store Listing

The screenshot shows the 'My Apps' section of iTunes Connect. A new app, 'Snapaday', is listed under 'iOS'. The status is '1.0 Prepare for Submission'. Navigation tabs include 'Versions', 'Prerelease', 'Pricing', 'In-App Purchases', 'Game Center', 'Reviews', 'Newsstand', and 'More'. A 'Save' button and a 'Submit for Review' button are at the top right. Below, the 'Version Information' section has tabs for 'App Video Preview and Screenshots' (selected), '4.7-Inch' (highlighted), and other device sizes. A large area for dragging screenshots is shown with the placeholder text 'Drag app video preview and up to 5 screenshots here.'.

Once you upload your application you should be able to see it in the **Build** section in iTunes Connect and will be able to attach it to your app store listing.

This dialog box is titled 'Snapaday - Rights and Pricing'. It asks to 'Select the availability date and price tier for your app.' Fields include 'Availability Date' (set to 06/Jun, 30, 2015) and 'Price Tier' (set to 'Free'). Other options include 'Discount for Educational Institutions' (checked) and 'Custom B2B App' (unchecked). A note states: 'Unless you select specific territories, your app will be available for sale in all App Store Volume Purchase Program territories.' A disclaimer for paid apps follows. At the bottom are 'Manage iCloud download settings for this app', 'Cancel', and 'Save' buttons.

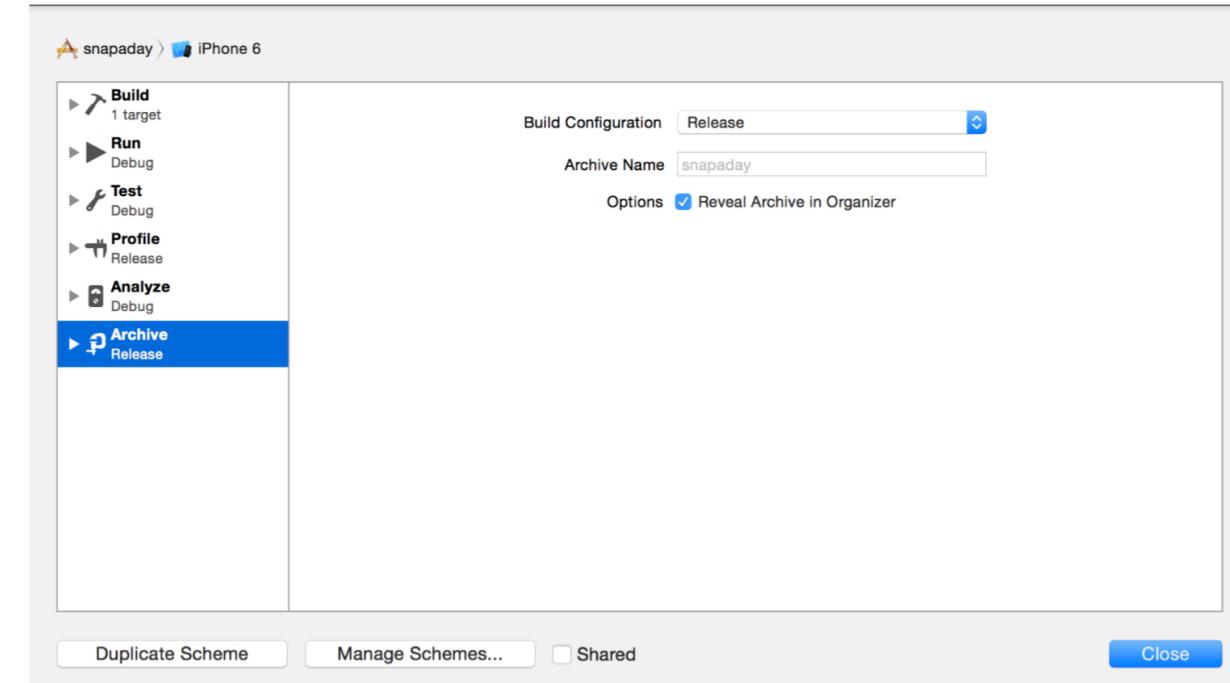
# Submitting an app using XCode

```
ionic build ios
```

## Converting a .xcodeproj file to .xcarchive

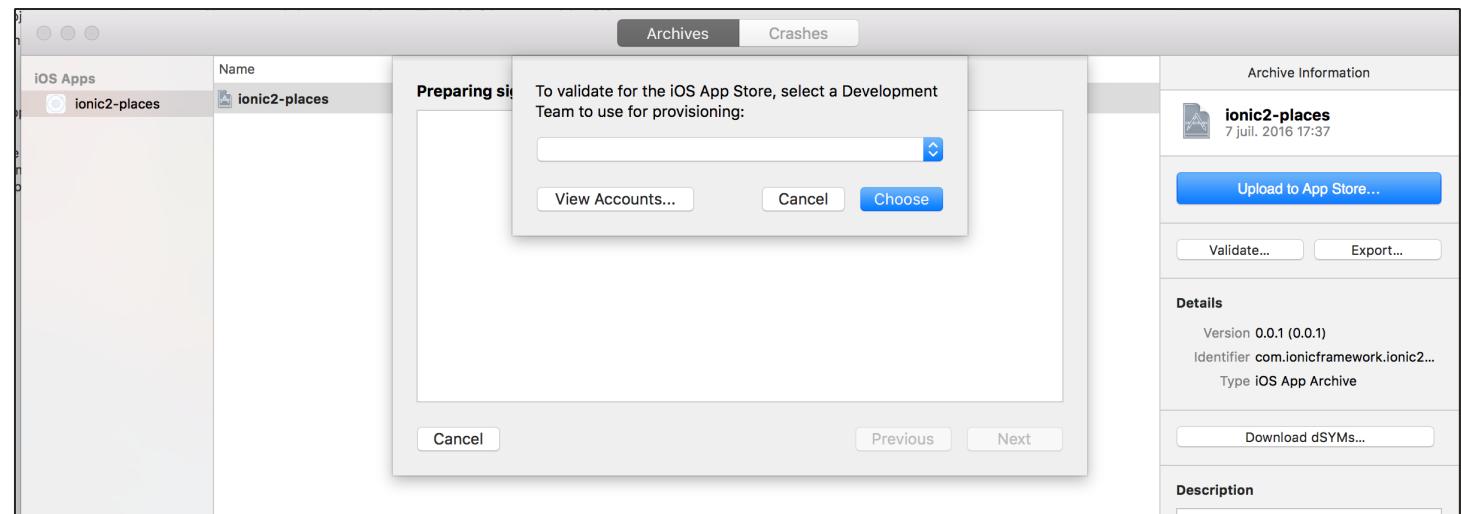
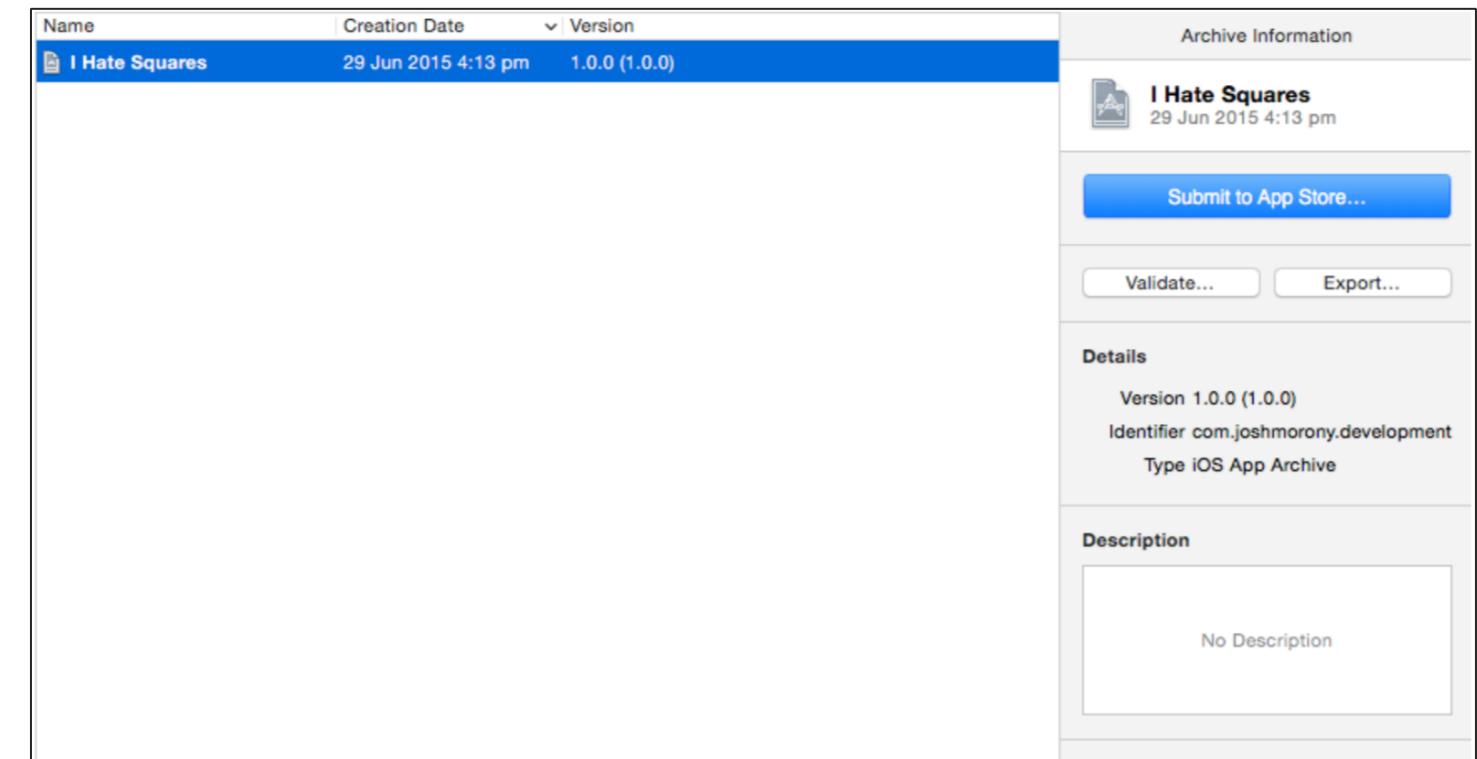
If you have a **.xcodeproj** file (which will be generated when you run the build command) you will first need to generate a **.xcarchive** file from it. To do that then you will need to follow these steps:

- Open your **.xcodeproj** file (located in **platforms/ios/snapaday.xcodeproj**) in XCode by double clicking it
- Go to **Product > Scheme > Edit Scheme** and make sure that the archive is set to a **Release** configuration:

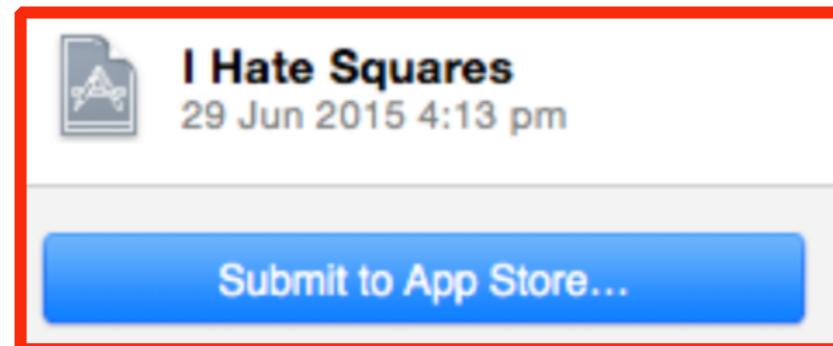
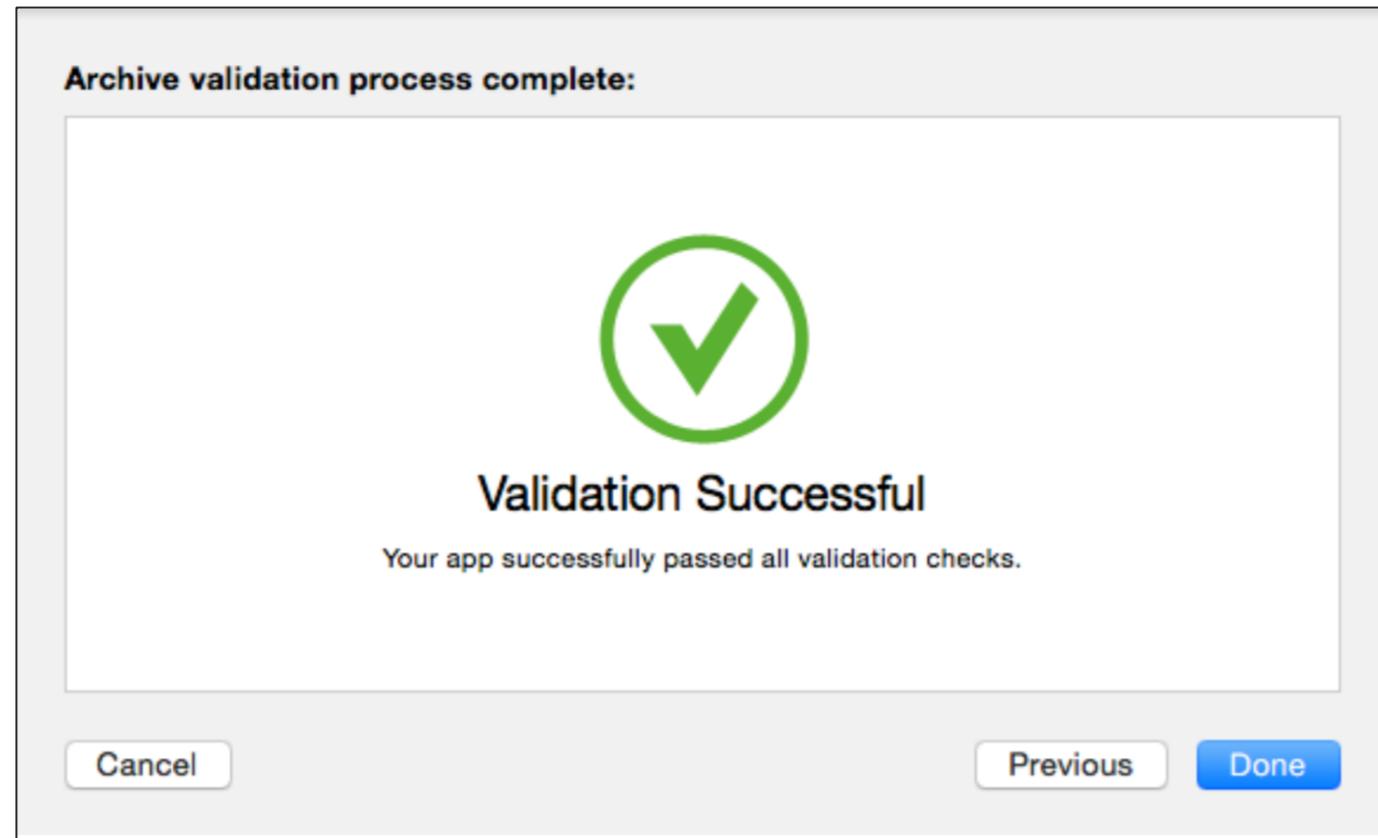


- Choose **Product > Archive**
- XCode should now generate a **.xcarchive** for you.

# Submitting an app using XCode

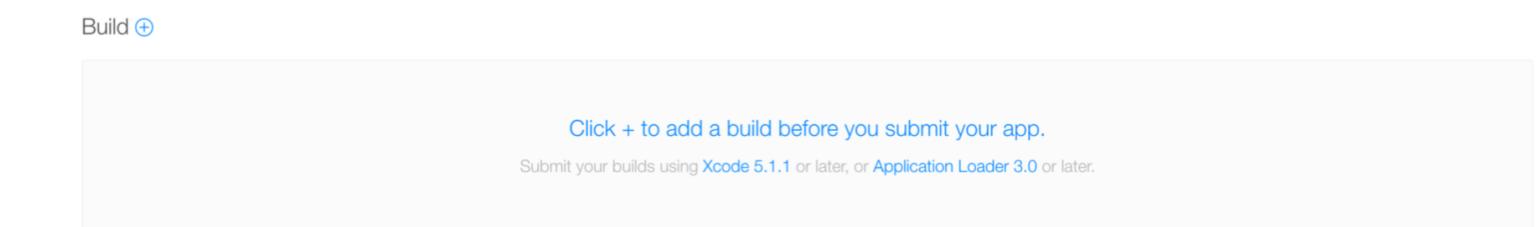


# Submitting an app using XCode

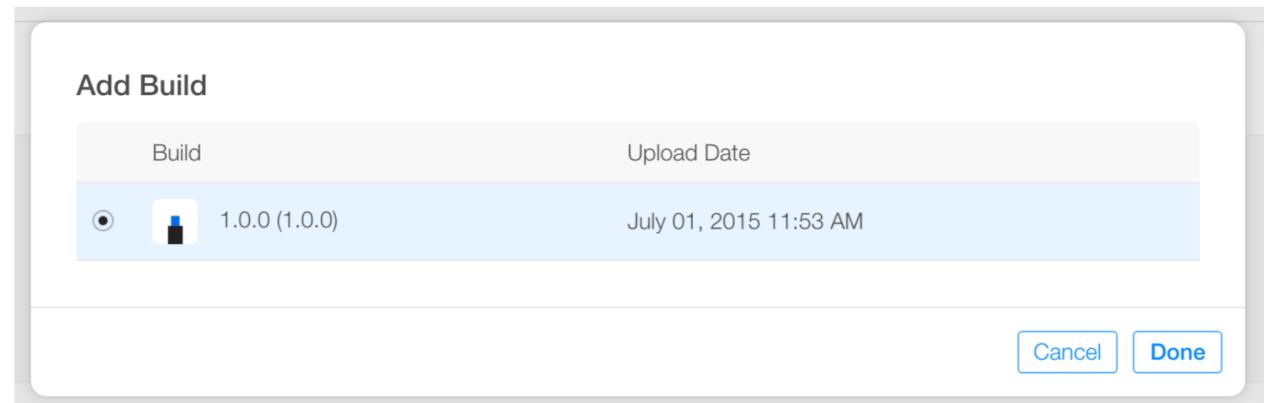


## Submit for Review

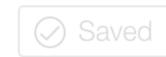
Once you have uploaded your application, either through XCode or Application Loader, you will need to finish up your app store listing in iTunes Connect. Go back to your application in iTunes Connect and go to the Build section:



You should now see a + icon as shown above. Click this, select the build you just uploaded and hit **Done**:



Double check everything in your listing, then go back to the top of the page, hit **Save** and then **Submit for Review**:



**Submit for Review**

To submit your application to Apple. Now you just have to cross your fingers and wait! **The Apple review process usually takes around 3-7 days**, which is a frustratingly long time. There's nothing you can do about it though but sit back and wait. Just make sure that you are complying with all of Apples rules and guidelines so that your app does not get rejected (otherwise you will have to fix it and wait another 3-7 days!).

# Soumettre au Google play store

# Submitting

- **Créer un compte développeur:**

<https://play.google.com/apps/publish/signup/>

Remember, before submitting to Google Play you must sign your **.apk** with a keystore file.

 \$25

Préparez votre carte de paiement pour régler les frais d'inscription de 25 USD à la prochaine étape.

Create a file at **platforms/android/release-signing.properties** and add the following:

```
storeFile=snapaday-release.keystore  
keyAlias=snapaday
```

This file tells the build process how we want the application to be signed. Here you will supply the **keystore** file you generated in the signing lesson, as well as the alias of the keystore. The first line should be the path to where the keystore file is stored, for simplicity I move the keystore file to the same location as this file, however you can also specify a different path if you want. The second line is the alias name.

---

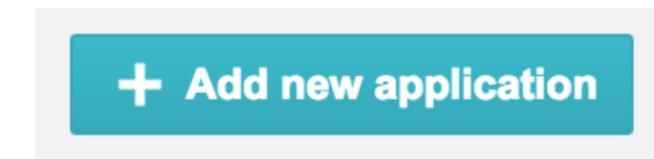
```
ionic build android --release
```

your **.apk** file (or files if you are using Crosswalk) will be generated for you at: **platforms/android/build/outputs/apk/**

# Submit for Review

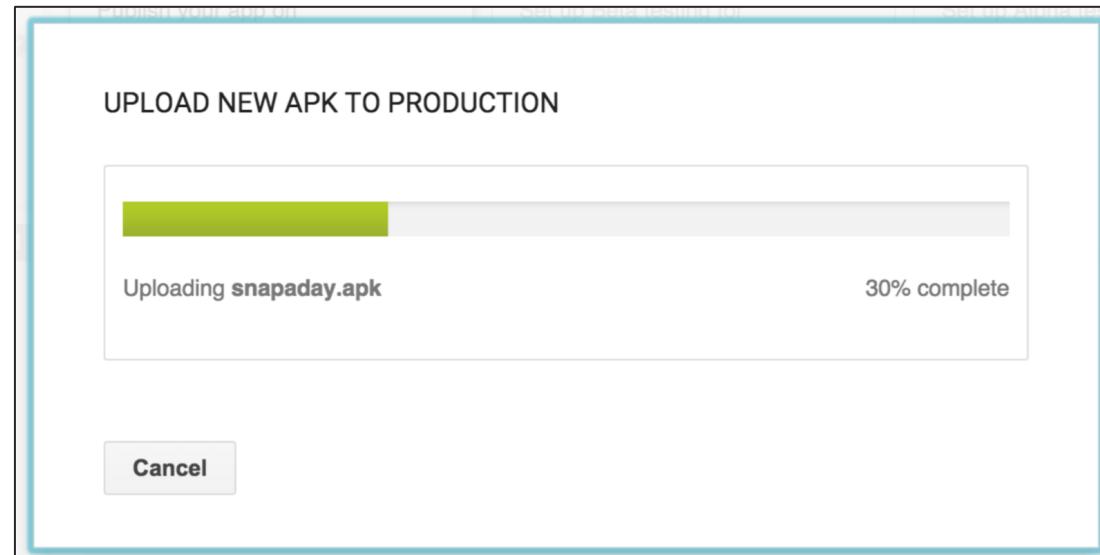
- Login here:

<https://play.google.com/apps/publish>

A screenshot of the Google Play Developer Console showing the app dashboard for "Snapaday". The dashboard is divided into several sections:

- APK** (highlighted):
  - Store Listing
  - Content Rating
  - Pricing & Distribution
  - In-app Products
  - Services & APIs
  - Optimization Tips
- APK**:
  - PRODUCTION**: Publish your app on Google Play
  - BETA TESTING**: Set up Beta testing for your app
  - ALPHA TESTING**: Set up Alpha testing for your app
- A central message: "License keys are now managed for each application individually. If your application uses licensing services (e.g. if your app is a paid app, or if it uses in-app billing or APK expansion files), get your new license key on the Services & APIs page." It features an Android icon with a yellow sunburst and the word "NEW".
- A large button at the bottom right: "Upload your first APK to Production".
- Below the production button: "Do you need a license key for your application? Get license key".

# Submit for Review



The screenshot displays the developer console for the app "Snapaday" (com.joshmorony.development). The status is listed as "DRAFT". On the right, there's a summary section for the "APK" tab, showing the current version is 18, and sections for "PRODUCTION", "BETA TESTING", and "ALPHA TESTING". Below this, under "PRODUCTION CONFIGURATION", there's a button labeled "Upload new APK to Production". The "CURRENT APK" was uploaded on Jul 1, 2015, at 1:10:02 AM. The "Supported devices" count is 6945, and "Excluded devices" count is 0. A table lists the current APK details: Version 18 (0.0.1), Uploaded on Jul 1, 2015, and Status Draft in Prod. At the bottom, there's a section for "APP TRANSLATION SERVICE" with a link to learn more about translating the application for other languages.

# Submit for Review

PRICING & DISTRIBUTION

This application is Paid Free

Setting the price to 'Free' is permanent. You cannot change it back to 'Paid' again after publishing. [Learn more](#)

DISTRIBUTE IN THESE COUNTRIES

You have selected 140 countries + Rest of the world

SELECT ALL COUNTRIES

Albania  
 Algeria  
 Angola  
 Antigua and Barbuda  
 Argentina  
 Armenia  
 Aruba  
 Australia

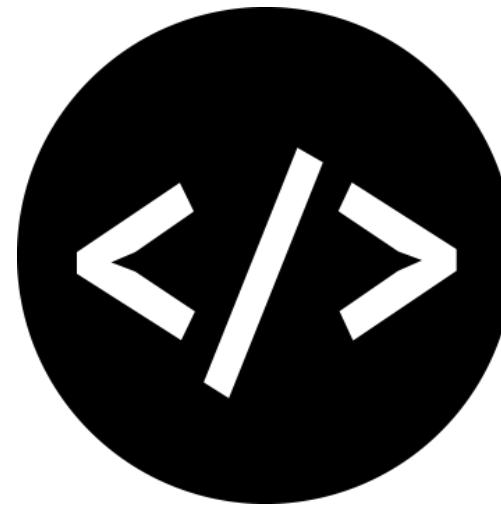
[Show options](#)

Once you have filled out all of the screens just hit **Publish App** and your application will be submitted! Unlike the Apple App store, your application should be available on Google Play within hours.

We are done !



Sass



A vous de  
jouer !