Publisher v3

Documentation

Description

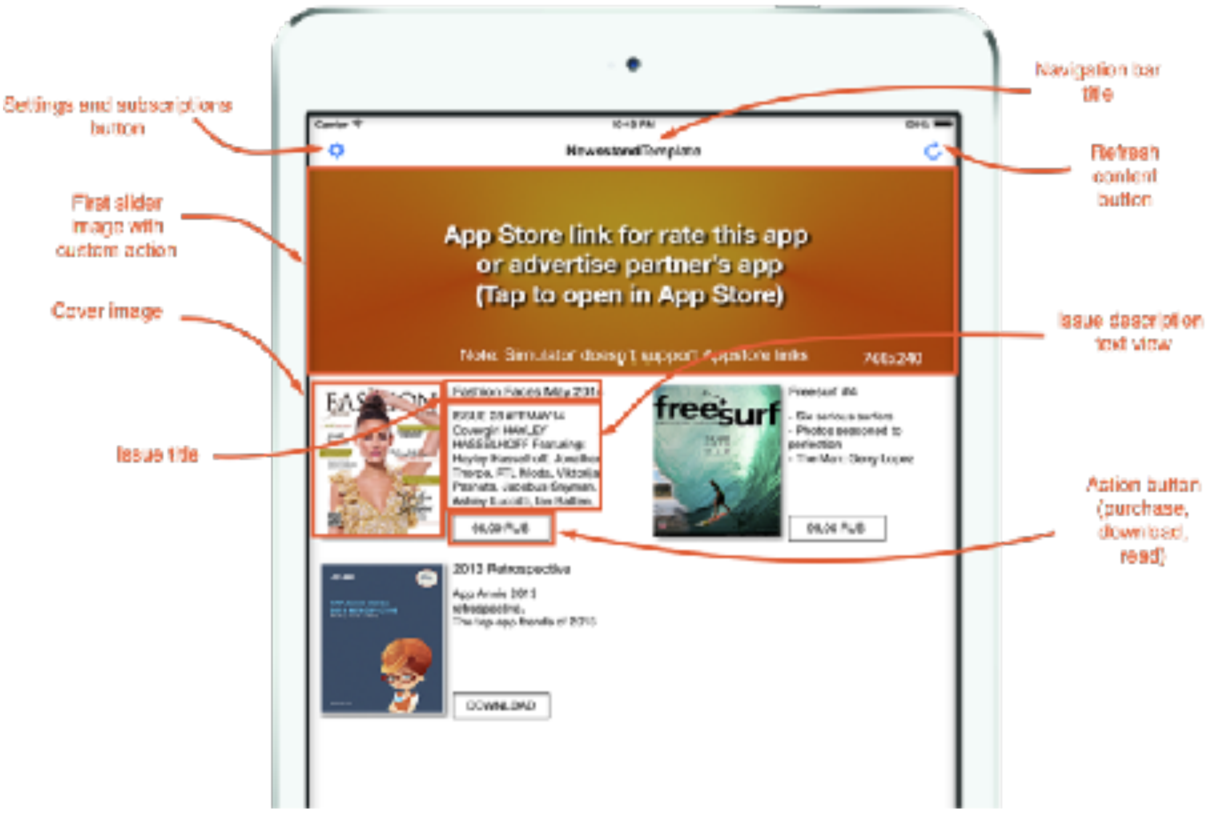
Publisher - Xcode project of universal iOS app for selling PDF publications. Programming language - Swift 3.

Complitable with iOS9 - iOS 10.3:
iPhone 4s, 5, 5c, 5s, 6, 6 Plus, SE, 6s, 6s Plus, 7, 7Plus
iPad 2, 3, Mini, Mini 2, Mini 3, Mini 4, Air, Pro
iPod Touch 5g, 6g

How does it works

Publisher app will connect to a LibraryConfig file that is hosted on a server (Amazon S3, Google Drive, GitHub etc.), then download it and parse it. LibraryConfig.plist file contains links to PDF files, cover images, publication title, description and other information.

This template contains demo files, just build and run Xcode project (Open in Xcode Publisher.xcworkspace and press «Build & Run» button.)


Settings and subscriptions button
First slider image with custom action
Cover image
Issue title
Navigation bar title
Refresh content button
Issue description text view
Action button (purchase, download, read)
NewsstandTemplate
App Store link for rate this app
or advertise partner's app
(Tap to open in App Store)
Note: Simulator doesn't support AppStore links
768x240
Fashion Faces May 2014
Freesurf #4
freesurf
2013 Retrospective
DOWNLOAD

Features:

- Awesome image slider with actions
- Supports offline usage
- Image caching
- Easy configurable
- Online updating and adding new issues
- Supports Push Notification (Pushbots.com)
- Internal Web browser
- Highlighting and handling PDF links on page
- Portrait and landscape orientations

Requirements

- An Apple Developer Account.
- A Mac with Xcode installed.
- No programming skills required!
- PushBots.com account for sending Push Notifications
- Hosting (Dropbox, Google Drive or any other)

Step by step guide

Step 1. Prepare magazine content (make slides, covers etc.).

Step 2. Upload to hosting (Amazone S3, Google Drive,).

Step 3. Edit LibraryConfig.plist file (set links to images, PDFs, set magazine title, issues's name , date, description etc.).

Step 4. Setup Xcode project (set bundle id, set URL to LibararyConfig.plist file, set PushBots App id).

Step 5. Create and configure an App in iTunes Connect.

Step 6. Add Push Notifications.

Step 7. Test your application.

Step 8. Publishing.

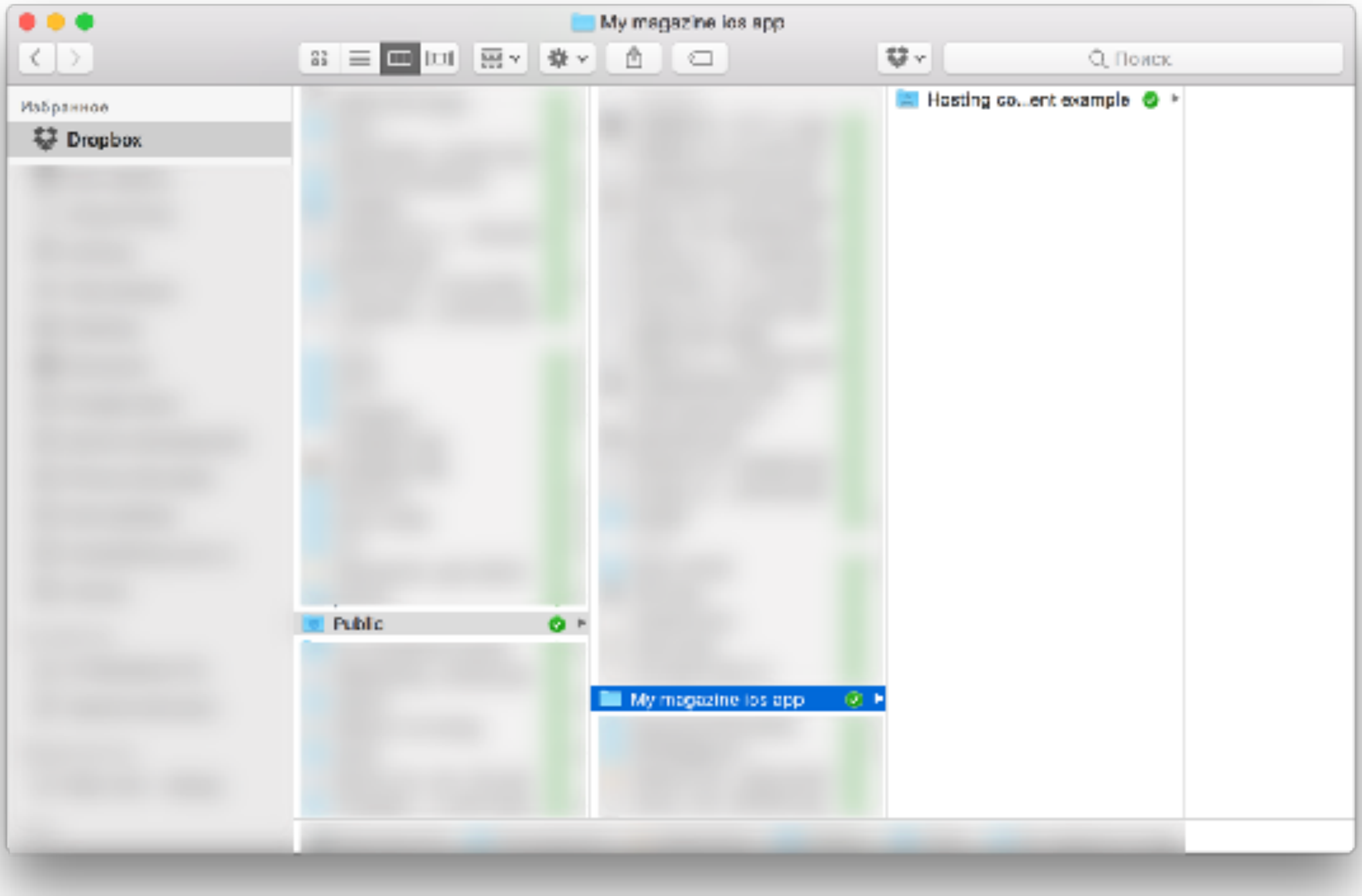
Step 1. Prepare magazine content.

Prepare your images according table below.
In folder **../demo/images/** you will see demo images.

Table of image sizes

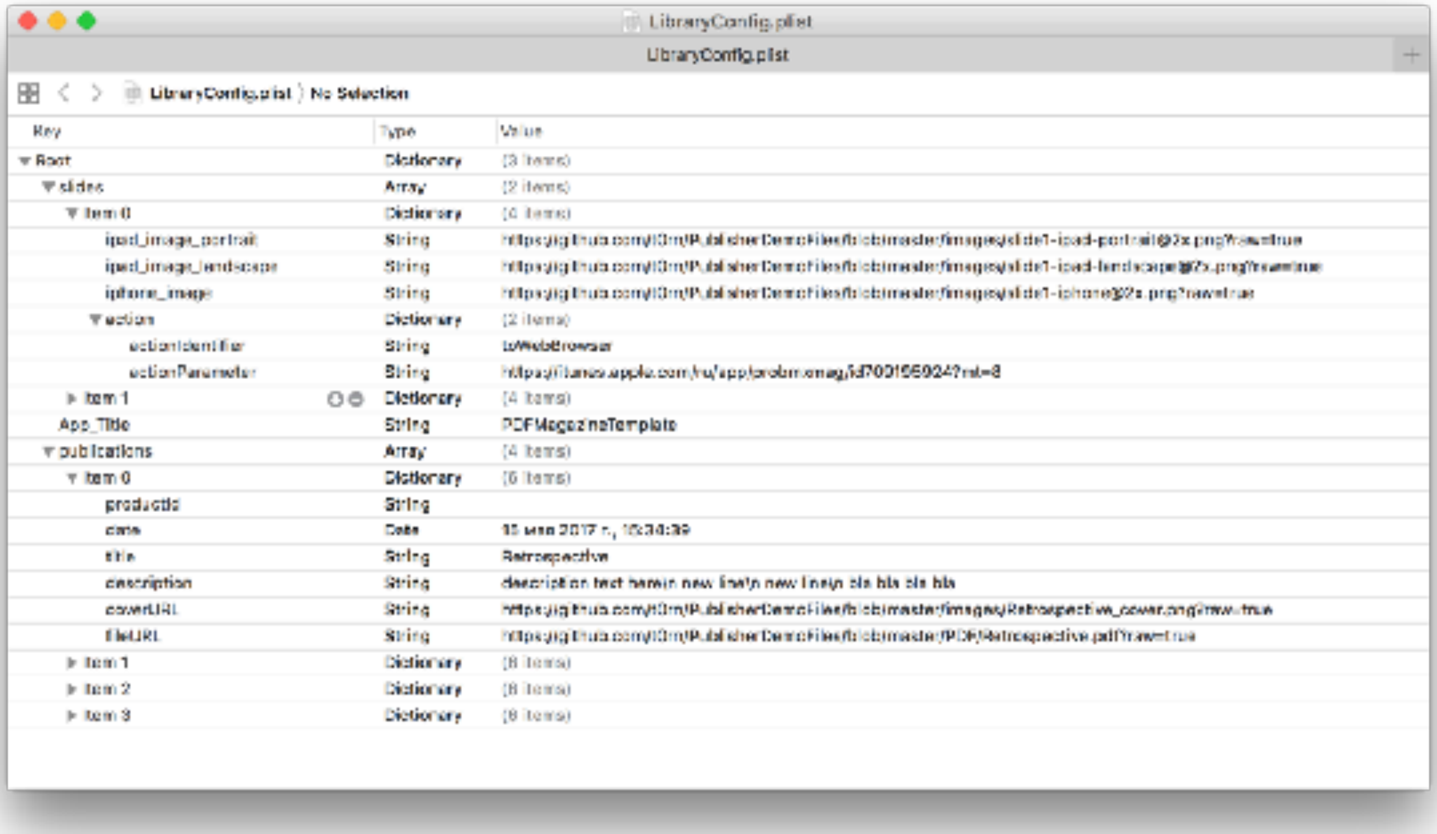
Image types	key in LibraryConfig.plist	iPhones	iPads
Slider images	ipad_image_portrait		1536x480 px
	ipad_image_landscape		2048x480 px
	iphone_image	640x240 px	
publication cover	coverURL	250x325px	250x325px

Step 2. Upload files to hosting (Amazone S3, Google Drive, Dropbox etc.).



Step 3. Edit *.plist files

3.1 Open LibararyConfig.plist file



Key	Type	Value
▼ Root	Dictionary	(3 items)
▼ slides	Array	(2 items)
▼ Item 0	Dictionary	(4 items)
ipad_image_portrait	String	https://github.com/t0rn/PublisherDemoFiles/blob/master/images/slide1-ipad-portrait@2x.png?raw=true
ipad_image_landscape	String	https://github.com/t0rn/PublisherDemoFiles/blob/master/images/slide1-ipad-landscape@2x.png?raw=true
iphone_image	String	https://github.com/t0rn/PublisherDemoFiles/blob/master/images/slide1-iphone@2x.png?raw=true
▼ action	Dictionary	(2 items)
actionIdentifier	String	toWebBrowser
actionParameter	String	https://itunes.apple.com/ru/app/probm.emag/id700195924?mt=8
▶ Item 1	Dictionary	(4 items)
App_Title	String	PDFMagazineTemplate
▼ publications	Array	(4 items)
▼ Item 0	Dictionary	(6 items)
productId	String	
date	Date	15 мая 2017 г., 15:34:39
title	String	Retrospective
description	String	description text here\n new line\n new line\n bla bla bla bla
coverURL	String	https://github.com/t0rn/PublisherDemoFiles/blob/master/images/Retrospective_cover.png?raw=true
fileURL	String	https://github.com/t0rn/PublisherDemoFiles/blob/master/PDF/Retrospective.pdf?raw=true
▶ Item 1	Dictionary	(8 items)
▶ Item 2	Dictionary	(8 items)
▶ Item 3	Dictionary	(8 items)

«App-Title» - Title in PublicationsViewController navigation item
«slides» - array of dictionaries with keys:

«ipad_image_portrait» - URL to slide image for iPads in portrait mode
«ipad_image_landscape» - URL to slide image for iPads in landscape mode
«iphone_image» - URL to slide image for iPhones

«action» - dictionary with action type and action parameter that handle when user tap on current slide

«actionIdentifier» - type of slide action.
Note: Currently supports one type of action:
«toWebBrowser» (open URL in internal browser)

«actionParameter» - URL for internal browser

«publications» - array of dictionaries that describes each publication

«productId» - Product ID for "non-consumable" purchase

«date» - Publication date

«title» - title publication

Note: publications will be ordered by date

«description» - description of current issue (use “\n” for new line)

«coverURL" - URL to cover image

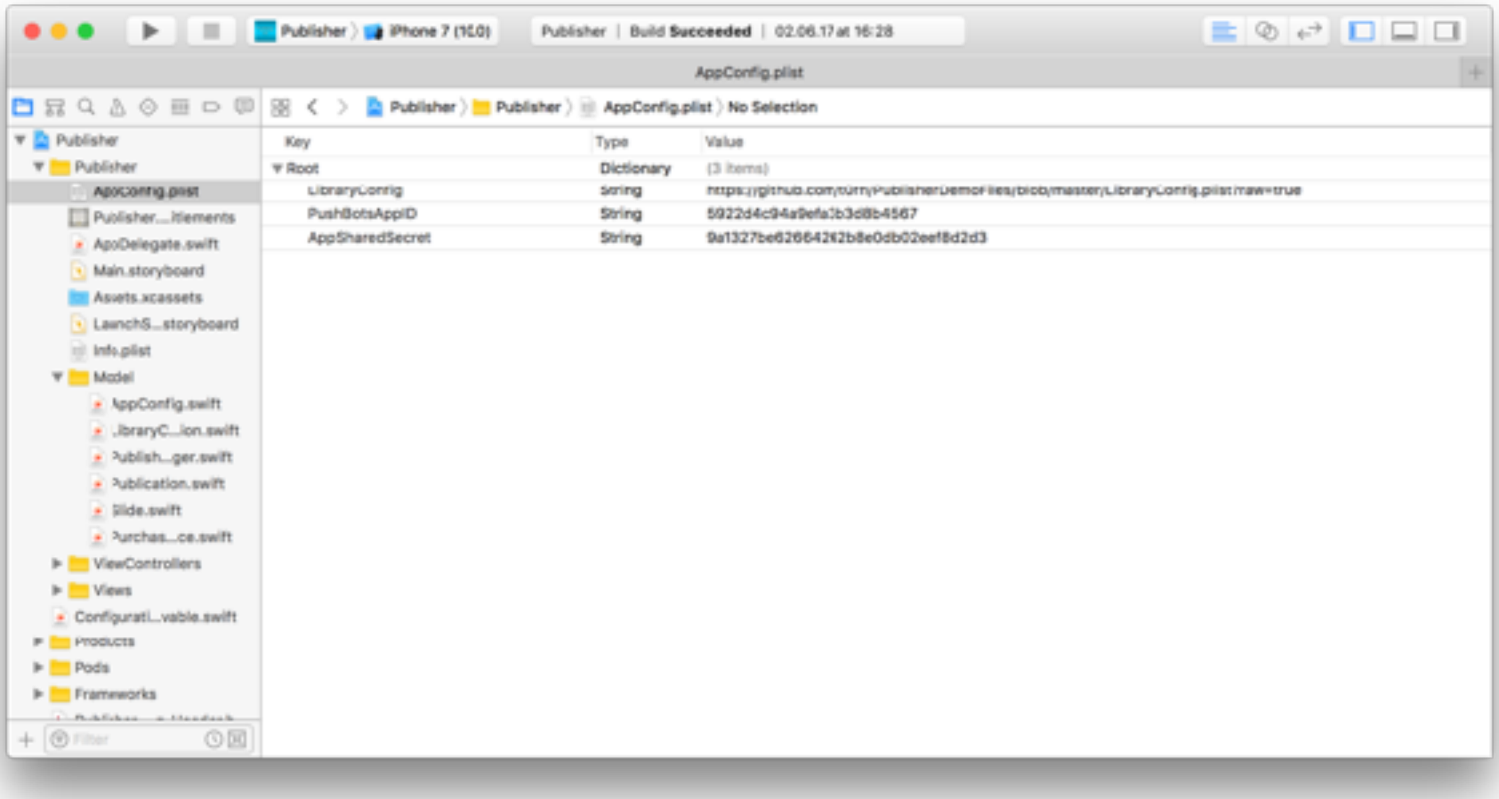
«fileURL» - URL to PDF file

Replace URLs from step 2, then save a LibraryConfig.plist file, upload to a hosting and copy direct URL to this file. You will need this URL for next step.

Step 4. Setup Xcode project

4.1 Prepare AppConfig.plist

Open Publisher.xcworkspace in Xcode and select AppConfig.plist in project navigator

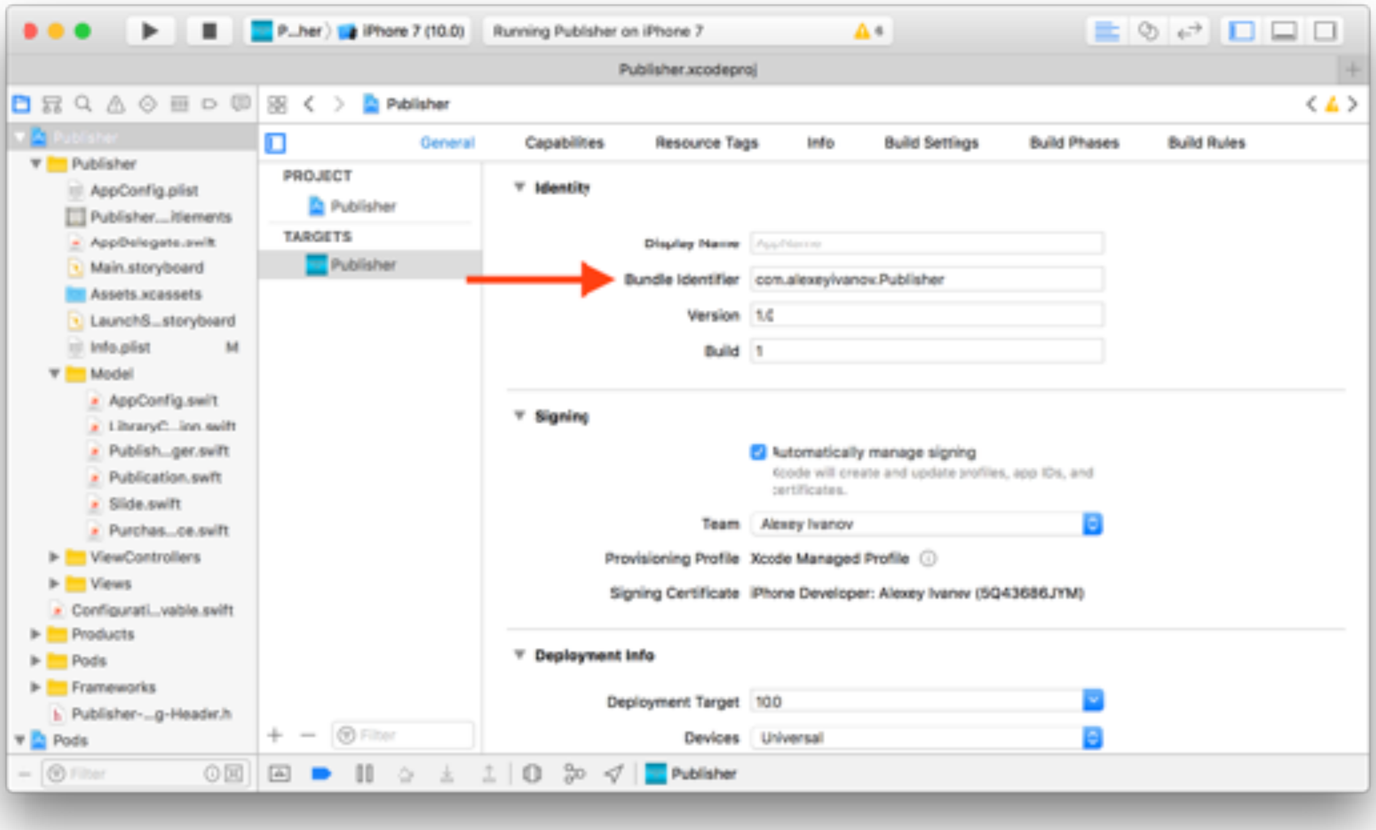


Key	Type	Value
▼ Root	Dictionary	(3 items)
LibraryConfig	String	https://github.com/t0rn/PublisherDemoFiles/blob/master/LibraryConfig.plist?raw=true
PushBotsAppID	String	5922d4c94a9efa3b3d8b4567
AppSharedSecret	String	9a1327be62664262b8e0db02eef8d2d3

Paste URL to LibraryConfig.plist from step 3.1 for LibraryConfig key

4.2. Bundle Identifier

Open Publisher.xcworkspace and set your Bundle Identifier



Identity

Display Name: AppName

Bundle Identifier: com.alexeyivanov.Publisher

Version: 1.0

Build: 1

Signing

☑ Automatically manage signing
Xcode will create and update profiles, app IDs, and certificates.

Team: Alexey Ivanov

Provisioning Profile: Xcode Managed Profile

Signing Certificate: iPhone Developer: Alexey Ivanov (5Q43686JYM)

Deployment Info

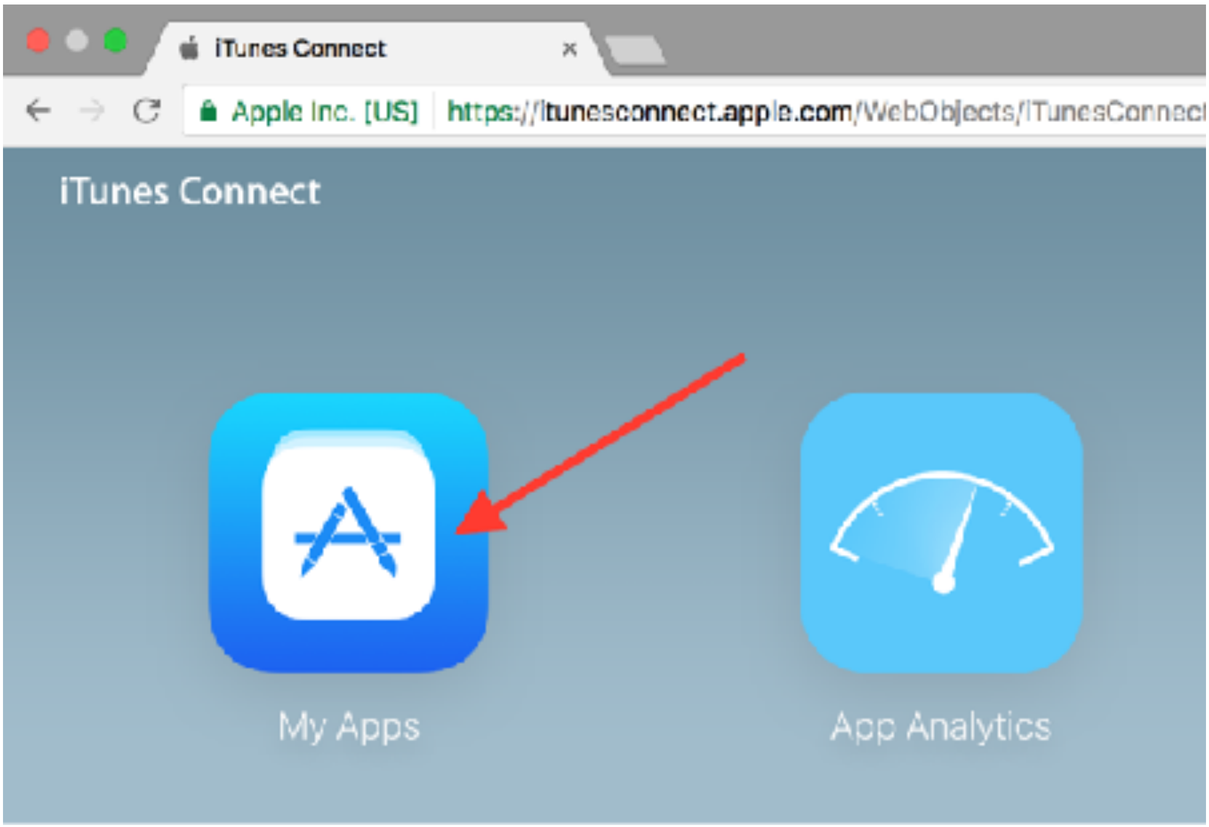
Deployment Target: 10.0

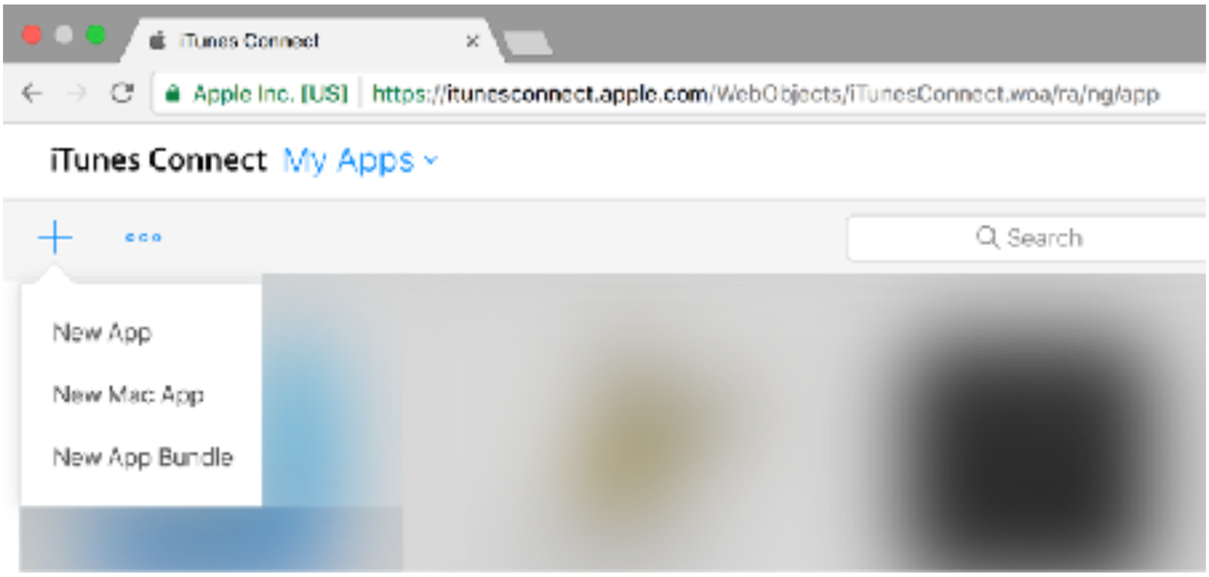
Devices: Universal

Step 5. Create app in iTunes Connect and add In-App Purchases

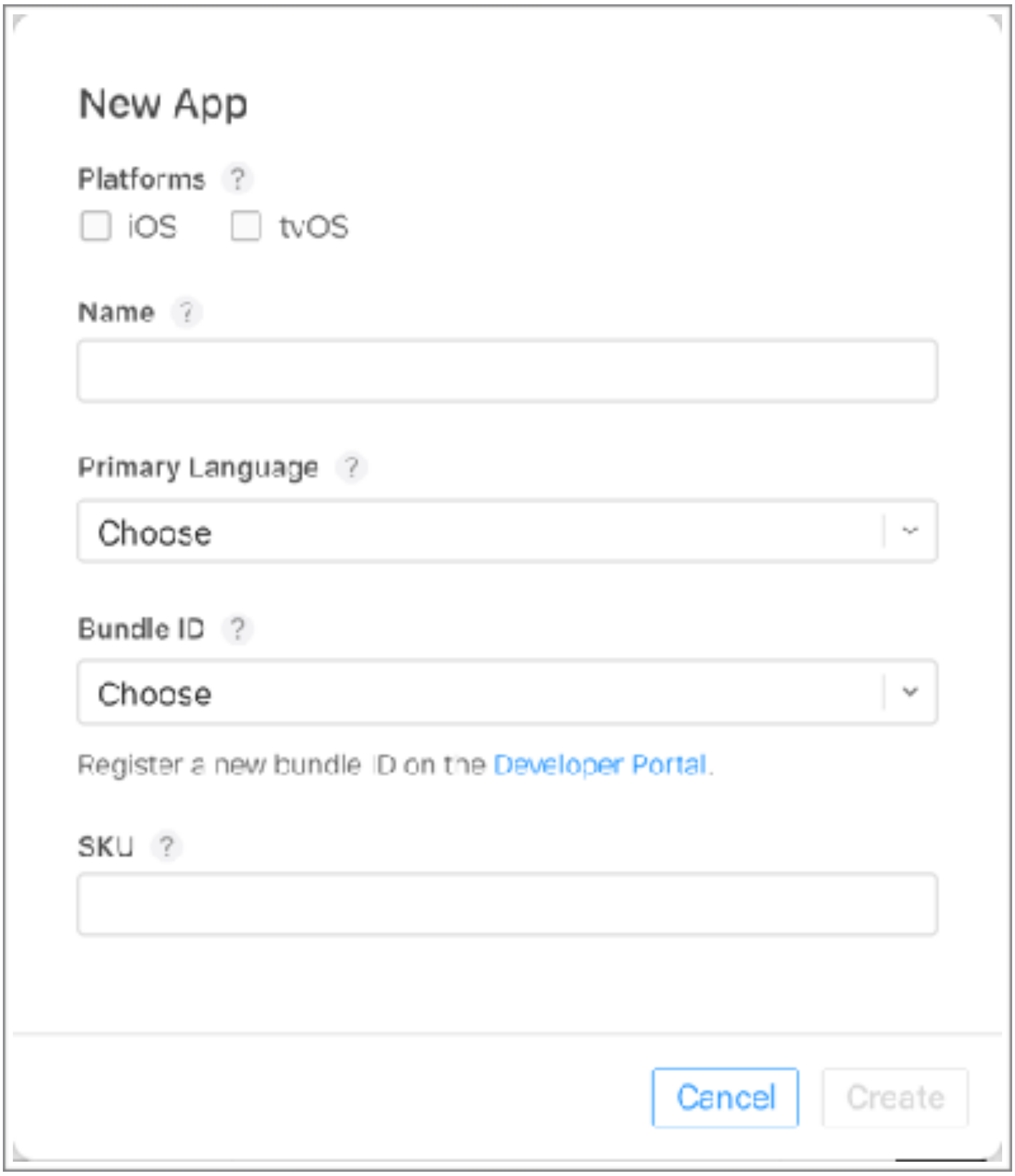
5.1. Create app

Login to iTunes Connect, click «My Apps», «+» and «New App»





Fill fields and create Bundle ID (click on Developer Portal link) same as at step 4.2



New App

Platforms ?

□ iOS □ tvOS

Name ?

Primary Language ?

Choose

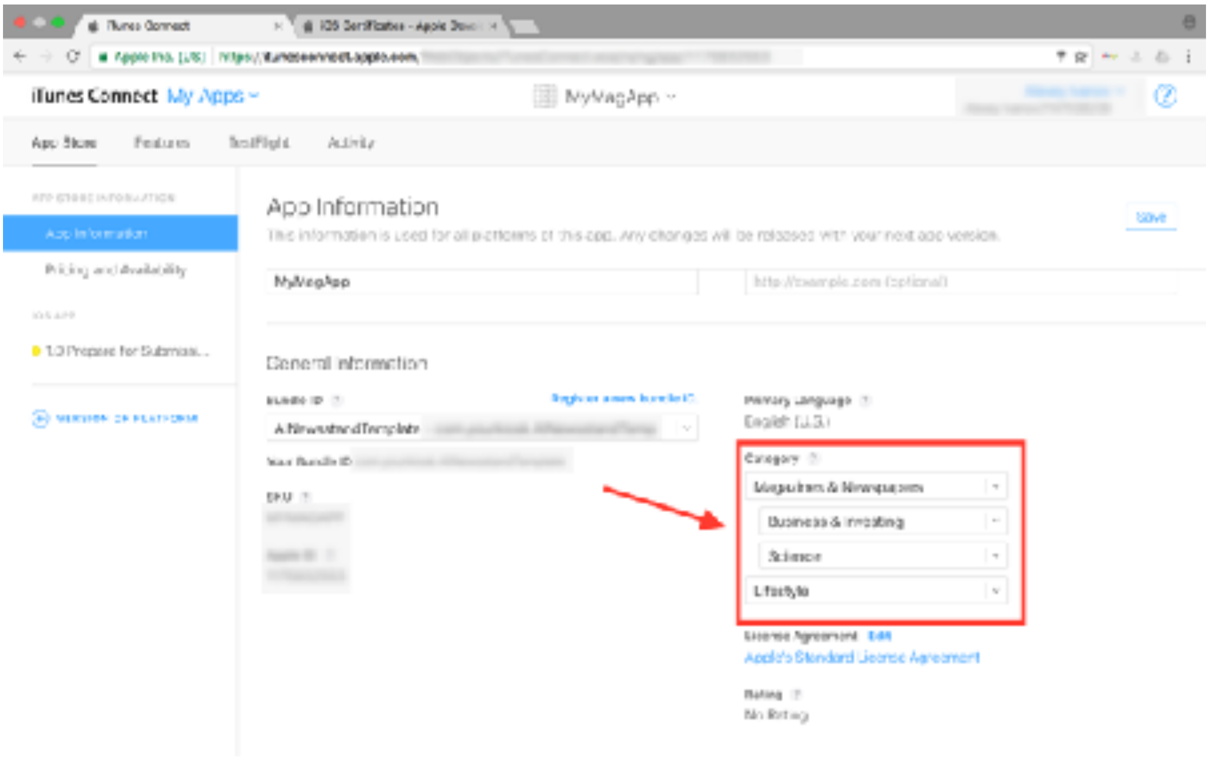
Bundle ID ?

Choose

Register a new bundle ID on the Developer Portal.

SKU ?

Fill App Information page and choose first categories



iTunes Connect | iOS Certificates - Apple Dev...

Apple Inc. [US] https://itunesconnect.apple.com/[redacted]

iTunes Connect My Apps

MyMagApp

App Store Features TestFlight Activity

APP STORE INFORMATION

App Information

Pricing and Availability

iOS APP

1.0 Prepare for Submission...

VERSION OR PLATFORM

App Information

This information is used for all platforms of this app. Any changes will be released with your next app version.

Save

MyMagApp

http://example.com (optional)

General Information

Bundle ID ? Register a new bundle ID

A.NewsstandTemplate [redacted]

Your Bundle ID [redacted]

SKU ? [redacted]

Apple ID ? [redacted]

Primary Language ?

English (U.S.)

Category ?

Magazines & Newspapers

Business & Investing

Science

Lifestyle

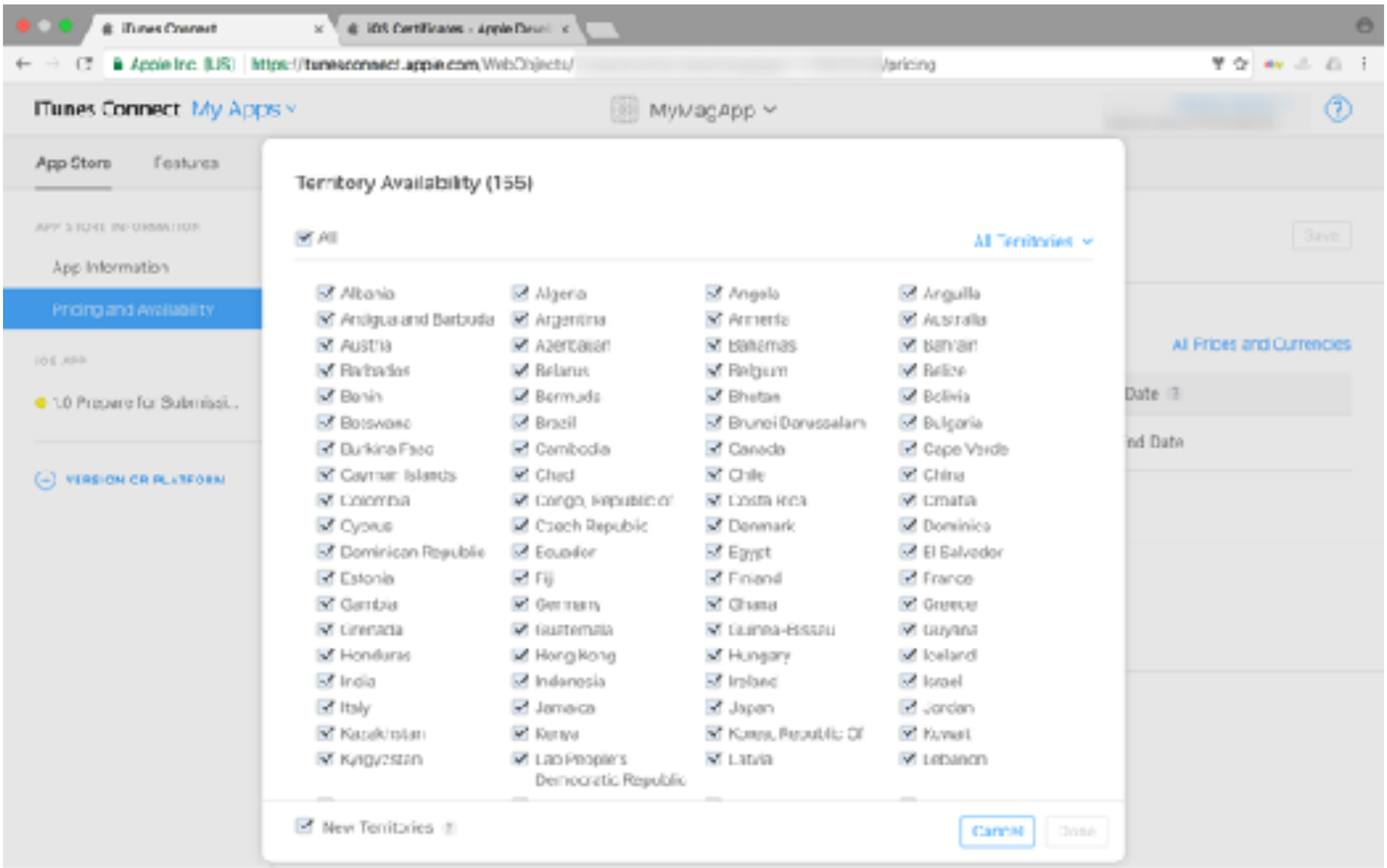
License Agreement Edit

Apple's Standard License Agreement

Rating ?

No Rating

In Pricing and Availability section set price of your app and Territory Availability



Territory Availability (155)

☑ All

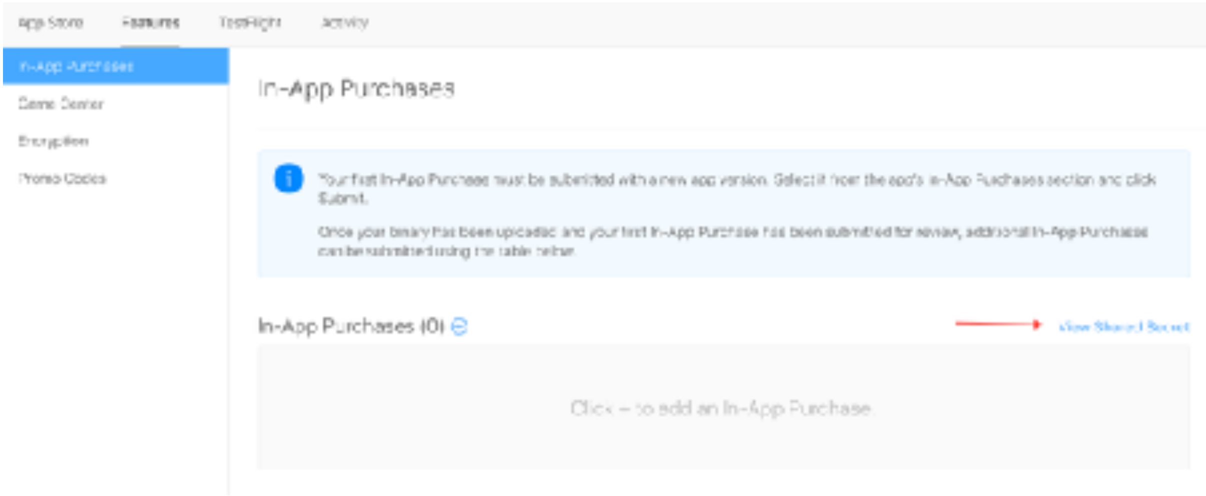
All Territories

☑ Albania ☑ Algeria ☑ Angola ☑ Anguilla
☑ Antigua and Barbuda ☑ Argentina ☑ Armenia ☑ Australia
☑ Austria ☑ Azerbaijan ☑ Bahamas ☑ Bahrain
☑ Barbados ☑ Belarus ☑ Belgium ☑ Belize
☑ Benin ☑ Bermuda ☑ Bhutan ☑ Bolivia
☑ Botswana ☑ Brazil ☑ Brunei Darussalam ☑ Bulgaria
☑ Burkina Faso ☑ Cambodia ☑ Canada ☑ Cape Verde
☑ Cayman Islands ☑ Chad ☑ Chile ☑ China
☑ Colombia ☑ Congo, Republic of ☑ Costa Rica ☑ Croatia
☑ Cyprus ☑ Czech Republic ☑ Denmark ☑ Dominica
☑ Dominican Republic ☑ Ecuador ☑ Egypt ☑ El Salvador
☑ Estonia ☑ Fiji ☑ Finland ☑ France
☑ Gambia ☑ Germany ☑ Ghana ☑ Greece
☑ Grenada ☑ Guatemala ☑ Guinea-Bissau ☑ Guyana
☑ Honduras ☑ Hong Kong ☑ Hungary ☑ Iceland
☑ India ☑ Indonesia ☑ Ireland ☑ Israel
☑ Italy ☑ Jamaica ☑ Japan ☑ Jordan
☑ Kazakhstan ☑ Kenya ☑ Korea, Republic Of ☑ Kuwait
☑ Kyrgyzstan ☑ Lao People's Democratic Republic ☑ Latvia ☑ Lebanon

☑ New Territories

Cancel Done

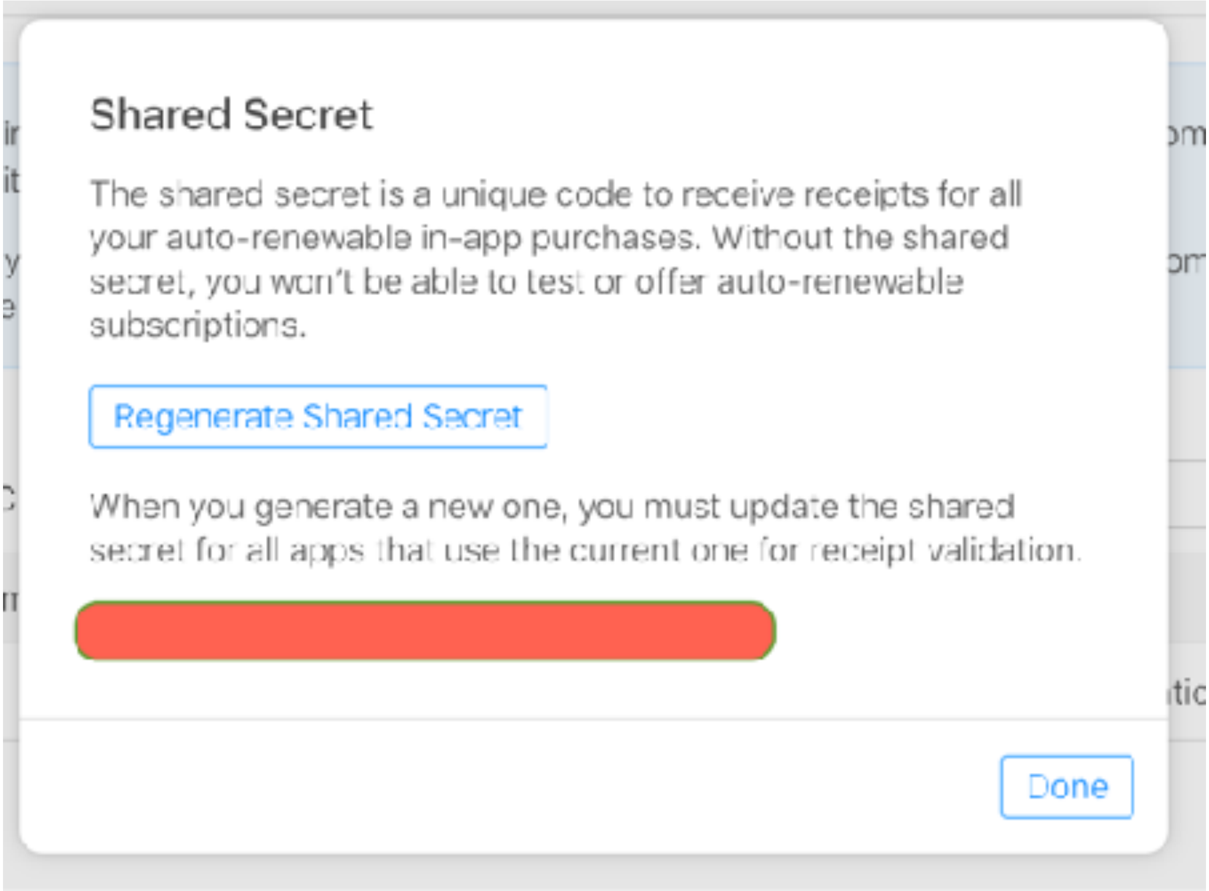
Copy and paste Shared Secret code into AppConfig.plist for AppSharedSecret key



Your first In-App Purchase must be submitted with a new app version. Select it from the app's In-App Purchases section and click Submit.

Once your binary has been uploaded and your first In-App Purchase has been submitted for review, additional In-App Purchases can be submitted using the table below.

Click + to add an In-App Purchase.



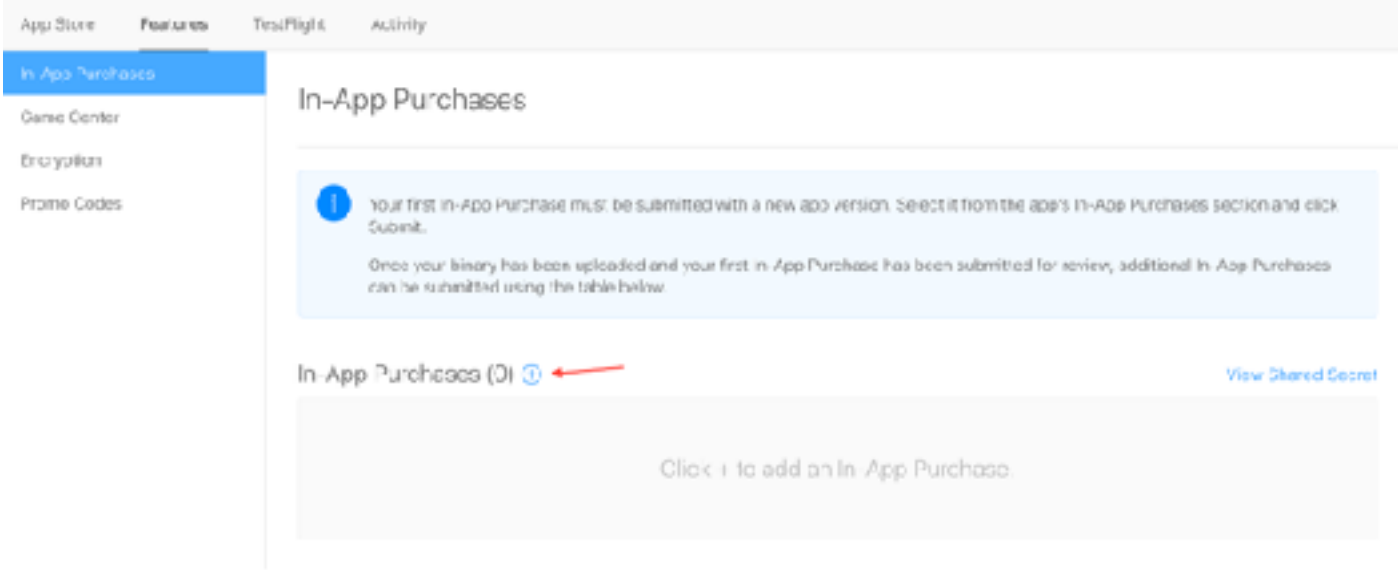
Shared Secret

The shared secret is a unique code to receive receipts for all your auto-renewable in-app purchases. Without the shared secret, you won't be able to test or offer auto-renewable subscriptions.

When you generate a new one, you must update the shared secret for all apps that use the current one for receipt validation.

[redacted]

5.2. Add In-App Purchases



In-App Purchases

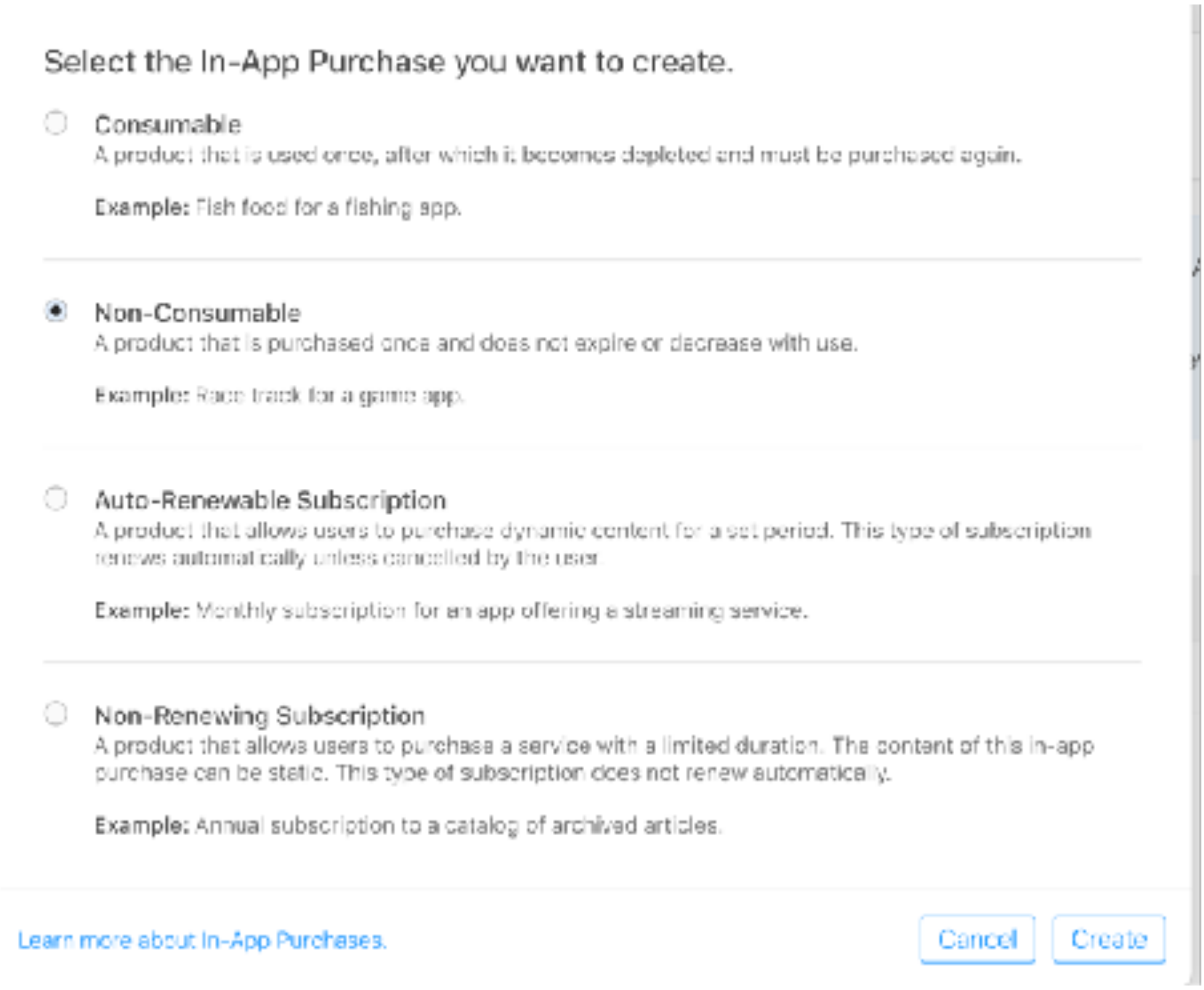
Your first In-App Purchase must be submitted with a new app version. Select it from the app's In-App Purchases section and click Submit.

Once your binary has been uploaded and your first In-App Purchase has been submitted for review, additional In-App Purchases can be submitted using the table below.

In-App Purchases (0)

Click + to add an In-App Purchase.

Use Non-Consumable In-App Purchase for paid publications.



Select the In-App Purchase you want to create.

○ **Consumable**
A product that is used once, after which it becomes depleted and must be purchased again.

Example: Fish food for a fishing app.

● **Non-Consumable**
A product that is purchased once and does not expire or decrease with use.

Example: Race track for a game app.

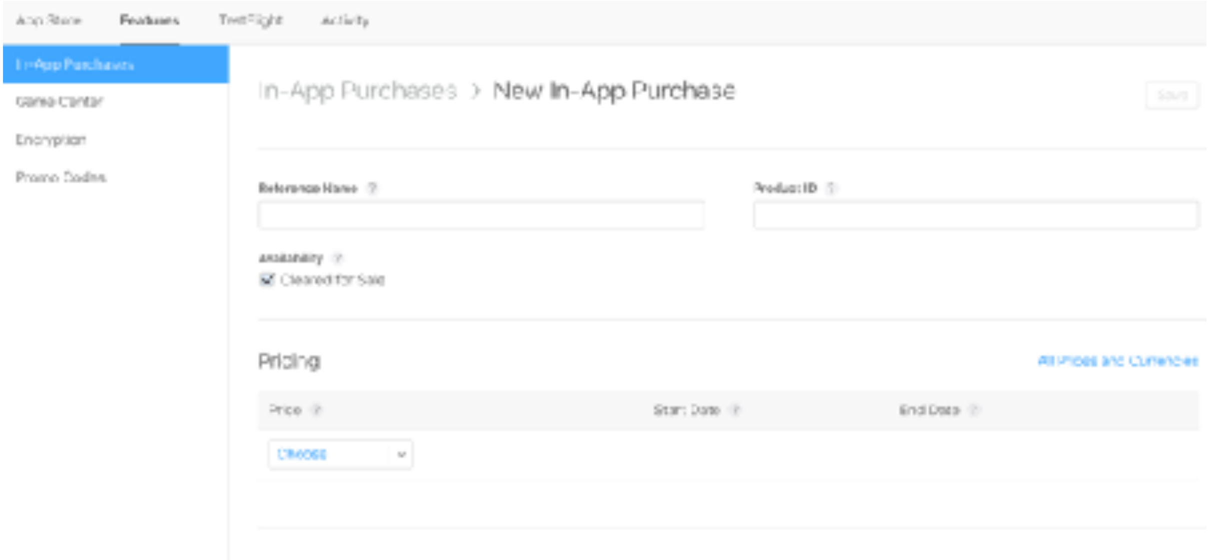
○ **Auto-Renewable Subscription**
A product that allows users to purchase dynamic content for a set period. This type of subscription renews automatically unless cancelled by the user.

Example: Monthly subscription for an app offering a streaming service.

○ **Non-Renewing Subscription**
A product that allows users to purchase a service with a limited duration. The content of this in-app purchase can be static. This type of subscription does not renew automatically.

Example: Annual subscription to a catalog of archived articles.

Fill In-App purchase information



In-App Purchases > New In-App Purchase

Reference Name ?

Product ID ?

Availability ?
☑ Cleared for Sale

Pricing

Price ?	Start Date ?	End Date ?
Choose		

Copy and paste «Product ID» into LibraryConfig.plist productId key's value.
Don't turn on Apple content hosting.

In-App purchases testing notes:

Double check Bundle ID in iTunes Connect and in Xcode project
Use real device for testing.
In-App Purchases in iTunes Connect should have «Ready to Submit» status.
Data about purchasing stored in Apple «Keychain», so user will be able to download publication even after reinstallation of application.

Please read Testing In-App Purchase Products for better understanding.

Step 6. Add Push Notifications

6.1. Generate App Id for PushBots.com SDK

Register on https://pushbots.com and create new app on dashboard. Copy «Application ID» for this app and paste into AppConfig.plist PushBotsAppID key's value

6.2. Configure App for push notifications

Go through steps 1-3 from https://pushbots.com/developer/docs/ios-configuration

Please test push notifications on real devices.