



GDxRepo 31 марта 2016 в 18:20

Сертификация в Apple Developer Center простым и понятным языком

Objective C, Xcode, Разработка под iOS

Из песочницы

Кратко о главном

В Apple Developer Center с незапамятных времен применяется довольно мудреная система сертификации ваших приложений на каждом из ключевых этапов — разработка, тестирование и публикация.

Зачастую при первом погружении в эту систему у начинающих (и не только) разработчиков возникают серьезные проблемы с пониманием как функционирует Apple Developer Center (будем называть его «девцентр» для простоты). В результате, мне в процессе профессиональной деятельности не раз приходилось наблюдать на новых местах работы огромные свалки из профилей и сертификатов в девцентре, в результате чего приходилось приступать к «разбору завалов».

При этом, в сети довольно не такой большой выбор материалов на эту тему. Конечно, в официальной документации Apple все хорошо структурировано и очень подробно описано, но зачастую просто не хватает времени на изучение такого количества материала. Как правил хочется быстро понять, что именно и в каком порядке нужно сделать для корректной работы приложения на этапах разработки, тестирования при публикации его в магазин App Store. В русском же сообществе подобных материалов, собранных в одном месте и в удобном доступе, я видел вовсе, поэтому и решил написать эту статью. Для всех интересующихся — добро пожаловать под кат.

Что мы будем разбирать?

Мы разберем процесс управления вашим приложением в Apple Developer Center от его создания до публикации в магазине App Store. Мы не будем говорить только о базовых вещах, таких, как разработка, тестирование и публикация, а также обсудим APNs (Push Notifications).

Отмечу тот факт, что далее я буду описывать принцип работы девцентра по состоянию на 31 марта 2016 года, поэтому если вы читаете эту статью позднее — все уже могло измениться.

Что нам понадобится?

Собственно, для работы нам нужно следующее:

- Рабочий Mac, либо PC с виртуальной машиной и установленной на ней Mac OS.
- Действующий Apple ID. Его всегда можно бесплатно зарегистрировать на [официальном сайте компании Apple](#).
- На вашем Apple ID (либо у одной из компаний, которая добавила ваш Apple ID в свою команду) должна быть активирована так называемая Apple Developer Program — оплачиваемая раз в год «подписка», дающая вам доступ к Apple Developer Center и возможность публиковать ваши приложения в App Store. На текущий момент стоимость в пересчете на год невелика и составляет в районе \$99 за год пользования.
- И, конечно же, навыки разработки под iOS.

Ориентировка по разделам

В девцентре для полноценной работы с вашими приложениями нам понадобятся только два пункта:

- *Certificates, Identifiers & Profiles*. Раздел обеспечивает управление всей системой сертификации ваших приложений. Работу именно с этим разделом мы и будем разбирать в данной статье.
- *iTunes Connect*. Дает доступ к внутреннему и внешнему тестированию через TestFlight, а также к управлению публикацией ваших приложений в App Store.

Терминология

Давайте подробно разберем понятия, лежащие в основе функционирования девцентра Apple.

Сертификаты (Certificates)

Этот раздел дает доступ к управлению сертификатами, которыми обладает ваша учетная запись Apple ID. Каждый из этапов, которые вы будете проходить, будь то разработка, тестирование или публикация, включая все значимые составляющие экосистемы Apple вроде Push Notifications, требует обязательного наличия актуального (действующего, Active) сертификата. Говоря проще, ваше приложение не сможет даже «чихнуть» имея на то разрешения из Apple Developer Center. Чуть подробнее о подразделах:

- *Pending*. Запрошенные вами сертификаты, находящиеся в процессе обработки от Apple. Для дев (Development) и прод (Production) сертификатов конкретно в моем случае этот подраздел чаще всего пустует.
- *Development*. Дев-сертификаты, обеспечивающие возможность отладки вашего приложения на конкретных девайсах (одном либо нескольких) через Xcode, а также создание дев-сборок «в отладочном режиме». Более подробно поговорим о них чуть ниже.
- *Production*. Прод-сертификаты, обеспечивающие работоспособность приложения при тестировании в TestFlight и при публикации в магазине App Store.

Теперь разберем типы сертификатов.

Сертификаты типа «Development»

В первую очередь, нужно знать, что девелоперский сертификат всегда привязывается к одной конкретной машине. Поэтому для отладки на вашем Mac вам понадобится доступ к этому сертификату. Тут есть варианты. Например, если, вы устроились на работу iOS-программистом, ваша задача входит отладка на устройствах (как правило, так и есть), то есть два пути решения (какой из них выбирать — зависит от вас и условий работы в вашей компании):

- Создать отдельный дев-сертификат конкретно для вашего Mac, скачать и установить его. Плюс понадобится сгенерировать и установить свой Mac девелоперский *профиль* на основе этого сертификата, но об этом позже.
- Либо экспортировать с машины, на которую заведен сертификат, файл *.p12/*.pfx (это можно сделать в связке ключей Apple). Такой файл защищается паролем при экспорте, и, зная этот пароль, информацию о сертификате можно будет импортировать на любом другом Mac. В этом случае отпадает необходимость создавать для каждого Mac отдельные Development-сертификаты и отдельные Development-профили. Небольшая оговорка: профиль хоть и должен быть сгенерирован для той машины, на которую выпущен экспортируемый сертификат, но этот профиль понадобится добавить UDID вашего устройства прежде, чем выдавать профиль вам для установки, иначе ничего работать не будет.

Инструкция по процессу будет показана вам в девцентре Apple при начале создания сертификата, там все расписано очень подробно и по шагам, сложностей возникнуть не должно. Если вкратце, то после выбора типа сертификата (*iOS App Development*, для отладки приложения либо *APNs Sandbox*, для отладки пушей) вам придется создать файл запроса к бюро сертификации (Certificate Signing Identity Request), на которого и будет сгенерирован девелоперский сертификат. Если вы хотите и отлаживать приложение, и отлаживать пуш-нотификации в «песочнице», то вам потребуются оба этих сертификата. Забегая вперед, упомяну, что аналогичный процесс применяется и при создании прод-сертификатов.

Наличие дев-сертификата означает, что, скачав его и установив двойным кликом в Связку Ключей (Apple Keychain), вы сможете запускать и приложение напрямую через Xcode в режиме отладки на устройстве, подключив это устройство проводом к вашему Mac. Перечень разрешенных конкретных устройств Apple нужно будет обязательно указать при генерации девелоперского профиля, но об этом позже. Также, вы сможете собрать и экспортировать сборку с дев-профилем, но стоит учесть, что в этом случае ваше приложение не будет иметь доступа к продакшн-возможностям (APNs будет только в режиме sandbox, например).

Сертификаты типа «Production»

Для начала на всякий случай поясню, что сборкой iOS-приложения называют *.ipa-файл, архив, выпускаемый с соблюдением правил сертификации Apple через команду Project — Archive в Xcode.

Теперь о сертификации. Прод-сертификаты обеспечивают функционирование различных подсистем приложения в «боевых» условиях, то есть в магазине App Store, а также на устройствах, где выполняется внутреннее и внешнее тестирование приложения через TestFlight. Здесь, по аналогии с Development-сертификацией, есть тип *App Store & Ad Hoc Production*, а также тип *APNs Production*, использующийся веб-сервером для рассылки push-уведомлений. Если вы планируете выпустить приложение, поддерживающее работу с пушами, то вам понадобятся оба сертификата, как App Store & Ad Hoc (на основе которого вы сделаете сборку и отправите приложение в iTunes Connect), так и APNs Production (его вы отдадите серверу, а тот воспользуется им для получения прав на рассылку пушей). В довершение к уже упомянутым подсистемам есть несколько других, обеспечивающих доступ к Wallet, Apple Watch и так далее, но их обзор выходит за рамки данной статьи.

Очень часто возникает вопрос о том, в чем же разница между App Store и тем самым Ad Hoc. Ранее они были представлены разными сертификатами, с некоторого времени Apple объединила их в единое целое, за что им большое спасибо. Чуть подробнее об этих разновидностях:

- *Выпуск сборок типа App Store*. Обеспечивает возможность тестировать приложение в TestFlight, как в режиме внутреннего, так и в режиме внешнего тестирования. Также дает возможность опубликовать приложение в App Store.

- *Выпуск сборок типа Ad Hoc.* Термин «Ad Hoc» можно перевести как «специальный», «для конкретной цели». Такой тип сертификации обеспечивает возможность запускать ваше приложение (включая все нужные подсистемы типа APNs) в боевых условиях, но только на конкретных девайсах, и без участия Xcode в процессе запуска. Другими словами, Ad Hoc необходим, если вы захотите поставить ваше приложение на стороннее устройство, не имея к нему прямого доступа (то есть не подсоединяя его проводом к вашему Mac, так как в случае вам бы хватило Development-сертификата), но при этом и не выкладывая приложение в iTunes Connect. Такой сертификат используется при создании специального Ad Hoc-профиля, о котором пойдет речь чуть позже.

Еще один частый вопрос: чем отличаются сборки, собранные на папе Development Certificate + Development Profile, и сборки, созданные в связке Distribution Certificate + Ad Hoc Profile? Ведь и там, и там нужно указывать перечень разрешенных для установки устройств, и то, и то устанавливаются через iTunes. В чем же различие? На деле, разница в том, что дев-сборка будет запускаться «в отладочном режиме», то есть например, APNs ей будут доступны только в режиме «sandbox». Продакшн-сборка будет обладать «боевыми» правами, с доступом во все подсистемы Apple вроде «настоящих» APNs, iCloud и так далее.

Intermediate Certificates

Некоторое время назад Apple внесла изменения в логику работы девцентра и своей системы сертификации, после чего на большинстве компьютеров пропала возможность делать сборки приложений, несмотря на наличие активных дев- и прод-сертификатов и актуальных профилей. Причина этого была в том, что Apple добавила дополнительное требование, чтобы на вашем Mac в связке ключей был установлен специальный сертификат под названием «*Worldwide Developer Relations Certificate Authority*». Он устанавливается автоматически с новыми версиями Xcode, но те, у кого Xcode уже был установлен ранее, просто должны были установить этот сертификат вручную, скачав его по ссылке из секции Intermediate Certificates в девцентре Apple, после чего проблемы со сборками исчезали. Больше никакой смысловой нагрузки этот сертификат не несет.

Идентификаторы (Identifiers)

Данный раздел обеспечивает управление идентификаторами. Для вашего приложения в минимальном исполнении понадобится App ID, управление которыми доступно в одноименном подразделе.

В буквальном переводе «App ID» означает «идентификатор приложения», что полностью отражает его суть. Любое ваше приложение, которое вы хотите отлаживать на устройстве Apple, тестировать через TestFlight и/или публиковать в магазин App Store, должно обладать собственным уникальным именем, по которому его можно однозначно идентифицировать среди тысяч других приложений. При добавлении нового App ID будет предложено ввести несколько элементов:

- *App ID Description.* Имя вашего приложения. К примеру, если ваше приложение называется *Mail Printer*, то прямо так его и записываем в текстовое поле.
- *App ID Prefix.* Префикс вашего приложения, он выдается вам автоматически и будет общим для конкретной команды Apple Team, где подключена и активна Apple Developer Program.
- *App ID Suffix.* Здесь нам понадобится выбрать *Explicit App ID*, чтобы указать бандл (bundle) приложения. Это идентификатор, обычно имеющий вид *com.mycompany.myappname*, где *mycompany* — имя вашей компании или вашего домена. Например, *com.homecompany.MailPrinter*. Обращаю ваше внимание, что точно такой же бандл должен быть выставлен в настройках таргета (Target) вашего приложения в Xcode (секция настроек General, поле Bundle Identifier).
- *App Services.* Здесь вам нужно отметить те сервисы, которые вы планируете использовать в вашем приложении. По умолчанию там отмечены только Game Center и In-App Purchase, их использование обязательно, удалить их нельзя. Остальные сервисы подключайте по мере необходимости.

После создания App ID вы можете использовать его для генерации любых типов профилей, об этом чуть позже.

Устройства (Devices)

В этом разделе размещено управление всеми устройствами Apple, которые вы можете использовать в рамках вашей Apple Developer Program. Есть ограничение, максимум 100 зарегистрированных девайсов одного типа (iPhone, iPad и так далее) на одну учетную запись в год, обычно более чем достаточно. При необходимости отладки на устройстве или выпуска Ad Hoc-сборки просто добавляйте сюда UDID нужных вам девайсов и используйте их при генерации профилей.

Профили (Provisioning Profiles)

Дословно название этого раздела переводится как «Профили обеспечения». Чуть более развернуто я бы описал понятие «профиль» как «Специальный файл, обеспечивающий доступ к некоторой функциональности в конкретной сборке вашего приложения». В данном разделе девцентра вы можете управлять вашими профилями, обеспечивая себе возможность выпускать сборки приложения для различных целей,

«профилировать» его. По сути, профиль является результатом объединения двух (иногда трех) компонентов:

- Активного сертификата определенного типа (раздел *Certificates*). С помощью сертификата профиль подтверждает, что ваше приложение имеет право на выполнение определенной группы действий.
- App ID (раздел *Identities*). Определяет конкретное приложение, для которого выпускается профиль.
- В некоторых случаях, еще нужен список зарегистрированных устройств (раздел *Devices*). Определяет перечень устройств, на которые разрешено устанавливать вашу сборку. Используется только с некоторыми типами профилей.

На выходе как раз и получаем профиль для выпуска сборок с определенными целями. Давайте рассмотрим разновидности профилей.

Профиль типа «Development»

Это профиль для разработки, то есть его основное назначение — отладка вашего приложения на конкретных устройствах через Xcode с подключением устройства проводом к вашему Mac. Дев-профили представлены двумя видами:

- *iOS App Development*. Требуется указания перечня разрешенных устройств из раздела *Devices*. Используется для отладки iOS-приложений.
- *tvOS App Development*. Аналогично, только используется для tvOS-приложений.

Профиль типа «Distribution»

Эти профили используются для выпуска сборок вашего приложения для различных целей. Продакшн-профили представлены четырьмя видами:

- *App Store*. Используется для тестирования (как внутреннего, так и внешнего) в TestFlight, а также для выпуска приложения в App Store.
- *tvOS App Store*. Аналогично предыдущему, только для tvOS.
- *Ad Hoc*. Требуется указания перечня разрешенных устройств из раздела *Devices*.
Используется, если вы хотите выпустить сборку, которую можно будет поставить в режиме «Production», но только на некоторых устройствах. Реальная ситуация, когда это может понадобиться, например, следующая. Вы разрабатываете приложение, а в процессе работы заказчик попросил у вас «дать ему пощупать приложение» на своем Apple-устройстве. В iTunes Connect для активации внешнего тестирования приложения не готовы, но просьбу заказчика нужно выполнять — вот тут как раз и пригодится Ad Hoc-профиль, сгенерированный на базе сертификата *App Store & Ad Hoc Production Certificate*. **Важный момент:** в моем случае часто возникали проблемы при экспорте сборки подобным способом, если в Xcode не был также установлен и Development-сертификат. Ошибки были разного рода, от невозможности подписать сборку до абсурдного «App ID is not available», хотя это фактически не так (замена на другой бандл ничего не давала). Поэтому моему предположению, для удачного экспорта Ad Hoc-сборок необходимо, чтобы, помимо Ad Hoc-профиля, был также установлен и дев-сертификат с соответствующим профилем.
- *tvOS Ad Hoc*. Аналогично предыдущему, только для tvOS.

Вкратце об iTunes Connect

Этот сервис предоставляет вам возможность управлять внутренним и внешним тестированием в TestFlight, а также выкладывать приложения в App Store. Рассмотрение этого процесса выходит за рамки данной статьи, упомяну лишь тот факт, что для корректной работы этому сервису необходимы сборки, созданные на базе профиля типа *Distribution* — *App Store* (для iOS либо tvOS). Другие типы профилей здесь не поддерживаются.

Резюмируем

По сути, при получении доступа к девцентру с активной Apple Developer Program ваш алгоритм действий должен сводиться к следующему:

1. Определиться, с каких конкретно машин будет производиться прямая отладка на устройствах через Xcode. Определить среди них основную машину (это может быть Mac разработчика, с которого чаще всего планируется производить отладку). Сгенерировать для основной машины сертификаты группы *Development*, скачать и установить их. По необходимости, экспортировать информацию об этих сертификатах в файл `*.p12/*.pfx`, которые потом можно будет разослать на другие целевые машины, где также планируется проводить отладку приложений.
2. Узнать, с какой машины планируется собирать сборки для тестирования и/или публикации в App Store. Сгенерировать для нее сертификаты группы *Distribution*. Повторить процедуру с экспортом из предыдущего пункта, если требуется поддержка нескольких машин.
3. Проконтролировать наличие нужного идентификатора приложения в разделе App IDs и соответствие указанного там бандла значению `Bundle Identifier` в проекте в Xcode, при наличии несовпадения — устранить его либо в девцентре, либо в Xcode (где именно это правило зависит от вашей конкретной ситуации).

4. Убрать (Revoke/Delete) все сертификаты, а затем и профили, которые обладают пометкой Expired (истекший сертификат) или Invalid (некорректный профиль). Также отмечу, что, в отличие от сертификатов, профили можно редактировать. То есть, сгенерировав новые сертификаты, вместо удаления старых профилей вы можете просто отредактировать их, указав им новые сертификаты в качестве под
5. Если профилей нет, либо не хватает нужных, то сгенерировать необходимые профили.
6. Скачать и установить нужные для вашей машины сертификаты и профили на свой компьютер. Установка производится двойным кликом файлу. Сертификаты будут установлены в Связку Ключей (Apple Keychain), профили — в Xcode.
7. Указать в настройках проекта Xcode нужные вам сертификаты в секции *Build Settings — Code Signing Identity — Development/Distribution* также указать необходимый *Provisioning Profile*.

На этом подготовка и чистка девцентра завершается. Далее вы можете выполнять любой из нижеследующих пунктов по необходимости:


- Произвести запуск в режиме отладки (Project — Run) через Xcode на разрешенном устройстве, используя дев-профиль.
- Создать сборку (Project — Archive с выбранным целевым устройством *Generic iOS Device*) на базе продакшн-профиля Ad Hoc для установки на конкретные устройства (такую сборку можно будет выслать, например, по электронной почте заказчику, чтобы он установил ее на свое разрешенное устройство).
- Создать сборку аналогично предыдущему пункту, но на базе продакшн-профиля App Store. Это будет сборка для внутреннего и/или внешнего тестирования, а также для выкладки в App Store, которую можно использовать в iTunes Connect.



Надеюсь, это подробное описание поможет вам разобраться в специфике функционирования системы сертификации приложений Apple, а поможет навести порядок в Apple Developer Center.

Теги: ios, ios development, ios разработка, ios programming, ios sdk, xcode, apple, apple store, app store, app store program development, apple developer program, apple developer center, apple development, objective-c


 +19 

 286  67k  16



 20,0 

Карма

 0,0

Рейтинг

16

Подписчики

[Написать](#) [Подписать](#)

Георгий Малюков @GDxRepo
iOS-программист

[Github](#)

ПОХОЖИЕ ПУБЛИКАЦИИ

4 сентября 2018 в 10:23

Как Яндекс создавал дополненную реальность в Картах для iOS. Опыт использования ARKit

 +44  6,7k  54  21

15 апреля 2018 в 20:13

Реактивное программирование в Objective-C

 +7  4,7k  31  7

18 февраля 2018 в 13:36

Способ управления цветовыми схемами приложения под iOS

 +11  3,9k  37  12

Комментарии 16

Отслеживать новые в ☐ почте ☐

 andrew8712 31 марта 2016 в 19:56  

«В первую очередь, нужно знать, что девелоперский сертификат всегда привязывается к одной конкретной машине.»
Вроде же можно экспортировать certificate + public key + private key в .p12 файл и потом импортировать на другой Mac.



Ответить



GDxRepo 31 марта 2016 в 19:56 # 📌 🔄

Да, можно, только вот запускать с этой машины в режиме отладки все равно будет нельзя. Сейчас p12 используется только для отправки сертификата на веб-сервер.

Ответить



DjPhoenixX 31 марта 2016 в 23:14 # 📌 🔄

Вот какую-то неправду вы говорите. Прямо вот сейчас используются у меня одни и те же сертификаты на домашнем компе, рабочем, и личном ноутбуке.

Ответить



GDxRepo 1 апреля 2016 в 02:03 # 📌 🔄

Почитал об этом, да, был неправ. Спасибо за поправку!

Ответить



modest_man 31 марта 2016 в 21:39 # 📌

Еще не указана возможность сделать In-House сборку приложения, которую можно будет устанавливать на любое устройство в обход App Store, и не привязывать UDID устройства в девцентре, но для этого надо иметь Enterprise Developer Program.

Ответить



GDxRepo 1 апреля 2016 в 02:16 # 📌 🔄

Я не сталкивался с Enterprise, поэтому об этом ничего не знаю))

Ответить



mish 1 апреля 2016 в 00:07 # 📌

Есть ограничение, максимум 100 зарегистрированных девайсов на одну учетную запись в год, обычно этого более чем достаточно

Справедливости ради, сейчас 100 девайсов одного типа (Apple TV, Apple Watch, iPhone, iPad, iPod Touch). Т.е., например, можно зарегистрировать 10 iPhone и 100 iPad.

Ответить



GDxRepo 1 апреля 2016 в 02:16 # 📌 🔄

Благодарю, внесу в статью!

Ответить



sacred 1 апреля 2016 в 02:02 # 📌

спасибо. сертификация у эпл одна из самых слабых сторон — как со стороны понимания всего этого, так и искусственные ограничения на запуск на девайсе

Ответить



mariavlasovamsk 1 апреля 2016 в 02:02 # 📌

Интересно...

Ответить



KursoRUS 1 апреля 2016 в 02:02 # 📌

Для работы с сертификатами и профилями еще есть очень удобные утилиты (sign и cert) из fastlane (<https://fastlane.tools/>). После небольшой настройки позволяют практически не заходить самому в dev center и itunes connect.

Ответить



alex1covo 2 апреля 2016 в 17:39 # 📌

Правильно ли я понял, что сборки с сертификатом «Development» / «iOS App Development» нельзя залить в testflight?

Просто мне не понятна принципиальная разница на этапе тестирования в сборках с сертификатами «Development» / «iOS App Development» и «Production» / «App Store and Ad Hoc»

И для первого и для второго типов сертификатов в Provisioning Profiles можно указать девайсы на которых это приложение запустится.

Я создавал сборки с обоими типами сертификатов и одинаково удачно запускал их на устройствах.

Дело в том, что я использую rhonedar, без Xcode. Может быть в Xcode какой-то вид сертификата дает преимущество по сравнению с другим?

Ответить



Ответить



Ответить



Ответить



Ответить

B / U ~~S~~ “ ” *

Отправить



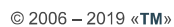
Приложения

Загрузите в
App Store



ДОСТУПНО В
Google

Конфиденциальность



О сайте

Служба поддержки

[Мобильная версия](#)