

Добавление собственного сертификата для всего приложения на iOS

[Архив для скачивания:](#)

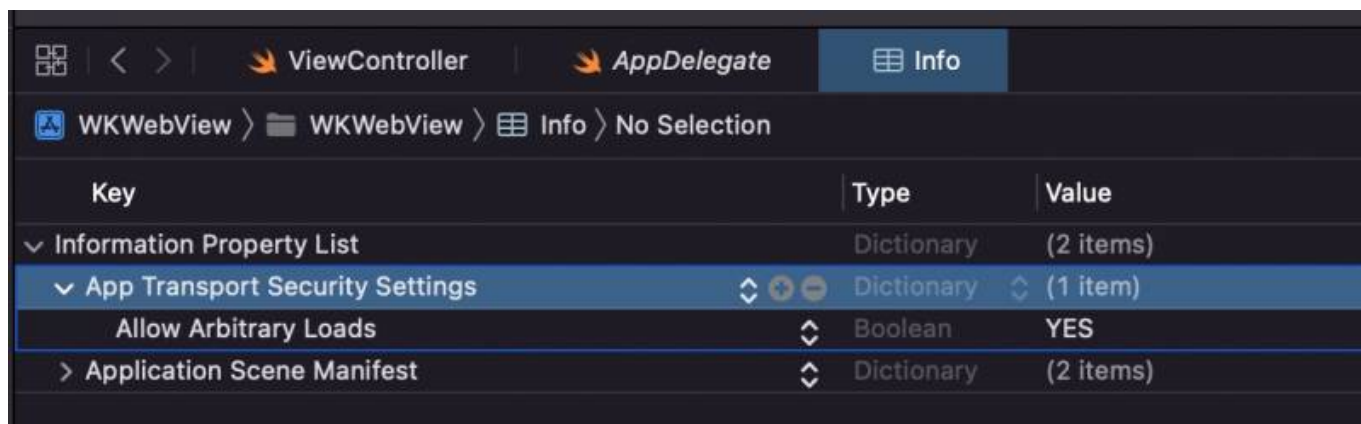
wkwebview_.zip

Для возможности открытия Web View с сертификатами Минцифры, чтобы сохранить работоспособность сервисов sberbank.ru, вам необходима доработка в своем проекте webView:

- установить публичный сертификат НУЦ Минцифры в качестве доверенного в коде вашего приложения

Перед тем как приступить к доработке Web View необходима конфигурация в основном файле Info.plist проекта.

Есть возможность самостоятельно управлять настройками сетевого соединения, посредством конфигурации App Transport Security Settings в основном файле Info.plist проекта. Необходимо добавить данную конфигурацию в Info.plist файл с ключом Allows Arbitrary Loads и значением YES (Скрин 1):



Скрин 1

Установка ключа AllowsArbitraryLoads со значением YES означает, что система не будет запрещать приложению взаимодействовать с любыми доменами по протоколу HTTP.

Важно! iOS 10.0 и более старшие версии:

ATS игнорирует установленное значение NSAllowsArbitraryLoads, если какой-либо из следующих ключей присутствует в конфигурации App Transport Security Settings:

NSAllowsArbitraryLoadsForMedia
NSAllowsArbitraryLoadsInWebContent
NSAllowsLocalNetworking

Более подробно о [NSAllowsArbitraryLoads](#).

Также важный момент с AllowsArbitraryLoads является прохождение ревью в AppStore. В документе [NSAllowsArbitraryLoads](#) указано уточнение, что если выставить значение YES, необходимо будет обоснование на ревью.

Пример:

- Обоснованием может являться то, что у вас используются самоподписанные сертификаты на серверах и можно добавить, что проводите валидацию по записанным сертификатам в приложении, тем самым обеспечиваете безопасное соединение.

По факту данное обоснование Apple не всегда спрашивают (даже скорее не спрашивают).

Для примера как провести в webView установку публичного сертификата НУЦ Минцифры в качестве доверенного в коде вашего приложения был создан проект WKWebView, который включает сущности, используемый в нашем приложении МП СБОЛ.

```
14 override func viewDidLoad() {
15     super.viewDidLoad()
16     settingLayout()
17
18     /// Нужно определить navigationdelegate у webview
19     webView.navigationDelegate = self
20
21     let sber = "https://securepayments.sberbank.ru/"
22     let url = URL(string: sber)!
23     webView.load(URLRequest(url: url))
24     webView.allowsBackForwardNavigationGestures = true
25 }
```

Скрин 2

Настоятельно рекомендуется использовать URLRequest без модификации у запроса переменной cachePolicy (рекомендуем не отключать кэш, оставить дефолтное значение .useProtocolCachePolicy неизменным), пример с инициализацией URLRequest(url: url) (Скрин 2).

P.S. На данный момент, к сожалению, нет точного ответа по какой причине не проходит валидация сертификатов при модификации cachePolicy у запроса.

Для удобства и избыточности все основные пояснения изложены в комментариях к коду в проекте WKWebView. Здесь приводится (Скрин 3) метод делегата WKNavigationDelegate webView didReceive challenge, перед этим определив navigationdelegate у webview. Также был скачан и предустановлен в проект сертификат Минцифры (преобразован с расширением .der).

```

» /// Реализовать метод делегата WKNavigationDelegate~
» func webView(_ webView: WKWebView, didReceive challenge: URLAuthenticationChallenge,
» completionHandler: @escaping URLSession.AuthChallengeDisposition, URLCredential?) -> Void) {-
»
» /// Добавить сертификат Минцифры в папку Certificates~
» /// скачать можно с https://www.gosuslugi.ru/crt~
» /// в папке Certificates представлен данный сертификат Минцифры~
» /// также можно добавить дополнительные доверенные сертификаты~
»
» /// Получить массив сертификатов в формате Data~
» let data = Config.certificates("Certificates")~
»
» /// Установки дополнительных доверенных сертификатов в WebView~
» AsyncAuthChallengeHandler.webViewAddTrusted(certificates: data).handle((challenge,
» completionHandler))~
»
» }~
}~

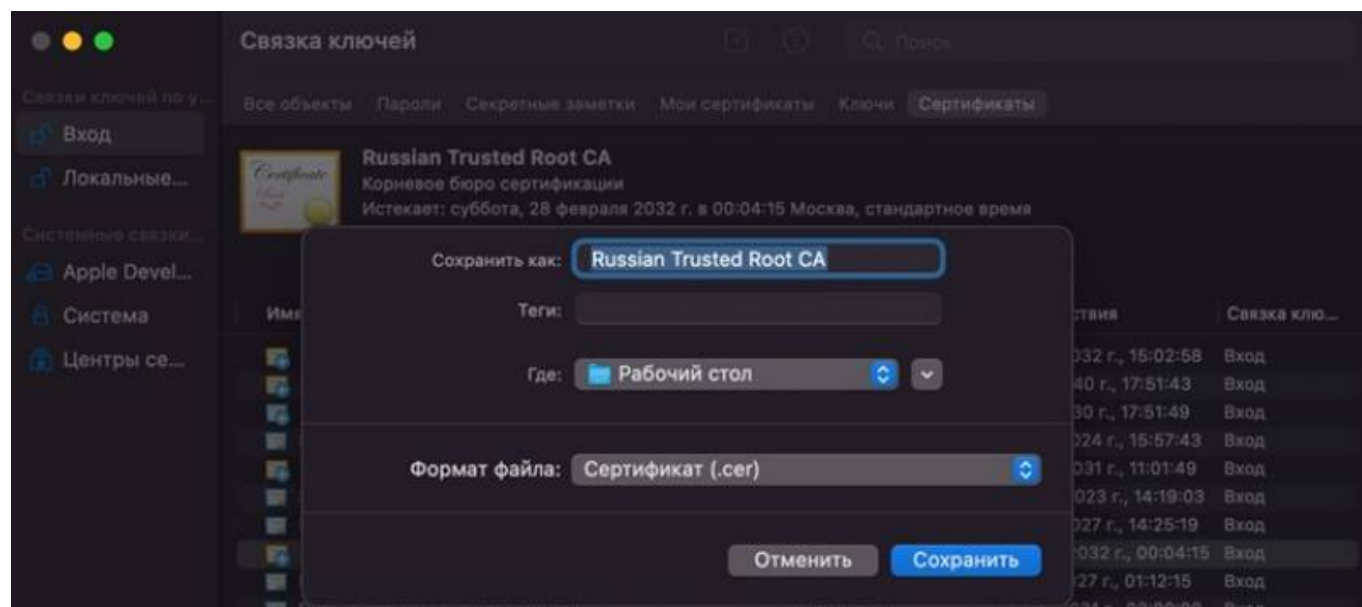
```

Скрин 3

Скачать действующие сертификаты Минцифры можно с сайта госуслуг <https://www.gosuslugi.ru/crt> в разделе «Сертификаты для MacOS».

Пример с сертификатом Russian Trusted Root CA:

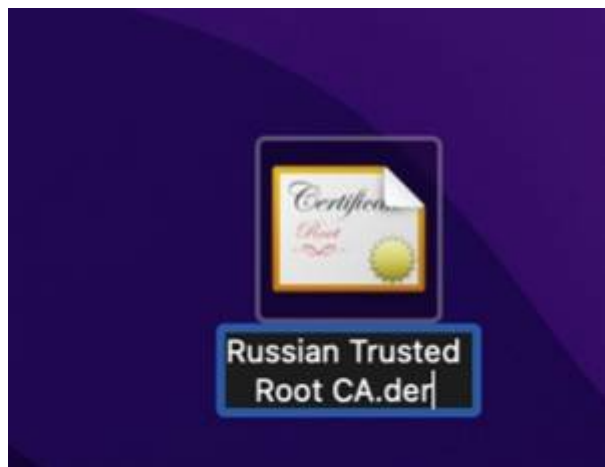
После загрузки экспортировать из связки ключей в любую удобную папку (Скрин 4).



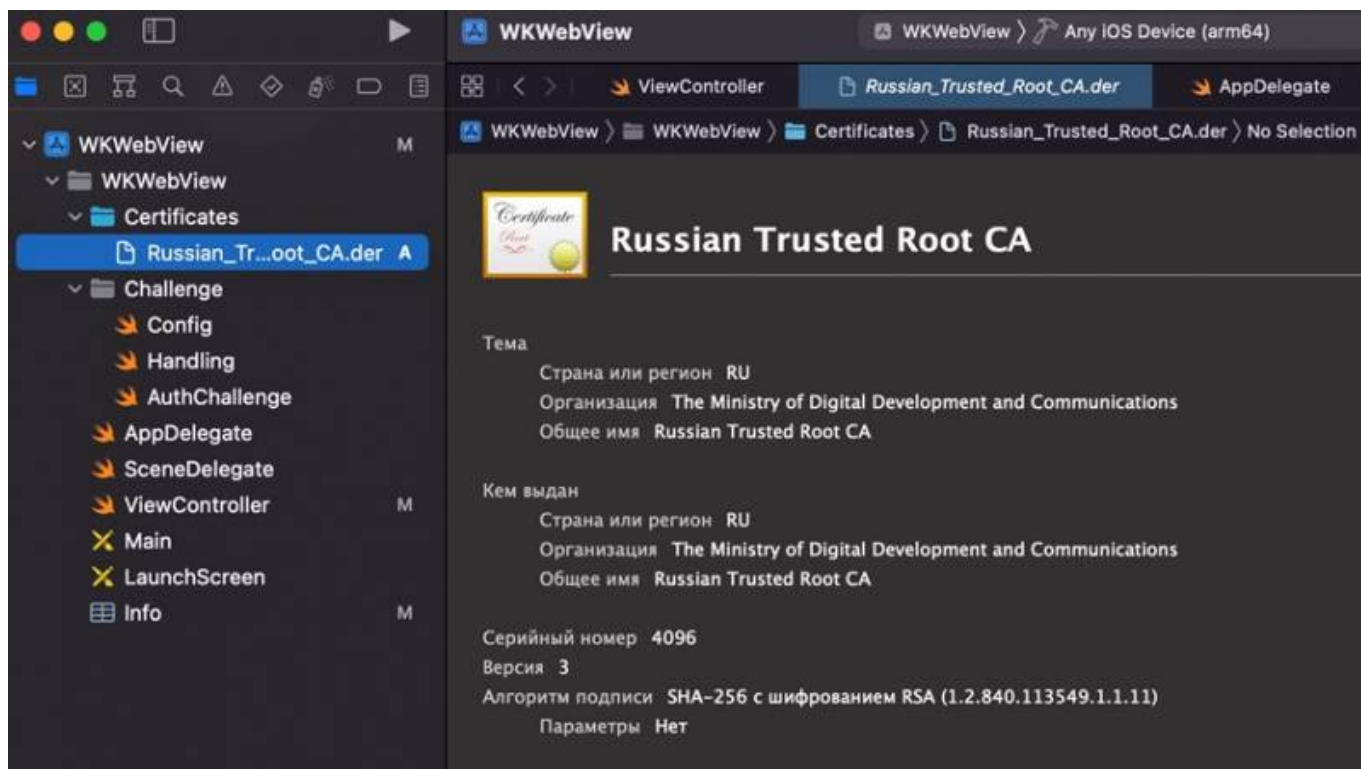
Скрин 4

Изменить расширение на .der (Скрин 5) и добавить его в папку Certificates.

Расширение .der необходимо для парсинга сертификата Xcode'ом и корректного отображение информации о сертификаты в нем (Скрин 6).



Скрин 5



Скрин 6

Сам проект лучше всего проверять на реальном устройстве (см. Архив).