самоподписанный сертификат - как его сделать и получить и подгрузить в хранилище сертификатов Ocelot? нужна подробная инструкция

**ОТВЕТ:**

Хорошо, вот пошаговая инструкция по созданию, получению и загрузке самоподписанного сертификата в хранилище сертификатов Ocelot:

**1. Создание самоподписанного сертификата:**

* **Используйте OpenSSL:** OpenSSL — это стандартный инструмент для работы с сертификатами. Выполните следующие команды в терминале:

openssl genrsa -out server.key 2048 # Создает закрытый ключ

openssl req -new -key server.key -out server.csr # Создает запрос на сертификат

openssl x509 -req -in server.csr -signkey server.key -out server.crt -days 365 # Подписывает запрос и создает сертификат

content\_copy Use code [with caution](https://support.google.com/legal/answer/13505487).Bash

* **Объяснение команд:**
  + genrsa: Создает закрытый ключ с размером 2048 бит.
  + req: Создает запрос на сертификат (CSR) с использованием ранее созданного ключа. Вам будет предложено ввести информацию о сертификате, такую как название организации, страну и т.д.
  + x509: Подписывает запрос на сертификат и создает самоподписанный сертификат с использованием закрытого ключа.

**2. Получение сертификата:**

* **Найдите созданный сертификат:** Сертификат будет сохранен в файле server.crt.
* **Конвертируйте в формат PEM:** Если ваш сертификат в формате .pem, пропустите этот шаг. В противном случае, конвертируйте сертификат в формат PEM:

openssl x509 -in server.crt -outform PEM -out server.pem

content\_copy Use code [with caution](https://support.google.com/legal/answer/13505487).Bash

**3. Загрузка сертификата в хранилище Ocelot:**

* **Найдите каталог хранилища Ocelot:** По умолчанию хранилище сертификатов Ocelot находится в каталоге %LOCALAPPDATA%\Ocelot\Certificates.
* **Создайте новый каталог:** Создайте новый каталог с именем вашего бэкенд-сервиса (например, MyBackendService).
* **Скопируйте сертификат:** Скопируйте файл server.pem в созданный каталог.
* **Настройте конфигурацию Ocelot:** В файле Ocelot.json добавьте следующую конфигурацию:

{

"GlobalConfiguration": {

// ... other settings ...

},

"ReRoutes": [

{

// ... other settings ...

"DelegatingHandlers": [

{

"Type": "Ocelot.DownstreamHttpProvider.Handlers.HttpClientHandler",

"Properties": {

"ServerCertificateCustomValidationCallback": "MyCustomValidationCallback"

}

}

]

}

]

}

content\_copy Use code [with caution](https://support.google.com/legal/answer/13505487).Json

* **Создайте метод MyCustomValidationCallback:** В вашем приложении добавьте метод MyCustomValidationCallback с такой реализацией:

public static bool MyCustomValidationCallback(object sender, X509Certificate certificate, X509Chain chain, SslPolicyErrors sslPolicyErrors)

{

// Проверяем, что сертификат соответствует имени хоста бэкенд-сервиса

if (certificate.Subject.Contains("CN=your-backend-hostname"))

{

return true;

}

return false;

}

content\_copy Use code [with caution](https://support.google.com/legal/answer/13505487).C#

**4. Перезапустите Ocelot:** Перезапустите Ocelot, чтобы изменения вступили в силу.

**Дополнительно:**

* **Настройка TrustedRootIssuers:** Если вам нужно добавить самоподписанный сертификат в список доверенных корневых сертификатов, вы можете использовать System.Security.Cryptography.X509Certificates.X509Certificate2Collection в .NET Framework или System.Security.Cryptography.X509Certificates.X509Certificate2 в .NET Core.
* **Установка переменной окружения:** Вместо копирования сертификата вручную, вы можете установить переменную окружения OCELOT\_CERTIFICATE\_STORE в каталог с сертификатом.

**Важно:**

* **Не используйте этот подход в продакшене:** Самоподписанные сертификаты не считаются безопасными в продакционной среде.
* **Используйте сертификаты от центра сертификации:** В продакшне всегда используйте сертификаты от центра сертификации (например, Let's Encrypt).
* **Безопасность:** Будьте осторожны с обработкой и хранением вашего закрытого ключа! Не размещайте его в открытом доступе.

Надеюсь, эта инструкция поможет вам! Если у вас возникнут вопросы, не стесняйтесь спрашивать!

