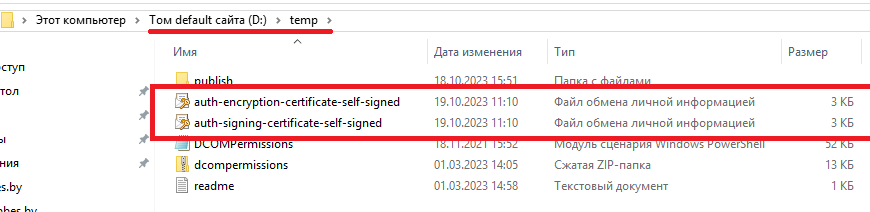
Настройка сертификатов Сервера авторизации на библиотеке OpenIdDic.

# Генерация сертификатов для сервера авторизации

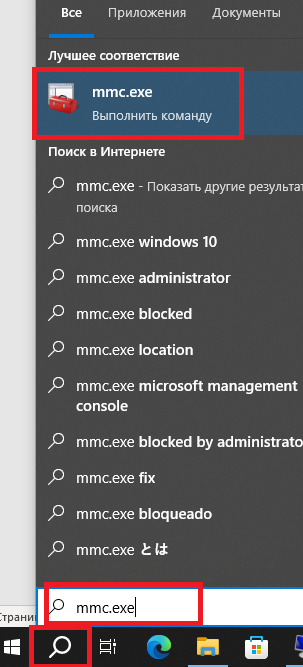
1. Подготовить «OpenIdDictCert»-утилиту для генерации. Например, для "auth.rupbes.by" сервера код будет выглядеть как на картинке ниже. Необходимо указать пароль для сертификатов. В нашем примере – это "Qq?01011967"

|  |
| --- |
| CreteCertAndCertRequest(X509KeyUsageFlags.KeyEncipherment,  "auth.rupbes.by", "rupbes.by Encryption Certificate", "BY",  "auth-encryption-certificate-request.der", "auth-encryption-certificate-request.pem",  "Self Signed Encryption Certificate for auth.rupbes.by",  "auth-encryption-certificate-self-signed.pfx", "Qq?01011967");  CreteCertAndCertRequest(X509KeyUsageFlags.DigitalSignature,  "auth.rupbes.by", "auth.rupbes.by Signing Certificate", "BY",  "auth-signing-certificate-request.der", "auth-signing-certificate-request.pem",  "Self Signed Signing Certificate for auth.rupbes.by",  "auth-signing-certificate-self-signed.pfx", "Qq?01011967"); |

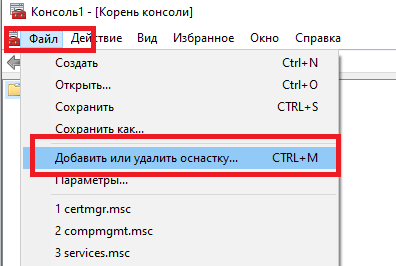
1. Запустить утилиту и сгенерировать файлы. В нашем примере нам понадобятся два:
   1. "auth-signing-certificate-self-signed.pfx"
   2. "auth-encryption-certificate-self-signed.pfx"
2. Перенести файлы на сервер с IIS в любую папку. К примеру, “d:\temp”



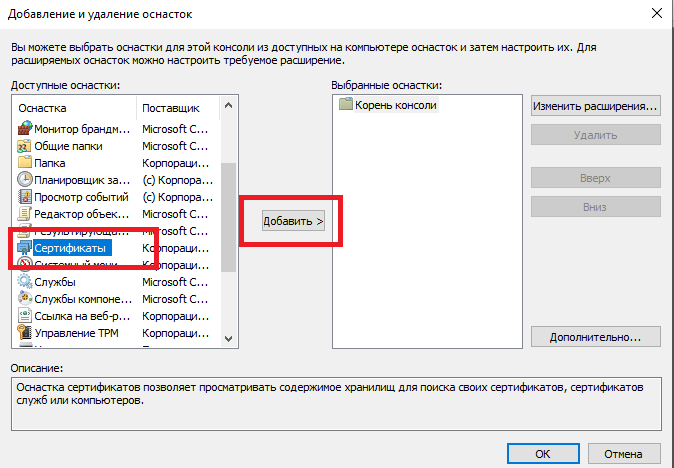
1. На сервере с IIS запустить MMC.EXE



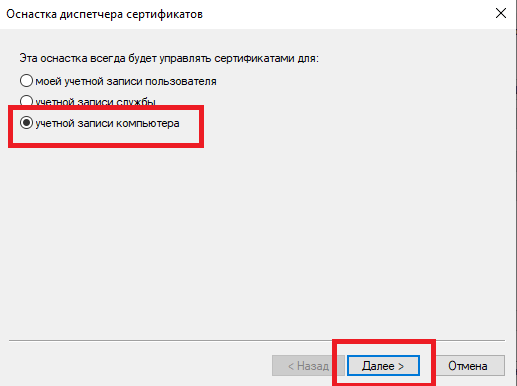
1. Выбрать пункт меню “Файл/Добавить или удалить оснастку”



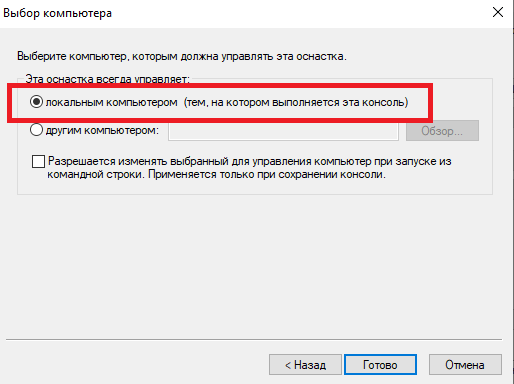
1. В блоке диалога оснасток выбрать “Сертификаты” и кликнуть “Добавить”



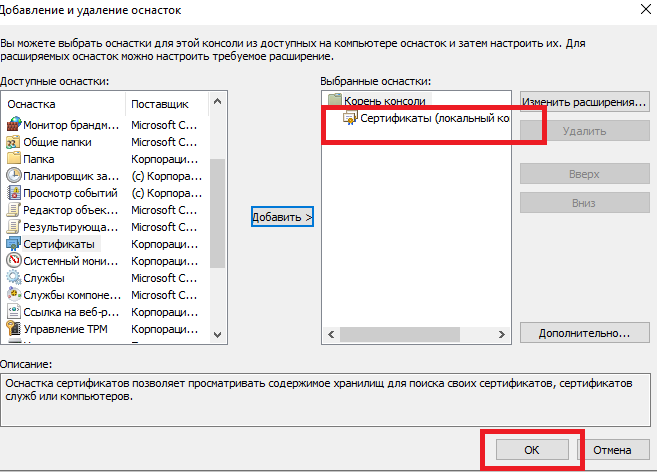
1. В появившемся блоке диалога выбрать “учетной записи компьютера” и кликнуть далее



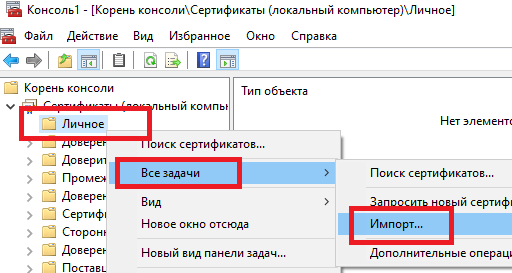
1. В следующем блоке выбрать “Локальный компьютер” и кликнуть “готово”



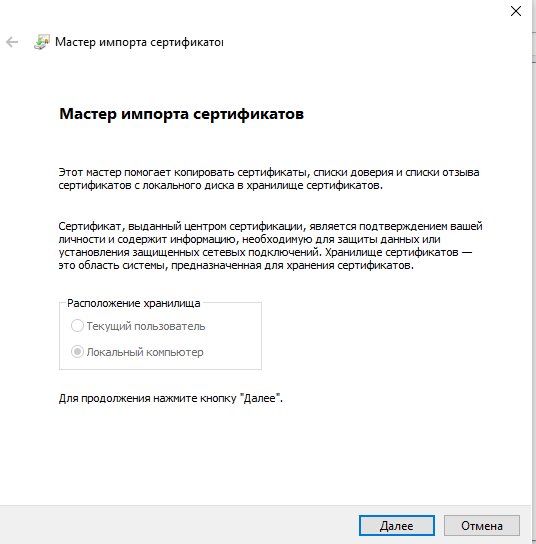
1. Вы вернулись к блоку “Добавления оснасток”. Кликните “OK”



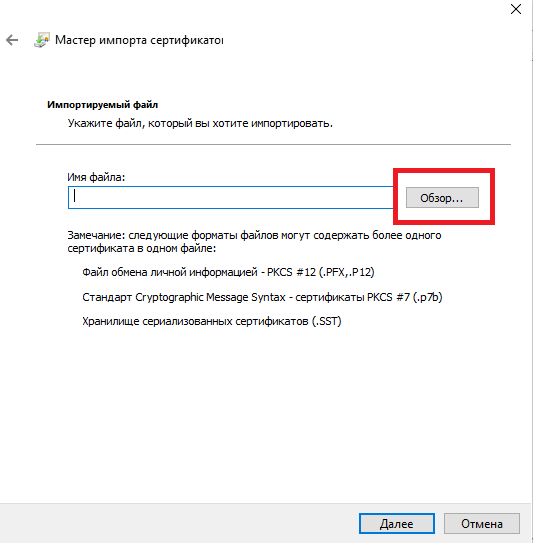
1. В Mmc.exe выбрать “Сертификаты/Личное/Все задачи/Импорт”



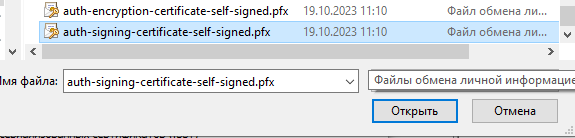
1. В мастере импорта кликнуть “Далее”



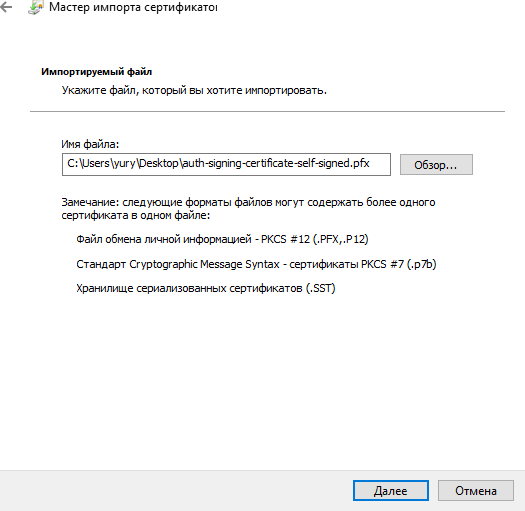
1. На следующей вкладке кликнуть “Обзор”



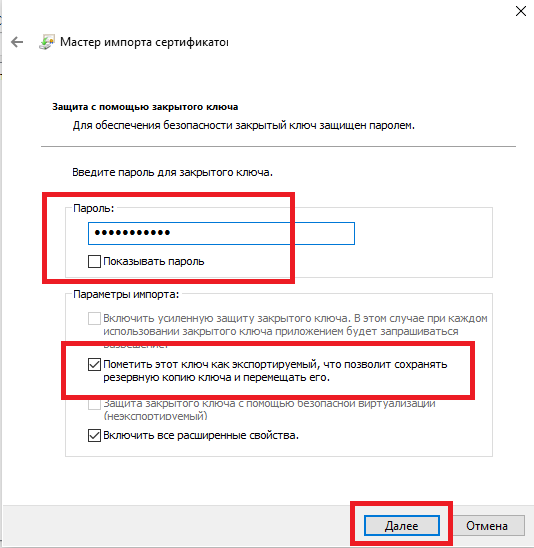
1. В появившемся блоке выбрать первый файл сертификата и нажать “Открыть”



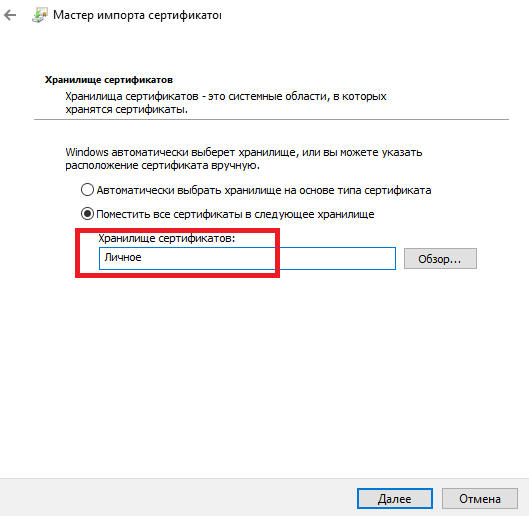
1. Это вернет на шаг 12. Нажать “далее”



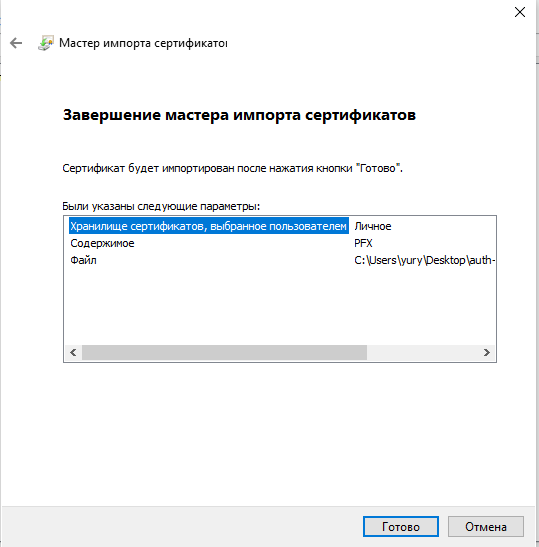
1. В следующем приглашении вводим пароль и влючаем флаги как на картинке ниже. Нажимаем “Далее”. **Пароль вводим тот, который указан на шаге 1**. (Для каждого сертификата лучше иметь свой пароль.)



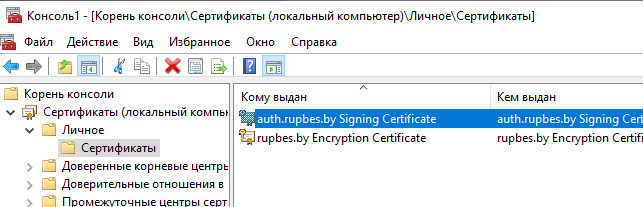
1. На следующем диалоге подтвердить “Личное” хранилище. **Ни в коем случае не выбирать другое хранилище!!!** Личное хранилище имеет предопределенный код, который прописан в прикладном ПО.



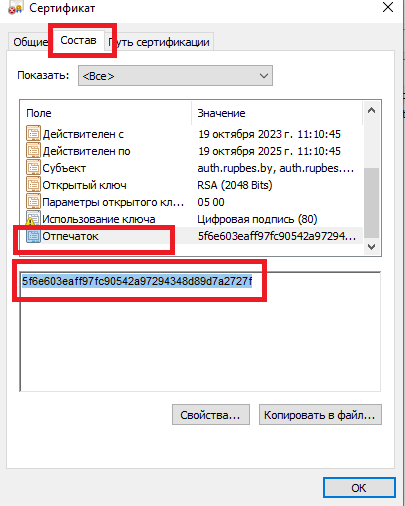
1. На следующем диалоге кликаем “Готово”



1. Для каждого (в нашем случае из двух) импортированного сертификата двойным щелчком мыши вызвать свойства сертификата



1. На вкладке “Состав”, выбрать “Отпечаток” и скопировать в текстовый документ. Отпечаток понадобится для настройки сервера авторизации.



1. По каждому сертификату вызвать пункт меню “Управление закрытыми ключами”



1. В блоке диалога “Безопасность” добавить
   1. “**IIS\_IUSRS**”. В нашем случае – это “WEBX-SRV\IIS\_IUSRS”.
   2. Или добавить пул приложения В нашем случае это “**IIS AppPool\auth.rupbes.by**”





1. Отредактировать “appsettings.json” как на фрагменте ниже. Где, “xxxStoreName": 5” это код “Личное” (см. шаг 10). “XXXStoreLocation": 2” это хранилище сертификатов локального компьютера (см. шаг 7). “XXXThumbprint": "….”” это отпечаток (см. шаг 19)

|  |
| --- |
| "Oidc": {  "UseReferenceTokens": {  "RefreshTokens": true,  "AccessTokens": true  },  "TokenEncryption": {  "DisableAccessTokenEncryption": true,  "EncryptionCertificateStoreLocation": 2,  "EncryptionCertificateStoreName": 5,  "EncryptionCertificateThumbprint": "3eceb436f0628f99b45a3176cba192705d214e9a",  "SigningCertificateStoreLocation": 2,  "SigningCertificateStoreName": 5,  "SigningCertificateThumbprint": "38d4230c19369d45c4c2266a79e3850aeffa28aa"  },  "TokenLifetime": {  "AccessTokenLifetimeFromMinutes": 30,  "IdentityTokenLifetimeFromMinutes": 30,  "RefreshTokenLifetimeFromMinutes": 240  }  }, |

1. Работа с сервером БД=MSSQL. Перед запуском сервера авторизации необходимо настроить доступ к БД. Убедитесь, что в файле “appsettings.json” указан “"UseMsSql": true” и “"DefaultConnection":”

|  |
| --- |
| "UseMsSql": true,  "ConnectionStrings": {  "DefaultConnection": "Server=DBSX;Database=OpenIdDictMvcSvr08;Persist Security Info=True;User ID=sa;Password=01011967;TrustServerCertificate=True;"  }, |

24. Перед запуском сервера авторизации необходимо создать базу данных и наполнить ее учетными записями администратора и менеджера. Для этого использовать утилиту “OpenIdDictDbMigrator”. После работы OpenIdDictDbMigrator будут созданы две учетные записи

“admin@gmail.com/ Qwerty123$” и “manager@gmail.com/ Qwerty123$”.

При запуске “OpenIdDictDbMigrator” необходимо указать в “appsettings.json” файле строку соединения и указать, что будем работать с MsSql

|  |
| --- |
| "UseMsSql": true,  "ConnectionStrings": {  "DefaultConnection": "Server=DBSX;Database=OpenIdDictMvcSvr08;Persist Security Info=True;User ID=sa;Password=01011967;TrustServerCertificate=True;"  }, |