



# Модуль интеграции с УЦ Контур

Версия и дата сборки: 5.5.1, 2025-01-17

*Модуль интеграции с УЦ Контур* является дополнительным модулем Docsvision и предназначен для регистрации заявок на выпуск сертификатов в удостоверяющем центре (УЦ) СКБ Контур.

Модуль представляет собой серверное расширение для модуля Web-клиент и позволяет выпускать сертификаты в удостоверяющем центре Контур с использованием API УЦ Контур.

Модуль поддерживает работу через Web-клиент.

# Назначение и функции модуля

Модуль позволяет выпускать сертификаты с подтверждением через ЕСИА или СМС, а также предоставляет методы API, позволяющие обращаться к УЦ Контур. API модуля предоставляет следующие возможности:

- Создать заявку на выпуск сертификата.
- Создать или изменить документ в заявке.
- Сгенерировать шаблон документа заявки.
- Загрузить одну или несколько страниц документа заявки.
- Отправить заявку на проверку.

Модуль интеграции с УЦ Контур добавляет в стандартные диалоги подписания возможность подписывать документы сертификатом облачной подписи, добавленным в профиль пользователя через диалог выбора сертификата.

# Разделы документации

*Документация модуля содержит следующие разделы:*

1. Общие сведения о модуле.
2. Администрирование модуля.
3. Руководство пользователя при работе в Web-клиенте.
4. Руководство разработчика Модуля интеграции с УЦ Контур.

# /dv5/kontur-integration/5.5.1/[Общие сведения о модуле]

## Структура модуля

Модуль интеграции с УЦ Контур добавляет в систему методы API, дающие возможность взаимодействия с УЦ Контур.

### Компоненты модуля

*Модуль состоит из следующих компонентов:*

- a. Серверные компоненты модуля интеграции с УЦ Контур.
- b. Специальное расширение Web-клиента, добавляющее поддержку работы с Модулем интеграции с УЦ Контур.

### Сведения о связях с другими системами

*Модуль интеграции с УЦ Контур является промежуточным звеном в связи системы Docsvision с оператором УЦ Контур.*

Модуль предоставляет публичные интерфейсы, которые используются системой Docsvision для вызова функций интеграции с УЦ Контур. Описание API модуля приведено в разделе руководстве разработчика модуля: "[Разработка компонентов модуля](#)".

## Необходимые ресурсы

### Необходимое техническое обеспечение

Специальные требования к аппаратному обеспечению не предъявляются.

### Необходимое программное обеспечение

- СКЗИ КриптоПро CSP версии 5.0 и выше должно быть установлено на клиентских машинах (если используется Windows-клиент) и сервере Web-клиента. См. подробнее требования в документации модуля [/dv5/backoffice/5.5.5/requirements/#crypto-pro](#)[Базовые объекты].
- [КриптоПро .NET Сервер](#) должно быть установлено на сервере Web-клиента.

### Требования к каналам связи

Администратор локальной сети должен обеспечить возможность подключения с

сервера Docsvision и серверов СУБП к сервису УЦ Контур.

- Стандартная точка подключения: <https://api.kontur.ru/kcr>.
- Адрес тестового сервера: <https://kcr-api.testkontur.ru>

## **Формирование и проверка подписи**

Корневые сертификаты СКБ Контур должны быть установлены в хранилище *Доверенные корневые центры сертификации* на уровне *Локальный компьютер*.

## **Требования к системе Docsvision**

### **Модули Docsvision**

1. Серверная и клиентская части модуля *Платформа* версии 5.5.5 и выше.
2. Серверная и клиентская части модуля *Web-клиент* версии 5.5.17 и выше для работы через web-интерфейс.
3. Серверная и клиентская части модуля *фоновых операций* версии 5.5.2, сборка 118 и выше.

## **Лицензия**

*Для работы с модулем Модуль интеграции с УЦ Контур:*

- Лицензия на систему Docsvision должна содержать дополнительную опцию *Docsvision Модуль интеграции с УЦ Контур*.
- В зависимости от используемой подписи<sup>[1]</sup> в документации модуля Web-клиент.] на стороне клиента может потребоваться лицензия на компоненты КриптоПро CSP.

## **Прочие требования**

1. Учётная запись в системе УЦ Контур.
2. Для создания, отправки, проверки сертификатов и заявок на сертификаты в УЦ Контур необходимо быть зарегистрированным пользователем данной системы. Регистрация организации в системе "УЦ Контур" выполняется самостоятельно.

Для создания и проверки заявок в УЦ Контур, добавления документов, создания шаблонов и других функций требуется подключение к "УЦ Контур" с тарифом, включающим оплату доступа к API. Поддерживается два режима работы:

- С использованием собственного личного кабинета УЦ Контур компании. Для этого требуется прямой договор с СКБ Контур, индивидуальный API-Key и TLS.



Информация о подключении к "УЦ Контур" приведена на сайте <https://ca.kontur.ru/>.

[1] См. раздел "/dv5/webclient/5.5.17/requirements-signature/[Формирование и проверка подписи

# Изменения, обновления и исправленные ошибки

## Модуль интеграции с УЦ Контур

### Общая документация

Общая информация об изменениях, исправленных ошибках и накопительных обновлениях.

- [Изменения в релизной версии](#)
- [Исправленные ошибки](#)
- [Накопительные обновления](#)

## Изменения в релизной версии

Выпущен Модуль интеграции с УЦ Контур.

## Исправленные ошибки

## Накопительные обновления

# Администрирование Модуля интеграции с УЦ Контур

## Установка и удаление модуля

### Требования

Ознакомьтесь с требованиями в разделе "[Необходимые ресурсы](#)".

### Состав установочного комплекта

1. Пакет установки серверной части модуля: `Docsvision 5 CA Kontur integration.msi`.

### Установка серверной части модуля

Серверная часть *Модуля интеграции с УЦ Контур* устанавливается на сервер Docsvision, сервер Web-клиент и СУБП. Установка должна выполняться от имени пользователя, являющегося локальным администратором и администратором Docsvision.



Администратор Docsvision — пользователь, включенный в группу безопасности **DocsVision Administrators** на компьютере с сервером Docsvision.

1. Запустите пакет установки `Docsvision 5 CA Kontur integration.msi`.

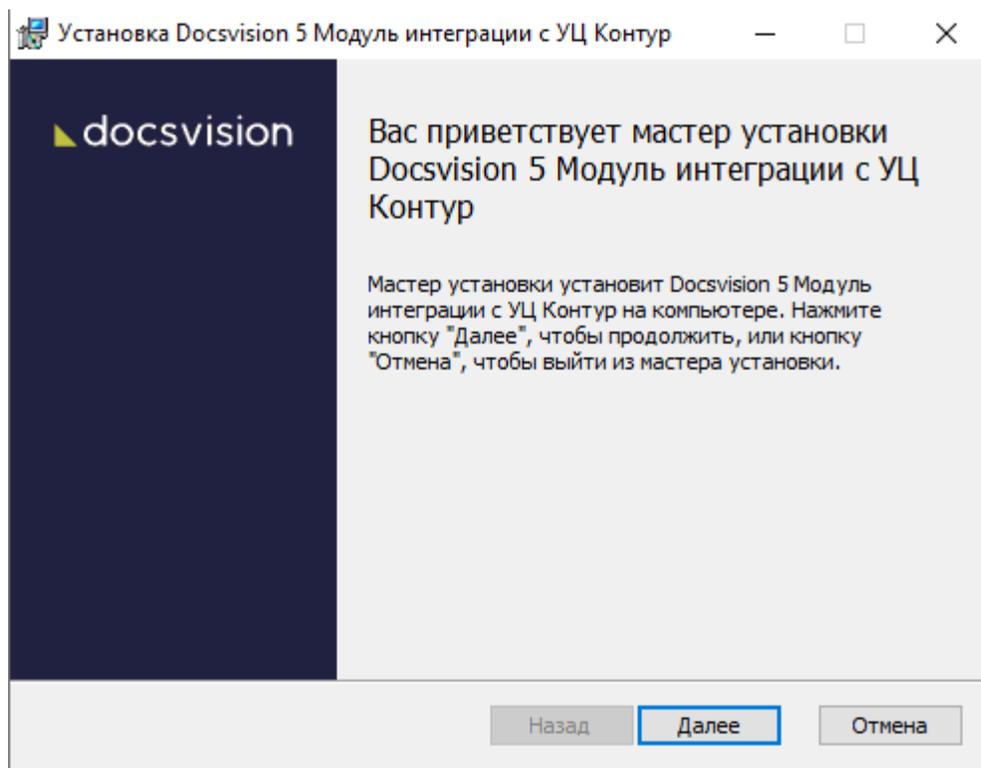


Рисунок 1. Мастер установки серверной части модуля

2. Примите условия лицензионного соглашения для продолжения установки.

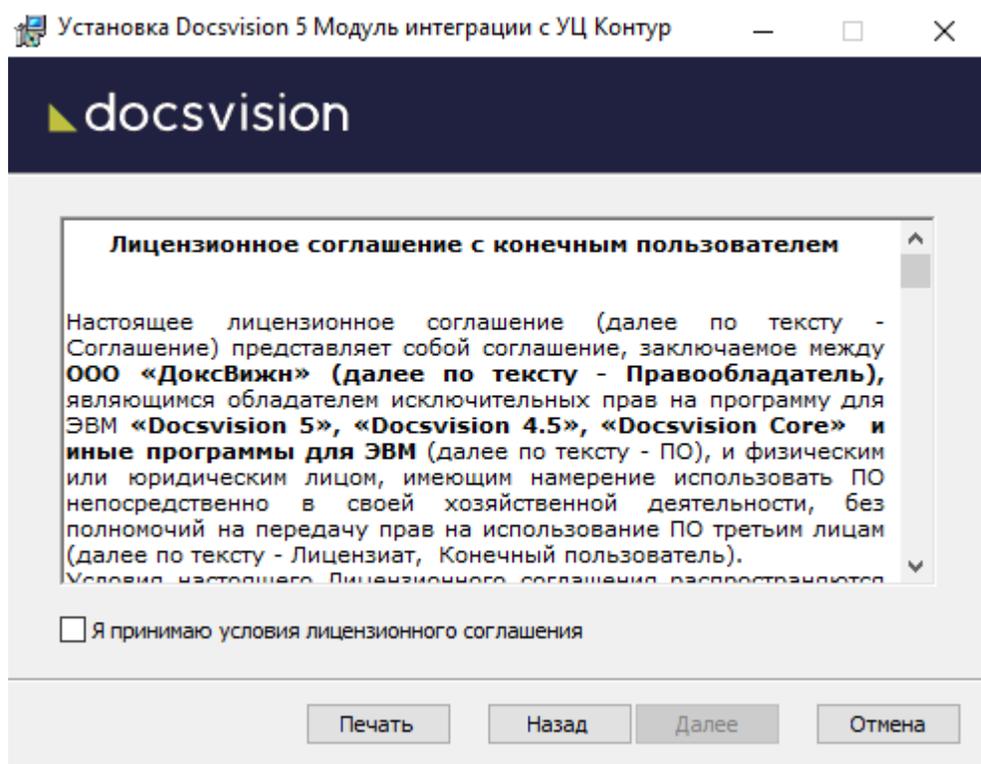


Рисунок 2. Условия лицензионного соглашения

3. Если требуется, измените набор устанавливаемых компонентов.

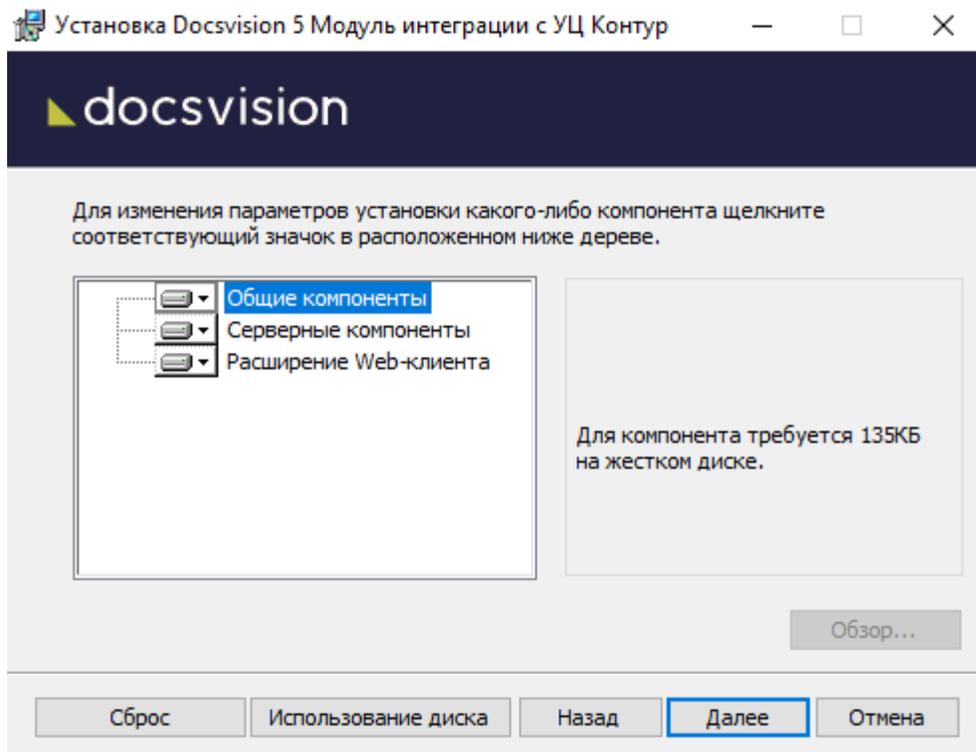


Рисунок 3. Устанавливаемые компоненты модуля

- *Общие компоненты* — обязательные для установки компоненты модуля, обеспечивающие базовую функциональность. Отображаются всегда.
- *Серверные компоненты* — основные серверные компоненты модуля обеспечивающие связь с сервером Docsvision.



Компоненты недоступны для установки, если не установлен модуль Платформа [требуемой версии](#).

- *Расширение Web-клиента* — компоненты, необходимые для работы Web-клиента. Обязательны к установке, неотключаемы.



Расширение недоступно для установки, если не установлен модуль Web-клиент [требуемой версии](#).

- Кнопка **Сброс** — сбрасывает выбор компонентов на стандартные.
- Кнопка **Использование диска** — позволяет посмотреть свободное место на системных дисках.

4. В следующем окне нажмите **Установить** и дождитесь завершения установки.

5. Нажмите на кнопку **Готово**.

6. Запустите "Консоль настройки Docsvision" и завершите настройку Docsvision стандартным образом согласно документации модуля Платформа /dv5/platform/5.5.5/admin/config-master/[здесь] и /dv5/platform/5.5.5/admin/post-config-server/[здесь].

Пользователь, от имени которого запускается "Консоль настройки Docsvision" должен являться администратором Docsvision, а также входить в группы Docsvision в Справочнике сотрудников: *Администраторы УД, Администраторы КС и Системная для WF.*

7. Выполните необходимые настройки модуля, описанные в разделе [Настройка модуля интеграции с УЦ Контур](#).

## Удаление модуля

1. Удалите серверный компонент модуля *Модуль интеграции с УЦ Контур*.

## Обновление модуля

Список изменений в текущей версии см. здесь: [Изменения в релизной версии](#).

### Условия обновления

- Установка новой версии должна выполняться в нерабочее время.
- Создайте резервную копию БД Docsvision.
- База данных Docsvision должна быть переведена на работу с расширенными метаданными.
- Если к новой версии модуля приложена инструкция по обновлению, следуйте ей.
- Если комплект обновления включает обновление для модуля Платформа, перейдите к инструкции, приведенной в /dv5/platform/5.5.5/admin/update-module/[документации] модуль Платформа. Руководство администратора, пункт "Установка новой версии модуля".

### Порядок обновления

*Чтобы установить новую версию модуля:*

1. Создайте резервную копию БД.
2. Остановите все сервисы Docsvision. Также остановите экземпляры Службы фоновых операций и все сервисы в кластере Docsvision или СУБП, если таковые используются.

3. Установите новую версию модуля из пакета установки, не удаляя существующей версии.

При обновлении используется каталог и область установки текущей версии.

4. Запустите остановленные сервисы Docsvision, затем нажмите **Готово**.

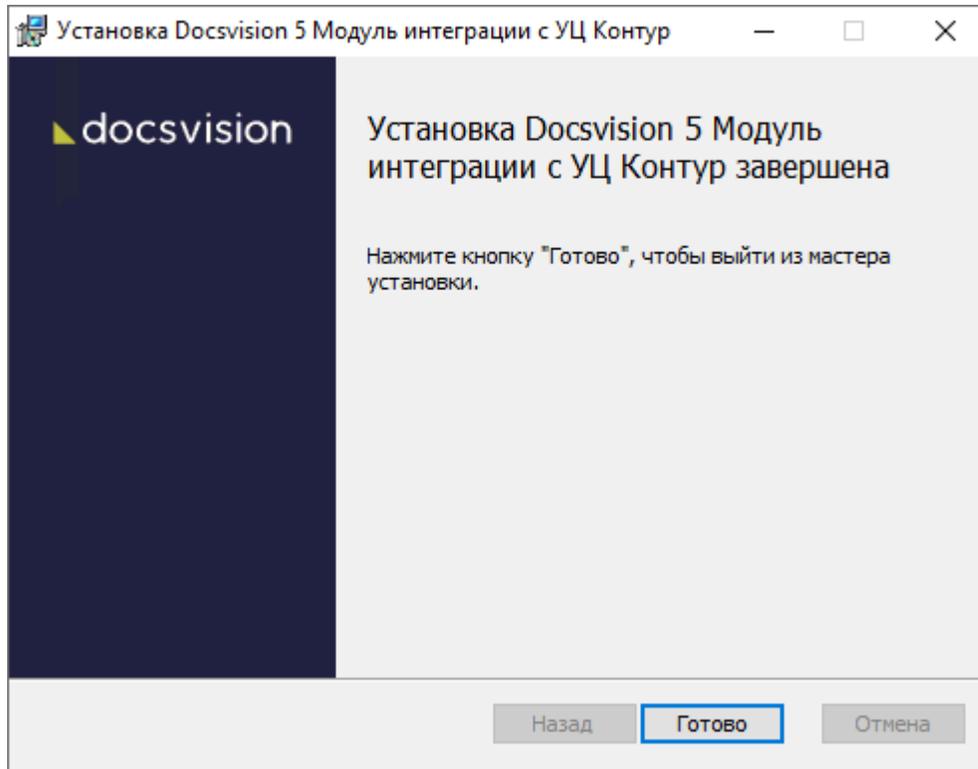


Рисунок 4. Завершение установки модуля

5. Загрузка библиотек карточек не требуется, т.к. модуль не имеет собственных библиотек карточек.
6. Запустите "Консоль настройки Docsvision" и завершите настройку Docsvision стандартным образом согласно документации модуля Платформа /dv5/platform/5.5.5/admin/config-master/[здесь] и /dv5/platform/5.5.5/admin/post-config-server/[здесь].

Пользователь, от имени которого запускается "Консоль настройки Docsvision" должен являться администратором Docsvision, а также входить в группы Docsvision в Справочнике сотрудников: *Администраторы УД, Администраторы КС и Системная для WF*.

7. Установите новую версию модуля:
  - На оставшихся узлах кластера Docsvision.

- На сервере СУБП, Web-клиента и других модулей, использующих компоненты модуля *Базовые объекты*.

## Обновление нескольких модулей

Если планируется обновлять несколько модулей, следующие действия следует выполнять один раз после установки новых версий всех модулей:

1. Откройте Консоль настройки Docsvision.

Программа будет запущена в режиме мастера настройки.



Пользователь, от имени которого запускается *Консоль настройки Docsvision* должен являться администратором Docsvision, а также входить в группы Docsvision в Справочнике сотрудников: *Администраторы УД, Администраторы КС и Системная для WF*.

2. На странице *Базы данных* выберите рабочую БД Docsvision, установите переключатель в режим **Использовать выбранную в списке базу данных** и нажмите **Далее**.

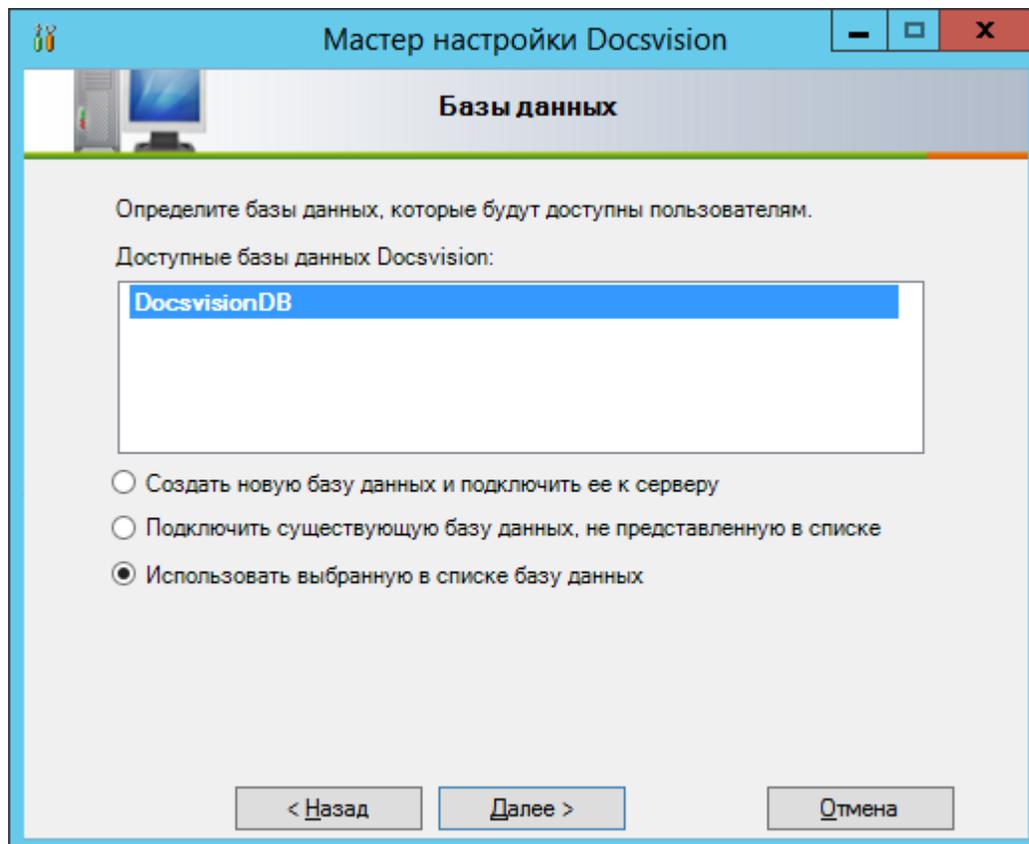


Рисунок 5. Обновление базы данных

3. Будет предложено обновить базу данных до новой версии. Выберите вариант **ДА**.
4. На следующей странице не изменяйте выбор обновляемых библиотек карточек. Нажмите **Далее**.

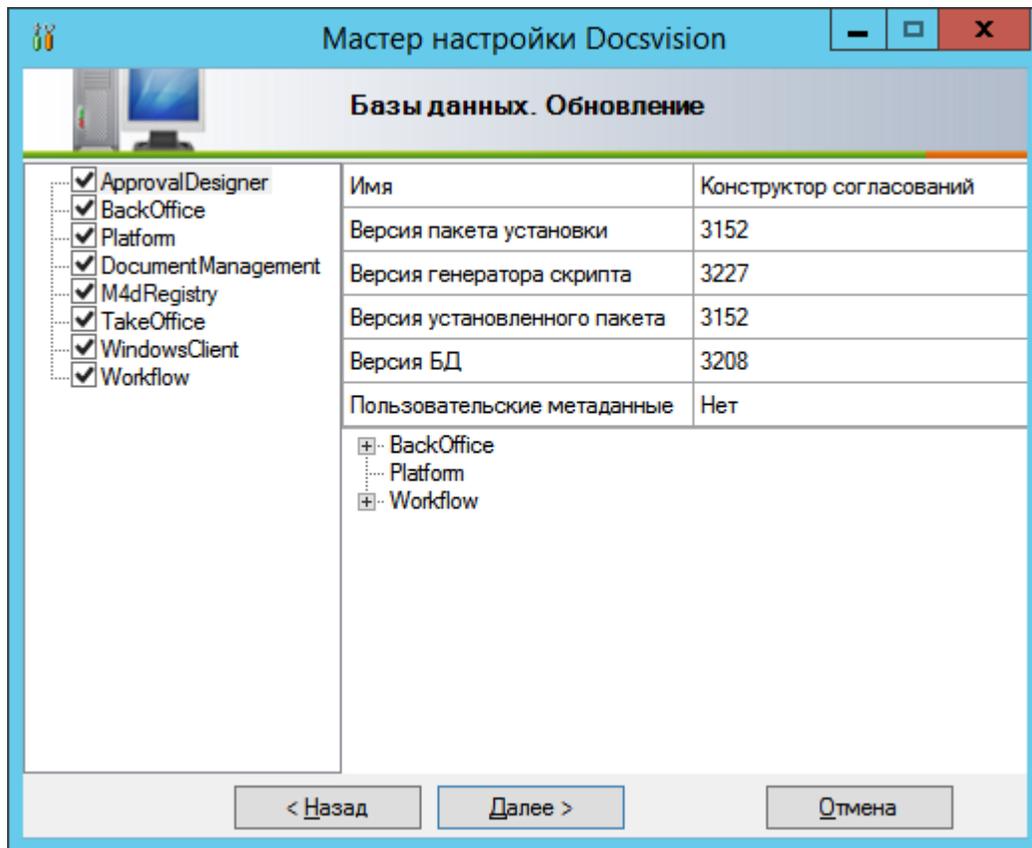


Рисунок 6. Обновление библиотек карточек



Обратите внимание, модуль Модуль интеграции с УЦ Контур не имеет собственных библиотек карточек и не отображается в данном списке.

5. На странице *Параметры базы данных* нажмите **Далее**. Появится запрос на подтверждение обновления — согласитесь. Процесс обновления займет некоторое время.
6. На странице настроек производительности и перезапуска IIS нажмите **Далее**.
7. На странице *Загрузка специальной конфигурации Управление документами* нажмите **Далее**, чтобы загрузить стандартную конфигурацию приложения Управление документами.

## Обратите внимание:

Данные действия приведут к загрузке стандартных настроек приложения *Управление документами*. Если требуется сохранить существующие настройки, на шаге загрузки специальной конфигурации *Управление документами* нажмите **Пропустить**.

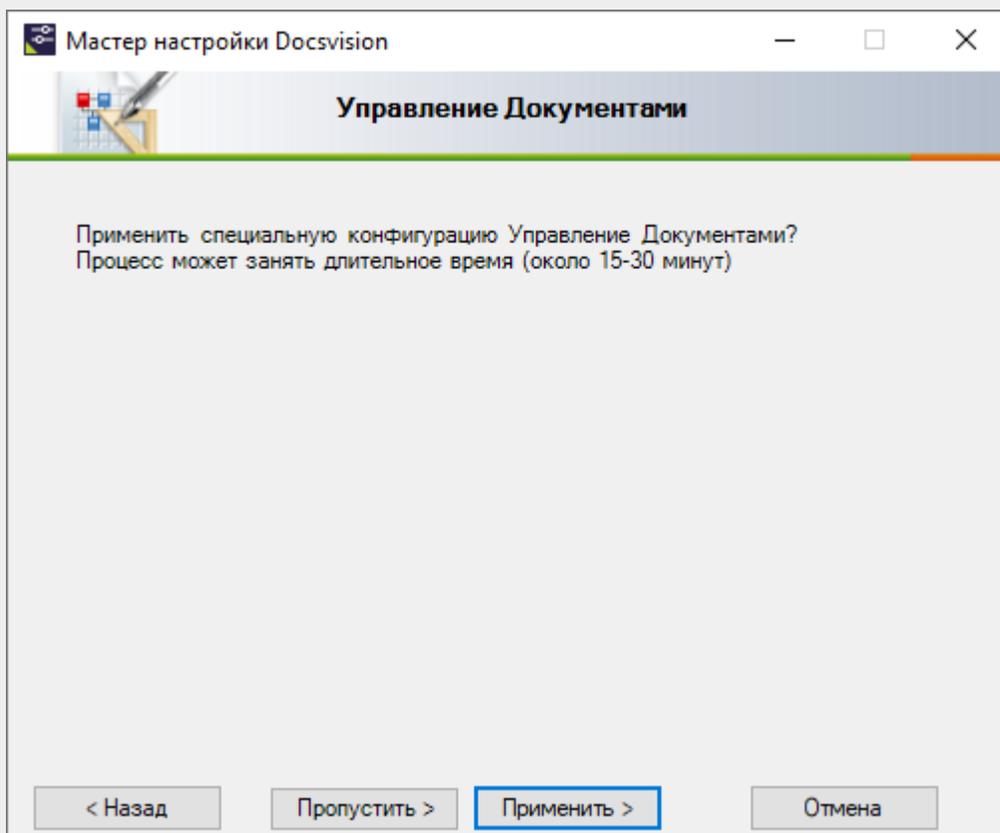


Рисунок 7. Специальная конфигурация "Управление документами"

Подробнее см. в </dv5/documentmgmt/5.5.4/admin/update-module/#update-no-overwrite> [документации по обновлению модуля "Управление документами"].

8. Выполните аналогичные действия на следующих шагах: *Загрузка специальной конфигурации Конструктор согласований* и *Загрузка специальной конфигурации Модуля интеграции с УЦ Контур*.
9. Подтвердите настройки Workflow и выйдите из мастера.
10. Обновите версию модуля на всех узлах кластера Docsvision, СУБП и Web-клиента.
11. Клиентские компоненты модуля на компьютерах пользователей будут обновлены автоматически при запуске Windows-клиента.

Самостоятельно обновить клиентские компоненты можно из пакета установки. *Модуль интеграции с УЦ Контур* не имеет клиентской части.

## Настройка модуля интеграции с УЦ Контур

Выпуск сертификата выполняется с использованием методов API модуля. Описание API модуля приведено в разделе руководстве разработчика модуля: "[Разработка компонентов модуля](#)".

Модуль использует API УЦ Контур и требует получить доступ к Crypto API УЦ Контур.

1. Инструкция по получению доступа к Crypto API [приведена на сайте УЦ Контур](#).
2. Обратите внимание, что работа в продуктовой среде осуществляется по HTTPS-соединению и потребует настроить защищенный канал связи, использующий ГОСТ-криптографию. Подробности приведены по ссылке: [настройка ГОСТ-TLS](#).
3. Для подачи заявления в ОС Windows используется приложение PkiTools (см. инструкцию из шага выше). При настройке ГОСТ-криптографии через приложение откажитесь от установки пароля:



В окне установки пароля для контейнера не задавайте пароль, оставьте поле *Новый пароль* пустым.

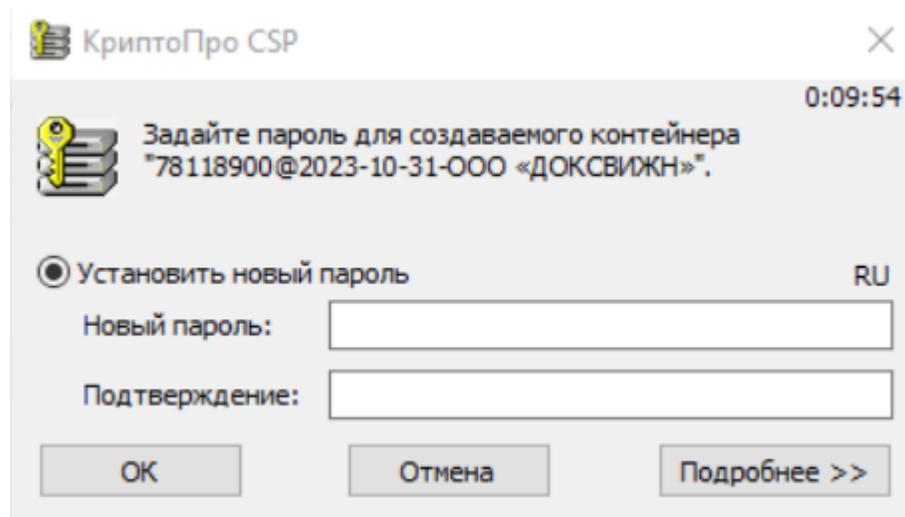


Рисунок 8. Пароль для контейнера

4. На финальном шаге настройки по инструкции [настройка ГОСТ-TLS](#) в приложении на сервере будет создан файл запроса и контейнер закрытого

ключа. Этот файл запроса и заявление нужно отправить в УЦ Контур по адресу, указанному на странице "[Получение доступа к Crypto API](#)".

- Полученный от КриптоПро сертификат нужно установить в контейнер закрытого ключа по инструкции [Установка личного сертификата через КриптоПро CSP](#).
- Убедитесь, что у пользователя, для которого будет выпускаться сертификат, в карточке сотрудника указано ФИО и заполнен номер телефона в поле "Рабочий телефон". На этот номер позже будут приходить СМС уведомления при подписании.
- Укажите значение следующих настроек в Конструкторе справочников:

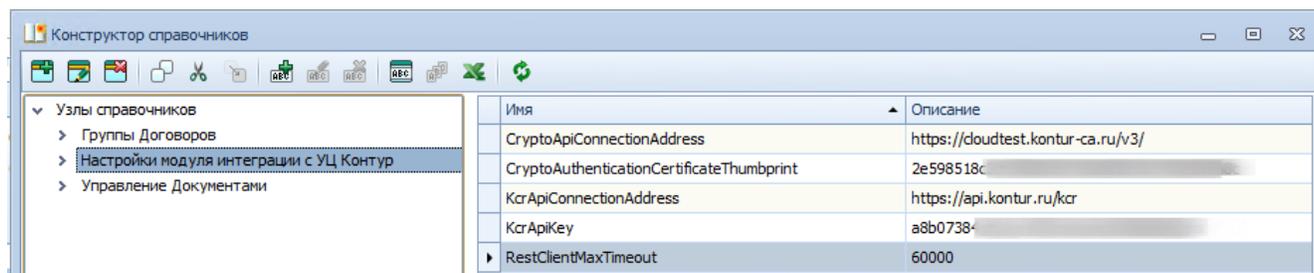


Рисунок 9. Заполненные настройки в конструкторе справочников

```
CryptoApiConnectionAddress "https://cloudtest.kontur-ca.ru/v3/" ①  
CryptoAuthenticationCertificateThumbprint "certificate-thumbprint" ②  
KcrApiConnectionAddress "https://api.kontur.ru/kcr/" ③  
KcrApiKey "api-key" ④  
RestClientMaxTimeout "60000" ⑤
```

- ① `CryptoApiConnectionAddress` — точка подключения к КриптоПро.
- ② `CryptoAuthenticationCertificateThumbprint` — отпечаток сертификата, переданного контуром по заявке.
- ③ `KcrApiConnectionAddress` — точка подключения к API УЦ Контур.
- ④ `KcrApiKey` — API-ключ УЦ Контур.
- ⑤ `RestClientMaxTimeout` — интервал времени (в миллисекундах), в течение которого будет ожидаться ответ от УЦ Контур. Значение по умолчанию 60 секунд.

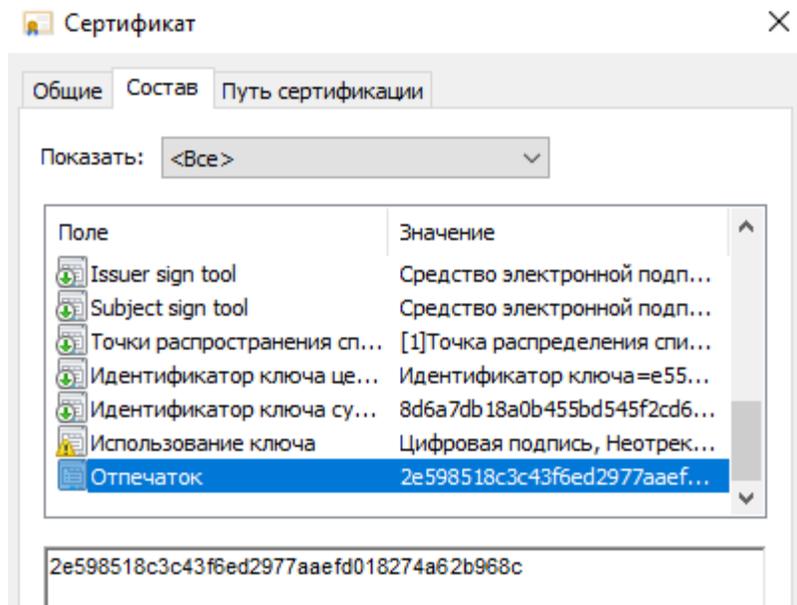


Рисунок 10. Отпечаток сертификата

8. Подайте заявку на сертификат с использованием "[API модуля](#)".
9. Убедитесь, что в справочнике сотрудников в карточке сотрудника указан полученный сертификат.

# Руководство пользователя модуля

## Подписание с помощью сертификата УЦ Контур

1. Откройте Web-клиент.
2. Создайте исходящий документ.
3. Отправьте документ на согласование.
4. Вызовите стандартный диалог подписания Web-клиента, описанный в разделе `"/dv5/webclient/5.5.17/user/docs-sign/[Подписание карточек типа "Документ"]"` Документации модуля *Web-клиент*.
5. Согласитесь выполнить поиск сертификата в облаке:

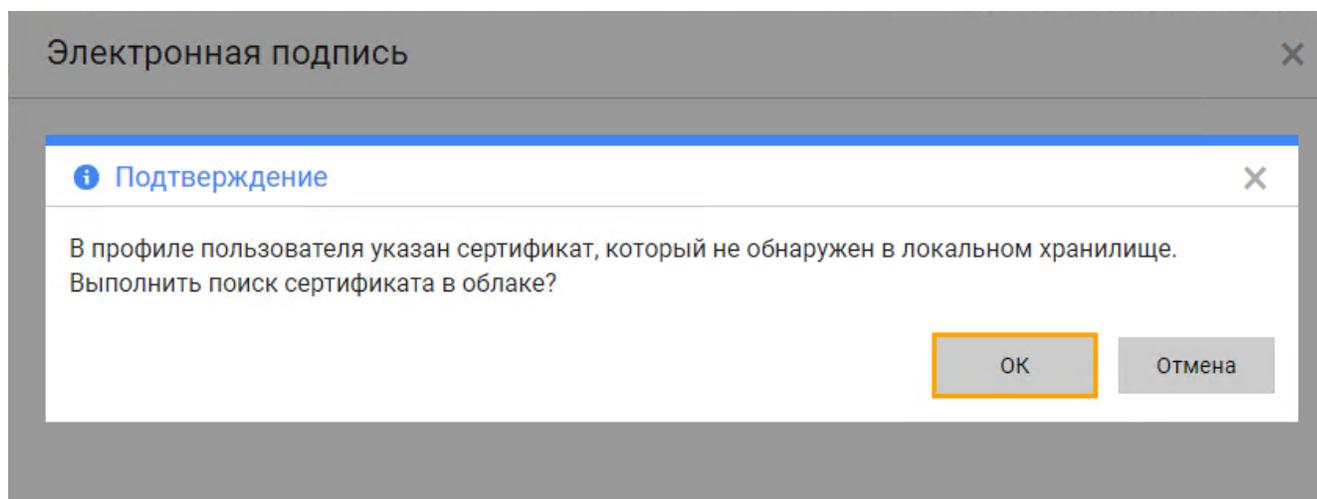


Рисунок 11. Поиск сертификата в облаке

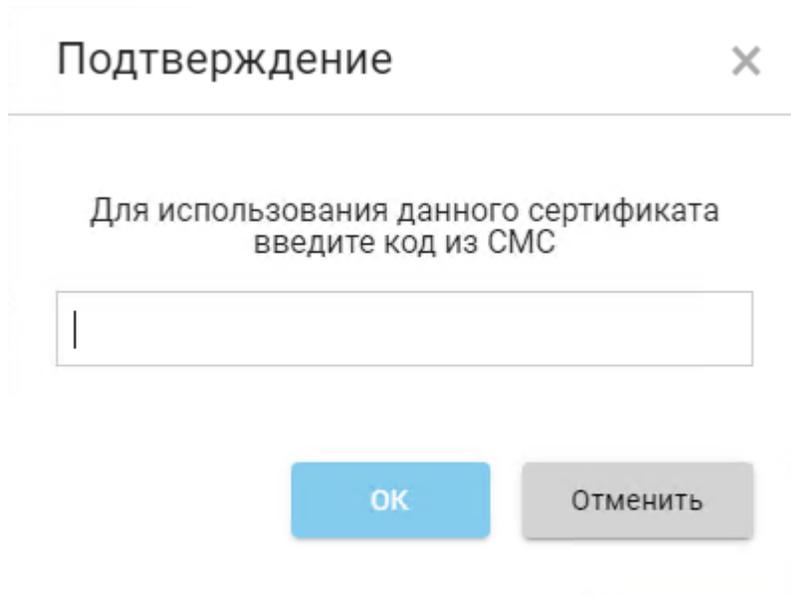
6. Выберите облачный сертификат.
7. Нажмите **Подписать**.

Возникновение ошибки на данном этапе может означать, что [при настройке модуля](#) был установлен пароль для контейнера.

Чтобы исправить ошибку, выполните следующие действия:

- i. Откройте КриптоПро CSP.
- ii. На вкладке *Сервис* нажмите **Удалить сохранённые пароли**.
- iii. Затем слева нажмите на кнопку **Изменить пароль** и оставьте поле *Пароль* пустым.

8. Введите код из СМС и нажмите **ОК**:



Подтверждение

Для использования данного сертификата  
введите код из СМС

ОК Отменить

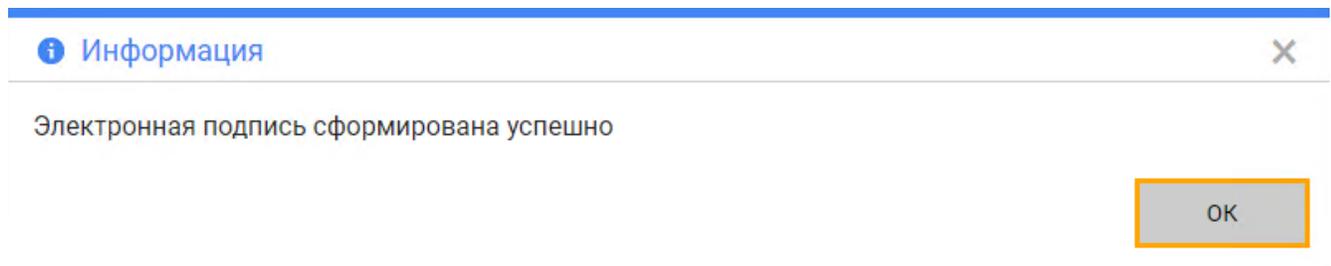
Рисунок 12. Введите код из СМС

9. После нажатия кнопки и подтверждения Web-клиент выполняет запрос к УЦ Контур и возвращает результат проверки.



Обращение к УЦ Контур может занять время. Не закрывайте страницу до появления сообщения об успешном подписании, в противном случае подпись не будет прикреплена к карточке.

10. Электронная подпись сформирована успешно.



Информация

Электронная подпись сформирована успешно

ОК

Рисунок 13. ЭП сформирована успешно

# Руководство разработчика модуля

## Разработка компонентов модуля

Модуль предоставляет для использования два блока методов API.

Первый блок API модуля предоставляет возможность выпуска сертификата УЦ Контур. Второй блок API позволяет использовать выпущенные сертификаты для подписания документов через сервис DSS.

## Выпуск сертификата УЦ Контур

Выпуск сертификата УЦ Контур осуществляется с помощью методов, определённых в сервисе `DocsVision.Kontur.DssService.ObjectModel.Services.IKonturIssuesService`.



При создании заявки на выпуск сертификата из УЦ Контур телефон необходимо передавать из поля `MobilePhone` карточки сотрудника. Заполнение поля *Мобильный телефон* в карточке сотрудника является обязательным для выпуска сертификата УЦ Контур.

Указанный сервис не добавляется в контекст объектов или DI автоматически, требуемые сервисы необходимо проинициализировать самостоятельно. Чтобы упростить процедуру регистрации сервисов API УЦ Контур, добавьте в проект следующий вспомогательный класс, с помощью которого можно проинициализировать необходимые сервисы для стандартного контекста объектов Docsvision:

```
internal static class ObjectContextEnrichment
{
    public static ObjectContext EnrichServices(this ObjectContext context)
    {
        if (context is null)
            throw new ArgumentNullException(nameof(context));

        AddService(context, GetKonturOperationsLogger());
        AddService(context, GetKonturServiceSettings(context));
        AddService(context, GetKonturKcrService(context));
        AddService(context, GetKonturIssuesService(context));

        return context;
    }
}
```

```

    private static IKonturIssuesService GetKonturIssuesService(this ObjectContext
context)
    {
        var logger = context.GetService<IKonturOperationsLogger>();
        var kcrService = context.GetService<IKonturKcrService>();
        return new KonturIssuesServiceImpl(context, kcrService, logger);
    }

    private static IKonturKcrService GetKonturKcrService(ObjectContext context)
    {
        var settingService = context.GetService<IKonturServiceSettings>();
        var logger = context.GetService<IKonturOperationsLogger>();

        return new KonturKcrServiceImpl(settingService, logger);
    }

    private static IKonturServiceSettings GetKonturServiceSettings(ObjectContext
context)
    {
        return new KonturServiceSettingsUniversalStorageImpl(context);
    }

    private static IKonturOperationsLogger GetKonturOperationsLogger()
    {
        return new KonturOperationsLoggerMock();
    }

    private static void AddService<TService>(IServiceContainer container, TService
serviceImpl)
    {
        container.AddService(typeof(TService), serviceImpl);
    }
}

internal class KonturOperationsDiagnosticsTraceLogger : IKonturOperationsLogger
{
    public void Debug(string message)
    {
        System.Diagnostics.Trace.WriteLine("[Debug] " + message);
    }

    public void Error(string message)
    {
        System.Diagnostics.Trace.WriteLine("[Error] " + message);
    }

    public void Info(string message)

```

```

    {
        System.Diagnostics.Trace.WriteLine("[Info] " + message);
    }

    public void Trace(string message)
    {
        System.Diagnostics.Trace.WriteLine("[Trace] " + message);
    }

    public void Warn(string message)
    {
        System.Diagnostics.Trace.WriteLine("[Warn] " + message);
    }
}

```

`KonturOperationsDiagnosticsTraceLogger` реализует простой способ логирования в канал `System.Diagnostics.Trace`. При необходимости реализуйте собственный способ логирования.

Используйте вспомогательный метод `EnrichServices` для добавления сервисов УЦ Контур в существующий контекст объектов:

```

var context = CreateObjectContext();
context.EnrichServices();

```

Приведённая выше стандартная реализация инициализации сервисов предполагает, что конфигурация работы с УЦ Контур хранить в Конструкторе справочников в стандартной секции настроек УЦ Контур (требуется установка модуля УЦ Контур). Для получения такой конфигурации используется реализация сервиса `DocsVision.Kontur.DssService.ObjectModel.Services.KonturServiceSettingsUniversalStorageImpl`. Если требуется хранить конфигурацию в нестандартном хранилище, возможно собственная реализация сервиса `DocsVision.Kontur.DssService.Interfaces.IKonturServiceSettings`.

## Работа с сервисом выпуска сертификатов Контур

Проинициализированный сервис `IKonturIssuesService` можно получить стандартным образом из контекста объектов. Если сервис был добавлен именно в `ObjectContext` — не используется DI:

```
var issuesService = context.GetService<IKonturIssuesService>();
```

## Создание заявки на выпуск сертификата

Прежде всего необходимо создать заявку на выпуск сертификата:

```
var employee = GetReceivingCertificateEmployee(); ①  
var inn = GetReceivingCertificateInn(); ②  
var result = issuesService.CreateCertificateIssue(employee, inn);
```

① `employee` — сотрудник (типа `StaffEmployee`), для которого формируется заявка на выпуск сертификата.

② `inn` — ИНН сотрудника

Метод `CreateCertificateIssue` вернёт результат операции с ID заявки на выпуск сертификата. Результат операции необходимо проверить на успешность завершения и, при необходимости, самостоятельно вернуть исключение, либо иным образом обработать ошибку:

```
if (result.Success) ①  
    return result.Data.Id; ②  
throw new ApplicationException(result.Message);
```

① Если успешно — возвращается ID заявки

② Иначе сгенерируем исключение с ошибкой, которую вернул УЦ Контур

Аналогичным образом необходимо обрабатывать результаты других методов сервиса `IKonturIssuesService`.

Возврат `OperationResult` не гарантирует обработку всех ошибок внутри метода — используйте `try/catch` при вызове методов.

Обратите внимание, в карточке сотрудника должны быть заполнены следующие поля:

- `employee.LastName`
- `employee.FirstName`
- `employee.MiddleName`

- `employee.Email`
- `employee.Phone`

## Проверка статуса актуализации заявки

После внесения любых изменений в заявку необходимо проверять статуса её актуализации, иначе следующий запрос к API УЦ Контур может завершиться ошибкой. Проверка статуса актуализации выполняется следующим способом:

```
var result = issuesService.GetIssue(GetIssueId());
if (!result.Success)
    throw new ApplicationException(result.Message);
return result.Data.Actualized;
```

Метод `GetIssueId` возвращает ID заявки, для которой выполняется операция.

После выполнения любой операции с заявкой всегда проверяйте её статус `Actualized` и выполняйте следующий запрос только если `Actualized` в значении `true`.

## Добавление подтверждающего документа в заявку

После создания заявки необходимо добавить в неё подтверждающий документ, например, паспорт РФ:

```
var result = issuesService.AddDocumentRequisites(GetIssueId(),
    new IdentityDocument ①
    {
        DocumentType = DocumentTypes.Passport,
        Series = PassportSeries.Text,
        Number = PassportNumber.Text,
        IssuerCode = PassportIssuer.Text,
        IssueDate = PassportDate.SelectedDate.Value
    });

if (result.Success)
    throw new ApplicationException(result.Message);
```

① Устанавливается тип документа и заполняются данные документа.

## Прикладывание скан документа к заявке

После добавления данных документа нужно приложить его скан:

```

var scanData = new SendingFileInfo ①
{
    Data = Resource.passport,
    Name = "Паспорт.pdf",
    FileId = Guid.NewGuid(),
    LongSize = Resource.passport.LongLength
};

var result = issuesService.SendDocumentScan(GetIssueId(), scanData,
DocumentTypes.Passport);
if (!result.Success)
else
    throw new ApplicationException(result.Message);

```

① В `scanData` записываются бинарные данные отправляемого файла, его имя и размер.

## Получение заявления на выпуск сертификата

Пользователю необходимо подписать заявление на выпуск сертификата. Получить печатную форму заявления можно:

```

var result = issuesService.GenerateRequestTemplate(GetIssueId(),
TemplateType.ReleaseStatement);
var path = Path.Combine(AppDomain.CurrentDomain.BaseDirectory, $"Заявление_
{issueId}.pdf"); ①

if (result.Success)
    File.WriteAllBytes(path, result.Data);
else
    throw new ApplicationException(result.Message);

```

① Бинарные данные файла заявления записываются в файл `Заявление_{issueId}.pdf`.

## Отправка скана заявления на выпуск сертификата

Пользователь должен подписать заявление. Далее скан документа требуется отправить в Контур УЦ с помощью рассмотренных ранее методов `AddDocumentRequisites` и `SendDocumentScan`:

```

var result = issuesService.AddDocumentRequisites(GetIssueId(), new
IdentityDocument { DocumentType = DocumentTypes.ReleaseStatement });
if (!result.Success)

```

```
        throw new ApplicationException(result.Message);
        var sendResult = issuesService.SendDocumentScan(GetIssueId(), scanData,
DocumentTypes.ReleaseStatement);
        if (!sendResult.Success)
            throw new ApplicationException(sendResult.Message);
```

## Отправка заявки на проверку

Сформированную заявку необходимо отправить на проверку:

```
var result = issuesService.SendRequestForValidation(GetIssueId());
if (!result.Success)
    throw new ApplicationException(result.Message);
```

## Удостоверение личности

Этот пункт нужен только для выпуска сертификатов на физических лиц или контрагентов.

Данная операция необходима для разбора инцидентов при нарушении правил УЛ.

```
var confirmedEmployee = GetConfirmedEmployee(); ①
var result = issuesService.SubjectIdentification(GetIssueId(),
confirmedEmployee);
if (!result.Success)
    throw new ApplicationException(result.Message);
```

① **confirmedEmployee** — сотрудник, который будет указан удостоверяющим личность.

## Отслеживание статуса заявки

После выполнения всех операций по формированию заявки, необходимо дождаться её одобрения, прежде чем выгружать сертификат.

```
var result = issuesService.GetIssue(GetIssueId());
if (!result.Success)
    throw new ApplicationException(result.Message);

if (result.Data.Actualized)
    return result.Data.Status; ①
```

① Значение **Status** в данном случае может принимать:

- **unknown** — может быть только возвращаемым типом.
- **preparing** — заявку создали и готовят к проверке.
- **validating** — заявку проверяет оператор УЦ. В этом статусе нельзя менять заявку.
- **correction** — при проверке заявки были выявлены замечания. Нужно их исправить и повторно отправить заявку на проверку. Замечания можно найти в поле **Data.CorrectionInfo**.
- **approved** — заявка проверена, замечаний не выявлено. Нужно удостоверить личность и перенаправить заявку в личный кабинет. В этом статусе нельзя менять реквизиты и документы заявки, кроме документов для удостоверения личности.
- **releasing** — субъект сертификата отправил запрос на выпуск в личном кабинете. В этом статусе нельзя менять заявку.
- **released** — сертификат выпущен. В этом статусе нельзя менять заявку. Заявка считается выпущенной, если **result.Data.Status** в значении **released**.

## Загрузка сертификата в карточку сотрудника

Выпущенный сертификат Контур должен быть загружен в карточку сотрудника:

```
var employee = GetReceivingCertificateEmployee();
var result = issuesService.DownloadCertificateAndAssignToEmployee(
GetIssueId(), employee);
if (!result.Success)
    throw new ApplicationException(result.Message);
```

После этого пользователь может использовать сертификат с помощью облачного подписания Контур.

## Удаление заявки на выпуск сертификата

Если требуется заявку на выпуск можно удалить, но только в статусах **preparing**, **correction**, **approved**:

```
var result = issuesService.DeleteCertificateIssueRequest(GetIssueId());
if (!result.Success)
    throw new ApplicationException(result.Message);
```

# Упрощённый выпуск сертификата

В данном разделе документации описан сценарий упрощённого выпуска облачного сертификата НЭП с подтверждением через ЕСИА. Для подтверждения необходимо обязательное наличие подтвержденной учетной записи Госуслуги.

Чтобы выпустить сертификат по упрощённой процедуре:

1. Создайте и заполните заявку на выпуск сертификата.
  - a. Создайте заявку на выпуск сертификата.
  - b. Добавьте подтверждающий документ в заявку.
2. Загрузите скан подтверждающего документа.



Загружать реальные сканы документов не требуется, но необходим любой "заместитель", например, черная точка размером 1x1 пиксель.

3. Создайте запрос на подтверждение выпуска сертификата с подтверждением в ЕСИА.

```
var snils = GetReceivingCertificateEmployeeSnils();
var result = issuesService.SendReleaseStatementWithEsia(GetIssueId(), employee, snils);
if (result.Success)
    throw new ApplicationException(result.Message);
```

После подтверждения выпуска сертификата пользователем в кабинете Госуслуги, он будет выпущен в УЦ Контур. Загрузить выпущенный сертификат в карточку сотрудника нужно стандартным методом `DownloadCertificateAndAssignToEmployee`.

## Подписание сертификатом УЦ Контур

Подписание с помощью сервиса облачного подписания Контур осуществляется с помощью методов, определённых в сервисе `IKonturDssService`.

## Описание API

**Руководство разработчика модуля интеграции с УЦ Контур**

## Общая документация

Общая информация об изменениях, исправленных ошибках и накопительных обновлениях.

- [IKonturIssuesService](#) — интерфейс
- [SendingFileInfo](#) — класс
- [IKonturDssService](#) — интерфейс
- [Больше информации в навигационном меню слева...](#)

## IKonturIssuesService — интерфейс

Сервис выпуска сертификатов Контур.

- **Пространство имён:** `DocsVision.Kontur.DssService.ObjectModel.Services`
- **Сборка:** `Kontur.DssService.ObjectModel.dll`

### Синтаксис

```
public interface IKonturIssuesService
```

### Методы

Имя	Описание
<code>CreateCertificateIssue(CertificateApplicantInfo)</code>	Создание заявки на выпуск сертификата.
<code>CreateCertificateIssue(StaffEmployee, string)</code>	Создание заявки на выпуск сертификата.
<code>AddDocumentRequisites(Guid, IdentityDocument)</code>	Добавляет документы в заявку.
<code>SendDocumentScan(Guid, SendingFileInfo, DocumentTypes)</code>	Отправка скана документа.
<code>SendReleaseStatement(Guid)</code>	Запрос на подтверждение операции.
<code>SendReleaseStatementWithEsia(Guid, CertificateConfirmationApplicantInfo)</code>	Создает запрос на подтверждение через госуслуги.

<b>Имя</b>	<b>Описание</b>
<code>SendReleaseStatementWithEsia(Guid, StaffEmployee, string)</code>	Создает запрос на подтверждение через госуслуги.
<code>SignReleaseStatement(Guid, string)</code>	Подпись заявления на выпуск сертификата НЭП.
<code>SignReleaseStatement(Guid)</code>	Подпись заявления на выпуск сертификата НЭП через госуслуги.
<code>SendRequestForValidation(Guid)</code>	Отправляет заявку на проверку.
<code>DeleteCertificateIssueRequest(Guid)</code>	Удаление заявки на сертификат.
<code>DownloadCertificate(Guid)</code>	Запрос файла сертификата УНЭП и загрузка его в карточку сотрудника.
<code>DownloadCertificateAndAssignToEmployee(Guid, StaffEmployee)</code>	Запрос файла сертификата УНЭП и загрузка его в карточку сотрудника.
<code>GetIssue(Guid)</code>	Запрос заявки на сертификат, проверка актуализации.
<code>GenerateRequestTemplate(Guid, TemplateType)</code>	Генерация шаблона документа для заявления на выпуск сертификата.
<code>EditCertificateIssueRequest(Guid, CertificateUpdateApplicantInfo)</code>	Изменение заявки на сертификат.
<code>EditCertificateIssueRequest(Guid, CertificateUpdateApplicantInfo)</code>	Изменение заявки на сертификат.
<code>GetEventList(int)</code>	Запрос ленты событий заявок.
<code>GetEventList(int, string)</code>	Запрос ленты событий заявок.
<code>SubjectIdentification(Guid, StaffEmployee)</code>	Добавить или изменить удостоверение личности.
<code>ForwardToCabinet(Guid)</code>	Перенаправить заявку в личный кабинет.

### **CreateCertificateIssue — метод (CertificateApplicantInfo)**

Добавляет документы в заявку.

- **Пространство имён:** `DocsVision.Kontur.DssService.ObjectModel.Services`

- **Сборка:** `Kontur.DssService.ObjectModel.dll`

#### Синтаксис

```
OperationResult<KonturIssuesResponse> CreateCertificateIssue(CertificateApplicantInfo applicantInfo);
```

#### Параметры

##### **applicantInfo**

Тип: `CertificateApplicantInfo`

Данные сотрудника, для которого формируется сертификат.

#### Возвращаемое значение

Тип: `OperationResult`

Результат выполнения операции.

#### **CreateCertificateIssue** — метод (**StaffEmployee, string**)

Добавляет документы в заявку.

- **Пространство имён:** `DocsVision.Kontur.DssService.ObjectModel.Services`
- **Сборка:** `Kontur.DssService.ObjectModel.dll`

#### Синтаксис

```
OperationResult<KonturIssuesResponse> CreateCertificateIssue(StaffEmployee employee, string inn);
```

#### Параметры

##### **employee**

Тип: `StaffEmployee`

Сотрудник, для которого формируется сертификат.

##### **inn**

Тип: `string`

ИНН сотрудника.

## Возвращаемое значение

Тип: `OperationResult`

Результат выполнения операции.

## AddDocumentRequisites — метод (Guid, IdentityDocument)

Добавляет документы в заявку.

- **Пространство имён:** `DocsVision.Kontur.DssService.ObjectModel.Services`
- **Сборка:** `Kontur.DssService.ObjectModel.dll`

### Синтаксис

```
OperationResult AddDocumentRequisites(Guid issueId, IdentityDocument passData);
```

## Параметры

### issueId

Тип: `Guid`

ID заявки.

### employeeId

Тип: `IdentityDocument`

ID сотрудника, выполнившего запрос.

## Возвращаемое значение

Тип: `OperationResult`

Результат выполнения операции.

## SendDocumentScan — метод (Guid, SendingFileInfo, DocumentTypes)

Отправка скана документа.

- **Пространство имён:** `DocsVision.Kontur.DssService.ObjectModel.Services`
- **Сборка:** `Kontur.DssService.ObjectModel.dll`

### Синтаксис

```
OperationResult SendDocumentScan(Guid issueId, SendingFileInfo fileInfo, DocumentTypes
```

```
documentType);
```

## Параметры

### issueId

Тип: `Guid`

Идентификатор запроса.

### fileInfo

Тип: `SendingFileInfo`

Данные скана документа.

### documentType

Тип: `DocumentTypes`

Тип документа.

## Возвращаемое значение

Тип: `OperationResult`

Результат выполнения операции.

## SendReleaseStatement — метод (Guid)

Запрос на подтверждение операции.

- **Пространство имён:** `DocsVision.Kontur.DssService.ObjectModel.Services`
- **Сборка:** `Kontur.DssService.ObjectModel.dll`

## Синтаксис

```
OperationResult<string> SendReleaseStatement(Guid issueId);
```

## Параметры

### issueId

Тип: `Guid`

Идентификатор заявки.

## Возвращаемое значение

Тип: `OperationResult`

Результат выполнения операции.

## `SendReleaseStatementWithEsia` — метод (`Guid`, `CertificateConfirmationApplicantInfo`)

Создает запрос на подтверждение через госуслуги.

- **Пространство имён:** `DocsVision.Kontur.DssService.ObjectModel.Services`
- **Сборка:** `Kontur.DssService.ObjectModel.dll`

### Синтаксис

```
OperationResult<string> SendReleaseStatementWithEsia(Guid issueId,  
CertificateConfirmationApplicantInfo applicantInfo);
```

## Параметры

### `issueId`

Тип: `Guid`

Идентификатор документа.

### `applicantInfo`

Тип: `CertificateConfirmationApplicantInfo`

Данные сотрудника, для запроса подтверждения выпуска сертификата.

## Возвращаемое значение

Тип: `OperationResult`

Результат выполнения операции.

## `SendReleaseStatementWithEsia` — метод (`Guid`, `StaffEmployee`, `string`)

Создает запрос на подтверждение через госуслуги.

- **Пространство имён:** `DocsVision.Kontur.DssService.ObjectModel.Services`
- **Сборка:** `Kontur.DssService.ObjectModel.dll`

### Синтаксис

```
OperationResult<string> SendReleaseStatementWithEsia(Guid issueId, StaffEmployee employee, string snils);
```

## Параметры

### issueId

Тип: `Guid`

Идентификатор документа.

### employee

Тип: `StaffEmployee`

Сотрудник.

### snils

Тип: `string`

СНИЛС сотрудника.

## Возвращаемое значение

Тип: `OperationResult`

Результат выполнения операции.

## SignReleaseStatement — метод (Guid, string)

Подпись заявления на выпуск сертификата НЭП.

- **Пространство имён:** `DocsVision.Kontur.DssService.ObjectModel.Services`
- **Сборка:** `Kontur.DssService.ObjectModel.dll`

## Синтаксис

```
OperationResult SignReleaseStatement(Guid issueId, string smsCode);
```

## Параметры

### issueId

Тип: `Guid`

Идентификатор заявки.

## **smsCode**

Тип: `string`

СМС-код подтверждения.

## **Возвращаемое значение**

Тип: `OperationResult`

Результат выполнения операции.

## **SignReleaseStatement — метод (Guid)**

Подпись заявления на выпуск сертификата НЭП через госуслуги.

- **Пространство имён:** `DocsVision.Kontur.DssService.ObjectModel.Services`
- **Сборка:** `Kontur.DssService.ObjectModel.dll`

### **Синтаксис**

```
OperationResult SignReleaseStatement(Guid issueId);
```

## **Параметры**

### **issueId**

Тип: `Guid`

Идентификатор заявки.

## **Возвращаемое значение**

Тип: `OperationResult`

Результат выполнения операции.

## **SendRequestForValidation — метод (Guid)**

Отправляет заявку на проверку.

- **Пространство имён:** `DocsVision.Kontur.DssService.ObjectModel.Services`
- **Сборка:** `Kontur.DssService.ObjectModel.dll`

### **Синтаксис**

```
OperationResult SendRequestForValidation(Guid issueId);
```

## Параметры

### issueId

Тип: `Guid`

Идентификатор заявки.

## Возвращаемое значение

Тип: `OperationResult`

Результат выполнения операции.

## DeleteCertificateIssueRequest — метод (Guid)

Удаление заявки на сертификат.

- **Пространство имён:** `DocsVision.Kontur.DssService.ObjectModel.Services`
- **Сборка:** `Kontur.DssService.ObjectModel.dll`

## Синтаксис

```
OperationResult DeleteCertificateIssueRequest(Guid issueId);
```

## Параметры

### issueId

Тип: `Guid`

Идентификатор заявки.

## Возвращаемое значение

Тип: `OperationResult`

Результат выполнения операции.

## DownloadCertificate — метод (Guid)

Запрос файла сертификата УНЭП.

- **Пространство имён:** `DocsVision.Kontur.DssService.ObjectModel.Services`

- **Сборка:** `Kontur.DssService.ObjectModel.dll`

#### Синтаксис

```
OperationResult<byte[]> DownloadCertificate(Guid issueId);
```

#### Параметры

##### **issueId**

Тип: `Guid`

Идентификатор заявки.

#### Возвращаемое значение

Тип: `OperationResult`

Результат выполнения операции.

#### **DownloadCertificateAndAssignToEmployee** — метод (`Guid`, `StaffEmployee`)

Запрос файла сертификата УНЭП и загрузка его в карточку сотрудника.

- **Пространство имён:** `DocsVision.Kontur.DssService.ObjectModel.Services`
- **Сборка:** `Kontur.DssService.ObjectModel.dll`

#### Синтаксис

```
OperationResult DownloadCertificateAndAssignToEmployee(Guid issueId, StaffEmployee employee);
```

#### Параметры

##### **issueId**

Тип: `Guid`

Идентификатор заявки.

##### **employee**

Тип: `StaffEmployee`

Сотрудник, в карточку которого загружается сертификат Контур.

## Возвращаемое значение

Тип: `OperationResult`

Результат выполнения операции.

## GetIssue — метод (Guid)

Запрос заявки на сертификат (проверка актуализации).

- **Пространство имён:** `DocsVision.Kontur.DssService.ObjectModel.Services`
- **Сборка:** `Kontur.DssService.ObjectModel.dll`

### Синтаксис

```
OperationResult<KonturIssuesResponse> GetIssue(Guid issueId);
```

## Параметры

### issueId

Тип: `Guid`

Идентификатор заявки.

## Возвращаемое значение

Тип: `OperationResult`

Результат выполнения операции.

## GenerateRequestTemplate — метод (Guid, TemplateType)

Запрос заявки на сертификат (проверка актуализации).

- **Пространство имён:** `DocsVision.Kontur.DssService.ObjectModel.Services`
- **Сборка:** `Kontur.DssService.ObjectModel.dll`

### Синтаксис

```
OperationResult<byte[]> GenerateRequestTemplate(Guid issueId, TemplateType templateType);
```

## Параметры

### issueId

Тип: `Guid`

Идентификатор заявки.

### **templateType**

Тип: `TemplateType`

Тип документа получаемый от УЦ Контур.

### **Возвращаемое значение**

Тип: `OperationResult`

Результат выполнения операции.

### **EditCertificateIssueRequest – метод (Guid, CertificateUpdateApplicantInfo)**

Запрос заявки на сертификат (проверка актуализации).

- **Пространство имён:** `DocsVision.Kontur.DssService.ObjectModel.Services`
- **Сборка:** `Kontur.DssService.ObjectModel.dll`

#### **Синтаксис**

```
OperationResult EditCertificateIssueRequest(Guid issueId, CertificateUpdateApplicantInfo applicantInfo);
```

### **Параметры**

#### **issueId**

Тип: `Guid`

Идентификатор заявки.

#### **applicantInfo**

Тип: `CertificateUpdateApplicantInfo`

Обновленные данные сотрудника, для которого формируется сертификат.

### **Возвращаемое значение**

Тип: `OperationResult`

Результат выполнения операции.

### **EditCertificateIssueRequest – метод (Guid, StaffEmployee)**

Запрос заявки на сертификат (проверка актуализации).

- **Пространство имён:** `DocsVision.Kontur.DssService.ObjectModel.Services`
- **Сборка:** `Kontur.DssService.ObjectModel.dll`

#### Синтаксис

```
OperationResult EditCertificateIssueRequest(Guid issueId, StaffEmployee employee);
```

#### Параметры

##### **issueId**

Тип: `Guid`

Идентификатор заявки.

##### **employee**

Тип: `StaffEmployee`

Сотрудник, для которого формируется сертификат.

#### Возвращаемое значение

Тип: `OperationResult`

Результат выполнения операции.

#### **GetEventList** — метод (`int`)

Запрос ленты событий заявок.

- **Пространство имён:** `DocsVision.Kontur.DssService.ObjectModel.Services`
- **Сборка:** `Kontur.DssService.ObjectModel.dll`

#### Синтаксис

```
OperationResult<IssueEvents> GetEventList(int limit);
```

#### Параметры

##### **м**

Тип: `int`

Максимальное количество событий, которое нужно вернуть.

## Возвращаемое значение

Тип: `OperationResult`

Результат выполнения операции.

## GetEventList — метод (int, string)

Запрос ленты событий заявок.

- **Пространство имён:** `DocsVision.Kontur.DssService.ObjectModel.Services`
- **Сборка:** `Kontur.DssService.ObjectModel.dll`

### Синтаксис

```
OperationResult<IssueEvents> GetEventList(int limit, string lastId);
```

## Параметры

### limit

Тип: `int`

Максимальное количество событий, которое нужно вернуть.

### lastId

Тип: `string`

Идентификатор события, после которого нужно вернуть события.

## Возвращаемое значение

Тип: `OperationResult`

Результат выполнения операции.

## SubjectIdentification — метод (Guid, StaffEmployee)

Добавить или изменить удостоверение личности.

- **Пространство имён:** `DocsVision.Kontur.DssService.ObjectModel.Services`
- **Сборка:** `Kontur.DssService.ObjectModel.dll`

### Синтаксис

```
OperationResult SubjectIdentification(Guid issueId, StaffEmployee identifiedBy);
```

## Параметры

### issueId

Тип: `Guid`

Идентификатор заявки.

### StaffEmployee

Тип: `StaffEmployee`

Сотрудник, который удостоверил личность.

## Возвращаемое значение

Тип: `OperationResult`

Результат выполнения операции.

## ForwardToCabinet — метод (Guid)

Перенаправить заявку в личный кабинет.

- **Пространство имён:** `DocsVision.Kontur.DssService.ObjectModel.Services`
- **Сборка:** `Kontur.DssService.ObjectModel.dll`

## Синтаксис

```
OperationResult ForwardToCabinet(Guid issueId);
```

## Параметры

### issueId

Тип: `Guid`

Идентификатор заявки.

## Возвращаемое значение

Тип: `OperationResult`

Результат выполнения операции.

## CertificateApplicantInfo — класс

Данные сотрудника, для которого формируется сертификат.

- **Пространство имён:** DocsVision.Kontur.DssService.ObjectModel.Models
- **Сборка:** Kontur.DssService.ObjectModel.dll

### Синтаксис

```
public class CertificateApplicantInfo : CertificateUpdateApplicantInfo
```

### Свойства

Имя	Описание
LastName	Фамилия (обязательно для отправки заявки на проверку).
FirstName	Имя (обязательно для отправки заявки на проверку).
MiddleName	Отчество.
Email	Электронный адрес (обязательно для отправки заявки на проверку).
MobilePhone	Телефон (обязательно для отправки заявки на проверку).

## CertificateApplicantInfo — класс

Данные сотрудника для запроса подтверждения выпуска сертификата.

- **Пространство имён:** DocsVision.Kontur.DssService.ObjectModel.Models
- **Сборка:** Kontur.DssService.ObjectModel.dll

### Синтаксис

```
public class CertificateConfirmationApplicantInfo
```

### Свойства

Имя	Описание
Snils	СНИЛС.

Имя	Описание
BirthDate	Дата рождения.

### CertificateApplicantInfo — класс

Обновлённые данные сотрудника, для которого формируется сертификат.

- **Пространство имён:** DocsVision.Kontur.DssService.ObjectModel.Models
- **Сборка:** Kontur.DssService.ObjectModel.dll

#### Синтаксис

```
public class CertificateUpdateApplicantInfo
```

#### Свойства

Имя	Описание
INN	ИНН (обязательно для отправки заявки на проверку).

### SendingFileInfo — класс

Информация по файлу, отправляемому в УЦ Контур.

- **Пространство имён:** DocsVision.Kontur.DssService.ApiClient.Issues.Models
- **Сборка:** Kontur.DssService.ObjectModel.dll

#### Синтаксис

```
public class SendingFileInfo
```

#### Свойства

Имя	Описание
FileId	Идентификатор файла.
Name	Имя файла.
LongSize	Размер файла.
Data	Данные файла.

## Методы

Имя	Описание
<code>FromFileData(FileData)</code>	<p>Создаёт экземпляр <code>SendingFileInfo</code> по данным файла Docsvision.</p> <p>Параметры:: ObjectContext::: Тип: <code>fileData</code> + Контекст объекта.</p> <p>Возвращаемое значение:: Тип: <code>SendingFileInfo</code> + Результат выполнения операции.</p>

## DocumentTypes — перечисление

Предоставляет перечислитель возможных типов документов.

- **Пространство имён:** `DocsVision.Kontur.DssService.ApiClient.Issues.Models`
- **Сборка:** `Kontur.DssService.ApiClient.dll`

### Синтаксис

```
public enum DocumentTypes
```

### Члены

Имя члена	Описание
<code>Unknown</code>	Может быть только возвращаемым типом.
<code>Passport</code>	Паспорт РФ.
<code>OtherIdentity</code>	Другой документ удостоверяющий личность.
<code>Snils</code>	СНИЛС.
<code>Egrul</code>	Выписка из ЕГРЮЛ.
<code>ManagementCompanyEgrul</code>	Выписка из ЕГРЮЛ управляющей компании.
<code>Egrip</code>	Выписка из ЕГРИП.

Имя члена	Описание
<code>NaturalPersonInn</code>	ИНН физического лица.
<code>SigningAuthority</code>	Подтверждение права подписи.
<code>WarrantWithUseAreas</code>	Доверенность с областями применения.
<code>ReleaseStatement</code>	Заявление на выпуск сертификата.
<code>ApplicantWarrant</code>	Доверенность на получение.
<code>ApplicantPhoto</code>	Фото обратившегося.

## IdentityDocument — класс

Модель данных подтверждающего документа.

- **Пространство имён:** `DocsVision.Kontur.DssService.ObjectModel.Models`
- **Сборка:** `Kontur.DssService.ObjectModel.dll`

### Синтаксис

```
public class IdentityDocument
```

### Свойства

Имя	Описание
<code>DocumentType</code>	Тип документа
<code>Series</code>	Серия
<code>Number</code>	Номер
<code>IssuerName</code>	Кем выдан
<code>IssuerCode</code>	Код подразделения
<code>IssueDate</code>	Дата выдачи

## TemplateType — перечисление

Предоставляет тип документа получаемого от Контур.

- **Пространство имён:** `DocsVision.Kontur.DssService.ApiClient.Issues.Models`
- **Сборка:** `Kontur.DssService.ApiClient.dll`

## Синтаксис

```
public enum TemplateType
```

## Члены

Имя члена	Описание
<code>ReleaseStatement</code>	Заявление на выпуск сертификата.
<code>Receipt</code>	Расписка о том, что сертификат выпущен.

## KonturIssuesResponse — класс

Модель данных подтверждающего документа.

- **Пространство имён:** `DocsVision.Kontur.DssService.ApiClient.Issues.Models`
- **Сборка:** `Kontur.DssService.ApiClient.dll`

## Синтаксис

```
public class KonturIssuesResponse
```

## Свойства

Имя	Описание
<code>Id</code>	Список валидаций документов в СМЭВИдентификатор заявки.
<code>BaseIssueId</code>	Дата и время создания заявкиИдентификатор заявки, на основе которой происходит перевыпуск.
<code>Version</code>	Дата и время последнего изменения заявкиВерсия заявки.
<code>Actualized</code>	Актуализирована ли заявка после изменения.
<code>Status</code>	Статус заявки.
<code>Status.GetDescription</code>	Текстовое описание для статуса заявки.
<code>ForwardedStatus</code>	Статус перенаправления заявки.

Имя	Описание
<code>ForwardedStatus.GetDescription</code>	Текстовое описание для статуса перенаправления заявки.
<code>CertificateTemplateInfo</code>	Информация о шаблоне сертификата.
<code>SubjectInfo</code>	Информация о субъекте сертификата.
<code>CspInfo</code>	Информация о криптопровайдере.
<code>CertificateReleasedInfo</code>	Информация о выпущенном сертификате.
<code>Documents</code>	Список документов заявки.
<code>CorrectionInfo</code>	Информация о том, что нужно исправить в заявке.
<code>SubjectIdentification</code>	Информация об удостоверении личности.
<code>InnerErrors</code>	Список внутренних ошибок заявки.
<code>SmevValidations</code>	Список валидаций документов в СМЭВ.
<code>CreatedAt</code>	Дата и время создания заявки.
<code>UpdatedAt</code>	Дата и время последнего изменения заявки.

## KonturIssuesResponseForwardedStatus — перечисление

Статус заявки.

- **Пространство имён:** `DocsVision.Kontur.DssService.ApiClient.Issues.Models`
- **Сборка:** `Kontur.DssService.ApiClient.dll`

### Синтаксис

```
public enum KonturIssuesResponseStatus
```

### Члены

Имя члена	Описание
<code>Unknown</code>	Может быть только возвращаемым типом
<code>Preparing</code>	Заявку создали и готовят к проверке.

Имя члена	Описание
Validating	Заявку проверяет оператор УЦ. В этом статусе нельзя менять заявку.
Correction	При проверке заявки были выявлены замечания. Нужно их исправить и повторно отправить заявку на проверку. Замечания можно найти в поле <code>correctionInfo</code> .
Approved	Заявка проверена, замечаний не выявлено. Нужно удостоверить личность и перенаправить заявку в личный кабинет. В этом статусе нельзя менять реквизиты и документы заявки (кроме документов для удостоверения личности).
Releasing	Субъект сертификата отправил запрос на выпуск в личном кабинете. В этом статусе нельзя менять заявку.
Released	Сертификат выпущен. В этом статусе нельзя менять заявку.

## KonturIssuesResponseForwardedStatus — перечисление

Статус перенаправления заявки.

- **Пространство имён:** `DocsVision.Kontur.DssService.ApiClient.Issues.Models`
- **Сборка:** `Kontur.DssService.ApiClient.dll`

### Синтаксис

```
public enum KonturIssuesResponseForwardedStatus
```

### Члены

Имя члена	Описание
Unknown	Может быть только возвращаемым типом.

Имя члена	Описание
None	Заявка не перенаправлена.
ForwardedToCabinet	Заявка перенаправлена.

## CorrectionInfo — класс

Информация о том, что нужно исправить в заявке.

- **Пространство имён:** DocsVision.Kontur.DssService.ApiClient.Issues.Models
- **Сборка:** Kontur.DssService.ApiClient.dll

### Синтаксис

```
public class CorrectionInfo
```

### Свойства

Имя	Описание
Comment	Комментарий оператора Удостоверяющего Центра.
Remarks	Комментарий оператора Удостоверяющего Центра.

## Remark — класс

Замечание по исправлению.

- **Пространство имён:** DocsVision.Kontur.DssService.ApiClient.Issues.Models
- **Сборка:** Kontur.DssService.ApiClient.dll

### Синтаксис

```
public class Remark
```

### Свойства

Имя	Описание
Code	Код замечания.
Target	Тип сущности замечания.

Имя	Описание
<code>TargetDescription</code>	Текстовое описание типа сущности замечания.
<code>Message</code>	Описание замечания (включает неразрывные пробелы в виде HTML-кода , может меняться).

## RemarkTargets — перечисление

Тип сущности замечания.

- **Пространство имён:** `DocsVision.Kontur.DssService.ApiClient.Issues.Models`
- **Сборка:** `Kontur.DssService.ApiClient.dll`

### Синтаксис

```
public enum RemarkTargets
```

### Члены

Имя члена	Описание
<code>Unknown</code>	Может быть только возвращаемым типом.
<code>Common</code>	Общие замечания.
<code>Passport</code>	Паспорт РФ.
<code>OtherIdentity</code>	Другой документ удостоверяющий личность.
<code>Snils</code>	СНИЛС.
<code>Egrul</code>	Выписка из ЕГРЮЛ.
<code>ManagementCompanyEgrul</code>	Выписка из ЕГРЮЛ управляющей компании.
<code>Egrip</code>	Выписка из ЕГРИП.
<code>NaturalPersonInn</code>	ИНН физического лица.
<code>SigningAuthority</code>	Подтверждение права подписи.
<code>WarrantWithUseAreas</code>	Доверенность с областями применения.
<code>ReleaseStatement</code>	Заявление на выпуск сертификата.

Имя члена	Описание
<code>ApplicantWarrant</code>	Доверенность на получение.
<code>ApplicantPhoto</code>	Фото обратившегося.
<code>EsiaConfirmation</code>	Подтверждение выпуска НЭП через Госуслуги.

## SmevValidation — класс

Валидация документа в СМЭВ.

- **Пространство имён:** `DocsVision.Kontur.DssService.ApiClient.Issues.Models`
- **Сборка:** `Kontur.DssService.ObjectModel.dll`

### Синтаксис

```
public class SmevValidation
```

### Свойства

Имя	Описание
<code>DocumentType</code>	Тип документа.
<code>DocumentTypeDescription</code>	Текстовое значение типа документа.
<code>State</code>	Состояние.
<code>StateDescription</code>	Текстовое значение состояния.
<code>Errors</code>	Список ошибок от СМЭВ.

## IssueSmevValidationState — перечисление

Статус заявки.

- **Пространство имён:** `DocsVision.Kontur.DssService.ApiClient.Issues.Models`
- **Сборка:** `Kontur.DssService.ApiClient.dll`

### Синтаксис

```
public enum IssueSmevValidationState
```

## Члены

Имя члена	Описание
<code>Unknown</code>	Может быть только возвращаемым типом.
<code>InProgress</code>	Идёт проверка документа в СМЭВ.
<code>Valid</code>	Документ проверен в СМЭВ, ошибки не выявлены.
<code>Invalid</code>	Документ проверен в СМЭВ, выявлены ошибки.

## `SmevValidationError` — класс

Информация об ошибке проверки в СМЭВ

- **Пространство имён:** `DocsVision.Kontur.DssService.ApiClient.Issues.Models`
- **Сборка:** `Kontur.DssService.ObjectModel.dll`

### Синтаксис

```
public class SmevValidationError
```

### Свойства

Имя	Описание
<code>Code</code>	Код ошибки проверки СМЭВ.
<code>CodeDescription</code>	Текстовое описание кода ошибки проверки СМЭВ.
<code>Level</code>	Уровень ошибки проверки в СМЭВ.
<code>LevelDescription</code>	Текстовое описание уровня ошибки проверки в СМЭВ.

## `SmevValidationErrorCode` — перечисление

Код ошибки проверки СМЭВ.

- **Пространство имён:** `DocsVision.Kontur.DssService.ApiClient.Issues.Models`
- **Сборка:** `Kontur.DssService.ObjectModel.dll`

## Синтаксис

```
public enum SmevValidationError
```

### Члены

Имя члена	Описание
Unknown	Код ошибки проверки СМЭВ.
PassportExpired	Неизвестная ошибка, напишите о ней в техническую поддержку.
PassportReplaced	Срок действия паспорта истек.
PassportIssuedWithViolation	Паспорт заменен на новый.
PassportIsSearched	Паспорт выдан с нарушением.
PassportDestroyed	Паспорт числится в розыске.
PassportOwnerDied	Паспорт числится как изъятый или уничтоженный.
PassportDefect	Паспорт недействителен в связи со смертью владельца.
PassportLost	У паспорта стоит отметка о техническом браке, его использование запрещено.
SnilsNotMatchWithPfr	Паспорт утрачен.
InnNotMatchWithFns	По предоставленному номеру СНИЛС, ФИО и дате рождения нет данных в ПФР или данные не совпадают.
NotValidSnilsChecksum	Паспортные данные, указанные в ФНС для этого ИНН, не совпадают с предоставленными.
PfrNotAvailable	Неверное контрольное число у номера СНИЛС.
FnsNotAvailable	Не удалось проверить документ в ПФР.
FmsNotAvailable	Не удалось проверить документ в ФНС.
OrganizationNotFoundInFns	Не удалось проверить документ в ФМС.

## SmevValidationErrorLevel — перечисление

Уровень ошибки проверки в СМЭВ.

- **Пространство имён:** `DocsVision.Kontur.DssService.ApiClient.Issues.Models`
- **Сборка:** `Kontur.DssService.ObjectModel.dll`

### Синтаксис

```
public enum SmevValidationErrorLevel
```

### Члены

Имя члена	Описание
<code>Unknown</code>	Может быть только возвращаемым типом.
<code>Warning</code>	Предупреждение, можно игнорировать. Однако, если не исправить такую ошибку, то заявка может быть отклонена оператором УЦ. Можно загрузить скан-копию документа, чтобы подтвердить, что ошибка ложная.
<code>Fatal</code>	Фатальная ошибка. Заявку с такой ошибкой нельзя отправить на проверку

## IKonturDssService — интерфейс

Сервис подписания Контур.

- **Пространство имён:** `DocsVision.Kontur.DssService.ObjectModel.Services`
- **Сборка:** `DocsVision.M4dRegistry.ObjectModel.dll`

### Синтаксис

```
public interface IKonturDssService
```

### Методы

Имя	Описание
<code>SignDocument(SignRequest, StaffEmployee);</code>	Подписывает файлы документа.
<code>GetStatus(Guid);</code>	Возвращает статус подписания.
<code>ConfirmSignOperation(Guid, string);</code>	Подтверждает операцию подписания.
<code>OperationResult&lt;byte[]&gt; GetResult(Guid);</code>	Возвращает сформированную подпись.

## SignRequest — класс

Данные для подписания.

- **Пространство имён:** `DocsVision.Kontur.DssService.ApiClient.Issues.Models`
- **Сборка:** `Kontur.DssService.ObjectModel.dll`

### Синтаксис

```
public class SignRequest
```

### Свойства

Имя	Описание
<code>DocumentId</code>	Идентификатор документа.
<code>Parts</code>	Части подписи.
<code>Signature</code>	Подпись.

## DocumentSignaturePartInfo — класс

Подписываемая часть документа.

- **Пространство имён:** `DocsVision.Kontur.DssService.ObjectModel.Services`
- **Сборка:** `Kontur.DssService.ObjectModel.dll`

### Синтаксис

```
public class DocumentSignaturePartInfo
```

### Свойства

Имя	Описание
PartType	Тип подписи
FileId	Идентификатор поля

## KonturSignResponse — класс

Ответ на запрос регистрации.

- **Пространство имён:** DocsVision.Kontur.DssService.ApiClient.Crypto.Models
- **Сборка:** Kontur.DssService.ApiClient.dll

### Синтаксис

```
public class KonturSignResponse
```

### Свойства

Имя	Описание
OperationId	Идентификатор операции
ConfirmType	Способ подтверждения операции.
PhoneLastNumbers	Последние 4 цифры номера телефона пользователя. Передается при подтверждении операции через СМС или с помощью Applet на SIM-карте.
AdditionalDssInfo	Дополнительная информация для DSS-сертификата.

## DssSignStatusResponse — класс

Статус выполнения операции подписания.

- **Пространство имён:** DocsVision.Kontur.DssService.ApiClient.Issues.Models
- **Сборка:** Kontur.DssService.ObjectModel.dll

### Синтаксис

```
public class DssSignStatusResponse
```

## Свойства

Имя	Описание
<code>SignStatus</code>	Статус операции подписания.
<code>FileStatuses</code>	Статус файла.

## DssSignStatus — перечисление

Возможный статус подписания.

- **Пространство имён:** `DocsVision.Kontur.DssService.ApiClient.Issues.Models`
- **Сборка:** `Kontur.DssService.ObjectModel.dll`

## Синтаксис

```
public enum DssSignStatus
```

## Члены

Имя члена	Описание
<code>Unknown = 0</code>	Статус неизвестен
<code>WaitingUser = 1</code>	Ожидание пользователя
<code>InProgress = 2,</code>	В процессе
<code>Complete = 3</code>	Завершено
<code>Error = 4</code>	Ошибка