



КриптоАРМ Документы

Руководство разработчика

Модуль для хранения и подписи документов

Данный документ содержит описание программного взаимодействия с модулем КриптоАРМ Документы для CMS 1С-Битрикс для создания интерфейсов взаимодействия с документами и составления бизнес-процессов.

Важно:

Выполняемые функции «Подпись» и «Проверка подписи» реализуются через специализированное программное обеспечение КриптоАРМ ГОСТ, устанавливаемое пользователем самостоятельно на собственном рабочем месте.

Для использования шифрования и подписи алгоритмами ГОСТ на рабочем месте пользователя должен быть предустановлен ГОСТ-криптопровайдер (рекомендуется КриптоПРО CSP 5), а также должно быть обеспечено безопасное соединение по протоколу «HTTPS».

Содержание

Содержание	2
Подключение модуля КриптоАРМ Документы	3
Подключение модуля	3
Получение настроек модуля	3
Функции модуля КриптоАРМ Документы	4
Описание принципа хранения документов в модуле	4
Создание документа	4
Пример добавления документа	4
Пример добавления документа с дополнительным параметром	5
Методы класса Document	6
Изменение свойств документа и сохранение изменений	8
Получение всех загруженных документов	8
Обработка документов собранных в объект класса DocumentCollection	9
Получение документа по идентификатору	10
Получение коллекции документов по имени документа	11
Получение коллекции документов по типу дополнительного параметра	12
Получение коллекции документов по типу и значению дополнительного параметра	13
Получение дополнительных параметров документа по идентификатору	14
Обработка дополнительных параметров, собранных в объект класса PropertyCollection	15
Методы класса Property	16
Добавление дополнительных параметров к документу	16
Изменение дополнительного параметра документа и сохранение изменений	18
Javascript-функции модуля КриптоАРМ Документы	18
Отправка документов на подпись	19
Отправка документов на проверку подписи	22
Снятие блокировки документа	23
Генерация протокола документа	24
Функция «Поделиться документом»	25
Функция отправки документа по почте	26
Удаление документа	26
Пример организации бизнес-процесса с привязкой к заказу	28
Константы модуля КриптоАРМ Документы	29
Таблицы в базе данных	29
Типы документов	29
Статусы документов	29

1. Подключение модуля КриптоАРМ Документы

1.1. Подключение модуля

Для подключения модуля на странице достаточно использовать стандартную функцию подключения модулей в CMS 1С-Битрикс и создать псевдоним для пространства имен.

```
use Trusted\CryptoARM\Docs;  
CModule::IncludeModule("trusted.cryptoarmdocscrp");
```

PHP, пример подключения модуля

Примечание. Название подключаемого модуля зависит от редакции, которую вы установили.

Существующие редакции модулей ядра:

1. trusted.cryptoarmdocsstart
2. trusted.cryptoarmdocsbusiness
3. trusted.cryptoarmdocscrp

В дальнейшем в данном документе будем использовать "trusted.cryptoarmdocscrp"

Модуль автоматически подключит все необходимые для работы классы и функции, в том числе и javascript-библиотеку с высокоуровневыми функциями, предназначенными для взаимодействия с пользователем.

Создание псевдонима для пространства имен модуля не обязательно, но рекомендуется, т.к. позволяет использовать сокращенную запись при вызове функций модуля. Псевдоним позволяет использовать вызовы вида

```
Docs\Database::getDocuments();
```

Вместо

```
Trusted\CryptoARM\Docs\Database::getDocuments();
```

1.2. Получение настроек модуля

Модуль имеет собственный интерфейс для загрузки документов. При его использовании документы автоматически сохраняются в директории, указанной в настройках модуля. Получить путь до этой директории можно с помощью стандартных средств CMS 1С-Битрикс.

```
$DOCUMENTS_DIR = COption::GetOptionString("trusted.cryptoarmdocscrp",  
"DOCUMENTS_DIR", "docs");
```

PHP, пример получения настроек модуля

Эта директория задается относительно корня сайта. По умолчанию модуль использует директорию «docs» в корне сайта.

2. Функции модуля КриптоАРМ Документы

Модуль КриптоАРМ Документы предназначен для хранения и категоризации документов, в том числе организации различных бизнес-процессов, нуждающихся в документообороте и для прикрепления электронной подписи к документам.

Подписание документов осуществляется с помощью сторонних программ, установленных на компьютере пользователя.

2.1. Описание принципа хранения документов в модуле

В представлении модуля документ является записью в базе данных, где хранится вся информация о состоянии документа, его связях с другими документами и путь к самому файлу документа на диске.

Каждый раз, когда происходит подписание документа, создается новая запись о документе, в которой указан идентификатор оригинальной, «родительской» записи, а оригинальная запись отмечается, как имеющая «потомка». Новая запись теперь считается основной. Таким образом документы хранятся в виде структуры «двусвязный список», и для того, чтобы получить список актуальных документов, достаточно выбрать из базы данных только те документы, которые не имеют потомков.

К любому документу можно прикрепить неограниченное количество дополнительных параметров. Эти параметры хранятся в отдельной таблице в базе данных в виде пары «тип» и «значение». Дополнительные параметры позволяют привязывать к документам произвольную информацию и организовывать с ее помощью различные бизнес-процессы.

2.2. Создание документа

Функция создания документа добавляет новую запись в базу данных. Файл документа уже должен присутствовать на сервере и лежать в пределах корня сайта.

```
Docs\Utils::createDocument($path, $properties);
```

PHP, описание функции создания документа

Тип	Аргумент	Описание
string	\$path	Путь к файлу относительно корня сайта.
PropertyCollection	\$properties	Произвольные дополнительные свойства документа. Необязательный аргумент, по умолчанию NULL.

2.2.1. Пример добавления документа

Добавление нового документа в базу данных.

```
Docs\Utils::createDocument("/docs/document.doc");
```

PHP, пример вызова функции создания документа

После создания записи о документе в базе данных функция вернет получившийся объект класса Document.

```
Trusted\CryptoARM\Docs\Document Object
(
    [id:protected] => 459
    [name:protected] => document.doc
    [path:protected] => /docs/5qwe6324/document.doc
    [type:protected] => 0
    [signType:protected] => 0
    [status:protected] => 0
    [parentId:protected] =>
    [childId:protected] =>
    [originalId:protected] => 459
    [hash:protected] => 518214550347e1d0b9f2ba40d3a4b950
    [signatures:protected] =>
    [signers:protected] => 1
    [blockBy:protected] => null
    [blockToken:protected] => null
    [blockTime:protected] => 2020-02-18 15:28:47
    [created:protected] =>
    [requires:public] => Trusted\CryptoARM\Docs\RequireCollection Object
    (
        [item_:protected]=> Array
        (
        )
    )
    [properties:public] => Trusted\CryptoARM\Docs\PropertyCollection Object
    (
        [items_:protected] => Array
        (
        )
    )
)
```

PHP, пример объекта, который возвращает функция создания документа

- Документу автоматически будет назначен уникальный идентификатор
- Тип и статус будут равны значениям по умолчанию (см. [п.п. 3.2.](#) и [п.п. 3.3.](#))
- Документ не будет иметь потомка и сам не будет считаться родителем
- Список дополнительных параметров документа будет пустым

2.2.2. Пример добавления документа с дополнительным параметром

Дополнительные параметры документа могут быть использованы для привязки документа к определенному пользователю, заказу или любому другому параметру бизнес-процесса.

Рассмотрим пример привязки документа к определенному пользователю.

```
$props = new Docs\PropertyCollection();
$props->add(new Docs\Property("USER", "254"));
Docs\Utils::createDocument("/docs/document.doc", $props);
```

PHP, пример вызова функции создания документа с дополнительным параметром

После создания записи о документе в базе данных функция вернет получившийся объект класса Document.

```
Trusted\CryptoARM\Docs\Document Object
(
    [id:protected] => 557
    [name:protected] => document.doc
    [path:protected] => /docs/document.doc
    [type:protected] => 0
    [status:protected] => 0
    [signers:protected] =>
    [parentId:protected] =>
    [childId:protected] =>
    [created:protected] =>
    [properties] => Trusted\CryptoARM\Docs\PropertyCollection Object
        (
            [items_:protected] => Array
                (
                    [0] => Trusted\CryptoARM\Docs\Property Object
                        (
                            [id:protected] => 266
                            [documentId:protected] => 557
                            [type:protected] => USER
                            [value:protected] => 254
                        )
                )
            )
        )
    )
)
```

PHP, пример объекта, который возвращает функция создания документа при указании дополнительного параметра

- Документу автоматически будет назначен уникальный идентификатор
- Тип и статус будут равны значениям по умолчанию (см. [п.п. 3.2.](#) и [п.п. 3.3.](#))
- Документ не будет иметь потомка и сам не будет считаться родителем
- Список дополнительных параметров документа будет содержать один дополнительный параметр, соответствующий указанным в вызове функции аргументам.

Тип и значение параметра могут быть произвольными строками, но длина типа параметра ограничена 50 символами, а значения - 255 символами.

К документу можно прикрепить любое количество дополнительных параметров. Дополнительные параметры можно прикреплять к документу после его создания (см. [п.п. 2.9.](#)).

2.2.3. Методы класса Document

Объект класса Document имеет ряд методов упрощающих доступ к его свойствам.

Название метода	Тип возвращаемого значения	Описание метода
getId()	integer	Возвращает идентификатор документа.
getName()	string	Возвращает название документа.
getPath()	string	Возвращает путь к документу относительно корня сайта.
getType()	integer	Возвращает тип документа.
getStatus()	integer	Возвращает статус документа.
getSigners()	string	Возвращает список пользователей, подписавших документ
getSignersToArray()	array	Возвращает список пользователей, подписавших документ в виде массива
getSignatures()	string	Возвращает информацию о подписи документа.
getSignaturesToArray())	array	Возвращает информацию о подписи документа в виде массива.
getParentId()	integer	Возвращает идентификатор документа-родителя.
getParent()	Document	Возвращает документ-родитель.
getChildId()	integer	Возвращает идентификатор документа-потомка.
getChild()	Document	Возвращает документ-потомок.
getCreated()	string	Возвращает дату создания документа.
getBlockBy()	integer	Возвращает пользователя, который отправил документ на подпись (только для заблокированных документов)
getBlockToken()	string	Возвращает токен, используемый для проверки загрузки подписанного документа (только для заблокированных документов)
getBlockTime()	string	Возвращает время, когда документ был отправлен на подпись (только для заблокированных документов)
getHash()	string	Возвращает md5-хэш файла документа

getLastDocument()	Document	Возвращает последнего потомка документа.
getProperties()	PropertyCollection	Возвращает дополнительные свойства документа.
checkFile()	boolean	Проверяет наличие файла документа в хранилище.
getOwner()	integer	Возвращает владельца документа (если у документа есть владелец)
getSignType()	integer	Возвращает тип подписи
getOriginalId	integer	Возвращает Id первого документа в цепочке
getFirstParent	Document	Возвращает первый оригинальный документ в цепочке
getSignaturesToTable	string	Возвращает информацию о подписи документа в виде таблицы.
getRequires	RequiresCollection	Возвращает коллекцию запросов на подпись
getFullPath	string	Физический путь на сервере
getHtmlPath	string	Относительный путь до файла
getUrl	string	Ссылка до файла

2.2.4. Изменение свойств документа и сохранение изменений

Иногда возникает необходимость изменить какие-то свойства документа, например, если документы были перенесены в другую директорию.

Чтобы изменить путь до документа хранящийся в записи в базе данных, достаточно получить нужный документ и использовать один из его методов чтобы получить значение свойства пути.

```
$doc = Docs\Database::getDocumentById(557);
$doc_path = $doc->getPath();
```

PHP, пример получения объекта Document по идентификатору и получения пути к файлу этого документа с помощью метода

После того как необходимая информация была получена и изменена, ее нужно поместить обратно в объект Document и сохранить его, чтобы обновить информацию в базе данных.

```
$doc_path .= "/new/path";
```

```
$doc->setPath($doc_path);  
$doc->save();
```

PHP, пример указания нового пути документа и сохранения изменений в базе данных

2.3. Получение всех загруженных документов

Функция получения всех загруженных документов возвращает все конечные документы в цепочке «родитель-потомок», т.е. документы, не имеющие потомков (см. [п.п. 2.1.](#)).

```
Docs\Database::getDocuments();
```

PHP, пример вызова функции получения всех документов

Функция вернет объект класса DocumentCollection, содержащий объекты класса Document.

```
Trusted\CryptoARM\Docs\DocumentCollection Object  
(  
    [items_:protected] => Array  
        (  
            [0] => Trusted\CryptoARM\Docs\Document Object  
                (  
                    [id:protected] => 456  
                    [name:protected] => document01.txt  
                    [path:protected] => /docs/document01.txt  
                    [type:protected] => 0  
                    [status:protected] => 0  
                    [signers:protected] =>  
                    [parentId:protected] =>  
                    [childId:protected] =>  
                    [created:protected] => 2018-03-14 15:44:34  
                )  
            [1] => Trusted\CryptoARM\Docs\Document Object  
                (  
                    [id:protected] => 457  
                    [name:protected] => document02.txt  
                    [path:protected] => /docs/document02.txt  
                    [type:protected] => 0  
                    [status:protected] => 0  
                    [signers:protected] =>  
                    [parentId:protected] =>  
                    [childId:protected] =>  
                    [created:protected] => 2018-03-14 15:44:43  
                )  
            [2] => Trusted\CryptoARM\Docs\Document Object  
                (  

```

```

        [id:protected] => 460
        [name:protected] => document03.txt
        [path:protected] => /docs/document03.txt
        [type:protected] => 0
        [status:protected] => 0
        [signers:protected] =>
        [parentId:protected] =>
        [childId:protected] =>
        [created:protected] => 2018-03-14 16:27:01
    )
)
)

```

PHP, пример объекта возвращаемого функцией получения всех документов.

2.3.1. Обработка документов собранных в объект класса DocumentCollection

Для преобразования объекта DocumentCollection в массив объектов Document используется метод `getList`. Это позволяет проходить по коллекциям документов с помощью стандартной конструкции *foreach* языка PHP.

```
$doc_array = Docs\Database::getDocuments()->getList();
```

PHP, пример преобразования объекта класса DocumentCollection в массив объектов Document.

```

Array
(
    [0] => Trusted\CryptoARM\Docs\Document Object
        (
            [id:protected] => 456
            [name:protected] => document01.txt
            [path:protected] => /docs/document01.txt
            [type:protected] => 0
            [status:protected] => 0
            [signers:protected] =>
            [parentId:protected] =>
            [childId:protected] =>
            [created:protected] => 2018-03-14 15:44:34
        )

    [1] => Trusted\CryptoARM\Docs\Document Object
        (
            [id:protected] => 457
            [name:protected] => document02.txt
            [path:protected] => /docs/document02.txt
            [type:protected] => 0
            [status:protected] => 0
            [signers:protected] =>
            [parentId:protected] =>
        )
)

```

```

        [childId:protected] =>
        [created:protected] => 2018-03-14 15:44:43
    )

[2] => Trusted\CryptoARM\Docs\Document Object
(
    [id:protected] => 460
    [name:protected] => document03.txt
    [path:protected] => /docs/document03.txt
    [type:protected] => 0
    [status:protected] => 0
    [signers:protected] =>
    [parentId:protected] =>
    [childId:protected] =>
    [created:protected] => 2018-03-14 16:27:01
)

```

PHP, пример массива документов, полученных с помощью метода getList

```

$doc_array = Docs\Database::getDocuments()->getList();
foreach ($doc_array as $doc) {
    echo $doc->getId();
}

```

PHP, пример обхода коллекции документов с помощью конструкции foreach

2.4. Получение документа по идентификатору

Уникальный идентификатор документа представляет собой число, однозначно указывающее на запись о документе в базе данных. Имея идентификатор, можно получить полную информацию о документе.

```
Docs\Database::getDocumentById($id);
```

PHP, описание функции получения документа по идентификатору

Тип	Аргумент	Описание
integer	\$id	Путь к файлу относительно корня сайта.

```
Docs\Database::getDocumentById(463);
```

PHP, пример получения документа по идентификатору

```

Trusted\CryptoARM\Docs\Document Object
(
    [id:protected] => 463
    [name:protected] => document01.txt
    [path:protected] => /docs/document01.txt
    [type:protected] => 0

```

```
[status:protected] => 0
[signers:protected] =>
[parentId:protected] =>
[childId:protected] =>
[created:protected] => 2018-03-14 16:40:48
)
```

PHP, пример результата получения документа по идентификатору

2.5. Получение коллекции документов по имени документа

Функция получения документа по имени всегда возвращает коллекцию документов (объект `DocumentCollection`), т.к. в базе данных может одновременно находиться несколько разных документов с одинаковыми именами.

```
Docs\Database::getDocumentsByName($name);
```

PHP, описание функции получения коллекции документов по имени документа

Тип	Аргумент	Описание
string	\$name	Название документа

```
Docs\Database::getDocumentsByName("document1.txt");
```

PHP, пример вызова функции получения коллекции документов по имени документа

```
Trusted\CryptoARM\Docs\DocumentCollection Object
(
    [items_:protected] => Array
        (
            [0] => Trusted\CryptoARM\Docs\Document Object
                (
                    [id:protected] => 463
                    [name:protected] => document01.txt
                    [path:protected] => /docs/92660071fc/document01.txt
                    [type:protected] => 0
                    [status:protected] => 0
                    [signers:protected] =>
                    [parentId:protected] =>
                    [childId:protected] =>
                    [created:protected] => 2018-03-14 16:40:48
                )

            [1] => Trusted\CryptoARM\Docs\Document Object
                (
                    [id:protected] => 471
                    [name:protected] => document01.txt
                    [path:protected] => /docs/92ce6aa46f/document01.txt
                    [type:protected] => 0
                )
        )
    )
```

```

        [status:protected] => 0
        [signers:protected] =>
        [parentId:protected] =>
        [childId:protected] =>
        [created:protected] => 2018-03-14 17:08:38
    )
)
)

```

PHP, пример коллекции документов полученных в результате получения документа по имени

2.6. Получение коллекции документов по типу дополнительного параметра

Функция получения документов по типу дополнительного параметра позволяет получать коллекции документов по любому произвольному типу дополнительному параметра.

```
Docs\Database::getDocumentsByPropertyType($type);
```

PHP, описание функции получения коллекции документов по типу дополнительного параметра

Тип	Аргумент	Описание
string	\$type	Строка, задающая тип дополнительного параметра

```
Docs\Database::getDocumentsByPropertyType("USER");
```

PHP, пример получения коллекции всех документов с привязкой к пользователю

Функция вернет коллекцию конечных документов, к которым прикреплен заданный тип дополнительного параметра, независимо от его значения. После получения коллекции документов их можно обработать один за другим (см. [п.п. 2.3.1.](#)) и получить полный список их дополнительных параметров с помощью метода `getProperties` (см. [п.п. 2.2.3.](#)).

```

Trusted\CryptoARM\Docs\DocumentCollection Object
(
    [items_:protected] => Array
        (
            [0] => Trusted\CryptoARM\Docs\Document Object
                (
                    [id:protected] => 42
                    [name:protected] => test01.txt
                    [path:protected] => /docs/5b050696a0371/test01.txt
                    [type:protected] => 0
                    [status:protected] => 0
                    [signers:protected] =>
                    [parentId:protected] =>
                    [childId:protected] =>
                    [created:protected] => 2018-05-23 09:13:42
                )
            )
        )
    )

```

```

    )

    [1] => Trusted\CryptoARM\Docs\Document Object
    (
        [id:protected] => 43
        [name:protected] => test09.txt
        [path:protected] => /docs/5b0506a7581ce/test09.txt
        [type:protected] => 0
        [status:protected] => 0
        [signers:protected] =>
        [parentId:protected] =>
        [childId:protected] =>
        [created:protected] => 2018-05-23 09:13:59
    )

)

)

```

PHP, пример результата получения коллекции документов по типу дополнительного параметра

2.7. Получение коллекции документов по типу и значению дополнительного параметра

Функция получения документов по типу и значению дополнительного параметра позволяет получать коллекции документов, к которым были прикреплены дополнительные параметры определенного типа и с определенным значением.

```
Docs\Database::getDocumentsByPropertyTypeAndValue($type, $value);
```

PHP, описание функции получения коллекции документов по типу и значению дополнительного параметра

Тип	Аргумент	Описание
string	\$type	Строка, задающая тип дополнительного параметра
string	\$value	Строка, задающая значение дополнительного параметра

```
Docs\Database::getDocumentsByPropertyTypeAndValue("USER", "47");
```

PHP, пример получения коллекции всех документов с привязкой к пользователю с ID=47

Функция вернет все конечные документы, к которым прикреплен заданный тип дополнительного параметра с заданным значением. После получения коллекции документов их можно обработать один за другим (см. [п.п. 2.3.1.](#)) и получить полный список их дополнительных параметров с помощью метода `getProperties` (см. [п.п. 2.2.3.](#)).

```
Trusted\CryptoARM\Docs\DocumentCollection Object
```

```
(
  [items_:protected] => Array
  (
    [0] => Trusted\CryptoARM\Docs\Document Object
    (
      [id:protected] => 42
      [name:protected] => test01.txt
      [path:protected] => /docs/5b050696a0371/test01.txt
      [type:protected] => 0
      [status:protected] => 0
      [signers:protected] =>
      [parentId:protected] =>
      [childId:protected] =>
      [created:protected] => 2018-05-23 09:13:42
    )
  )
)
```

PHP, пример результата получения коллекции документов по типу и значению дополнительного параметра

2.8. Получение дополнительных параметров документа по идентификатору

Функция получения дополнительных параметров документа всегда возвращает коллекцию параметров (объект PropertyCollection), т.к. в базе данных может одновременно находиться несколько дополнительных параметров, привязанных к одному документу.

```
Docs\Database::getPropertiesByDocumentId($documentId, $tableName);
```

PHP, описание функции получения дополнительных параметров документа по идентификатору документа

Тип	Аргумент	Описание
integer	\$documentId	Уникальный идентификатор документа
string	\$tableName	Таблица, хранящая дополнительные параметры. Необязательный параметр, по умолчанию <code>DB_TABLE_PROPERTY</code> (см. п.п. 3.1.)

```
Docs\Database::getPropertiesByDocumentID(557);
```

PHP, пример получения дополнительных параметров документа по идентификатору документа

```
Trusted\CryptoARM\Docs\PropertyCollection Object
(
  [items_:protected] => Array
  (
    [0] => Trusted\CryptoARM\Docs\Property Object
    (
      [id:protected] => 266
    )
  )
)
```



```

        [documentId:protected] => 557
        [type:protected] => USER
        [value:protected] => 254
    )
)
)

```

PHP, пример результата получения дополнительных параметров документа

2.8.1. Обработка дополнительных параметров, собранных в объект класса PropertyCollection

Для преобразования объекта PropertyCollection в массив объектов Property используется метод getList. Это позволяет проходить по коллекциям дополнительных параметров с помощью стандартной конструкции *foreach* языка PHP.

```
$doc_props = Docs\Database::getPropertiesByDocumentID(625)->getList();
```

PHP, пример преобразования объекта класса PropertyCollection в массив объектов Property

```

Trusted\CryptoARM\Docs\PropertyCollection Object
(
    [items_:protected] => Array
    (
        [0] => Trusted\CryptoARM\Docs\Property Object
        (
            [id:protected] => 516
            [documentId:protected] => 625
            [type:protected] => ORDER
            [value:protected] => 1
        )

        [1] => Trusted\CryptoARM\Docs\Property Object
        (
            [id:protected] => 517
            [documentId:protected] => 625
            [type:protected] => ROLES
            [value:protected] => NONE
        )
    )
)

```

PHP, пример массива дополнительных параметров, полученных с помощью метода getList

```

$doc_props = Docs\Database::getPropertiesByDocumentID(625)->getList();
foreach ($doc_props as $prop) {
    echo $prop->getValue();
}

```

PHP, пример обхода коллекции документов с помощью конструкции foreach

2.8.2. Методы класса Property

Объект класса Property имеет ряд методов упрощающих доступ к его свойствам.

Название метода	Тип возвращаемого значения	Описание метода
getId()	integer	Возвращает уникальный идентификатор дополнительного параметра. Не путать с идентификатором документа!
getDocumentId()	integer	Возвращает идентификатор документа, к которому привязан данный дополнительный параметр.
getType()	string	Возвращает тип дополнительного параметра.
getValue()	string	Возвращает значение дополнительного параметра.

2.9. Добавление дополнительных параметров к документу

Некоторым бизнес-процессам для реализации необходимо прикреплять к документу более одного дополнительного параметра. После создания документа к нему можно прикрепить неограниченное количество дополнительных параметров.

Для этого нужно получить объект представляющий документ, а затем получить коллекцию его дополнительных параметров.

```
$doc = Docs\Database::getDocumentById(557);
$doc_props = $doc->getProperties();
```

PHP, получение документа и его дополнительных параметров

Дополнительные параметры будут доступны в виде объекта класса PropertyCollection.

```
Trusted\CryptoARM\Docs\PropertyCollection Object
(
    [items_:protected] => Array
    (
        [0] => Trusted\CryptoARM\Docs\Property Object
        (
            [id:protected] => 266
            [documentId:protected] => 557
            [type:protected] => USER
            [value:protected] => 254
        )
    )
)
```

```
)
    )
)
```

PHP, пример результата получения коллекции дополнительных параметров

Добавим к коллекции дополнительный параметр. Необходимо указать тип и значение параметра.

```
$doc_props->add(new Property("NEW PROP", "NEW VALUE"));
```

PHP, добавление нового дополнительного параметра к коллекции дополнительных параметров документа

Теперь коллекция дополнительных параметров содержит два параметра.

```
Trusted\CryptoARM\Docs\PropertyCollection Object
(
    [items_:protected] => Array
    (
        [0] => Trusted\CryptoARM\Docs\Property Object
        (
            [id:protected] => 266
            [documentId:protected] => 557
            [type:protected] => USER
            [value:protected] => 254
        )

        [1] => Trusted\CryptoARM\Docs\Property Object
        (
            [id:protected] =>
            [documentId:protected] => 557
            [type:protected] => NEW PROP
            [value:protected] => NEW VALUE
        )
    )
)
```

PHP, пример коллекции дополнительных параметров с двумя параметрами

Остается только вызвать сохранение документа в базе данных. При этом автоматически будут сохранены все его дополнительные параметры.

```
$doc->save();
```

2.9.1. Изменение дополнительного параметра документа и сохранение изменений

При выполнении бизнес-процесса часто возникает необходимость изменить какие-то дополнительные параметры документа, например, когда пользователь, связанный с документом, оплатил заказ или подписал документ.

Чтобы изменить дополнительные параметры документа хранящиеся в записи в базе данных, достаточно получить нужный документ и использовать один из его методов, чтобы получить коллекцию его дополнительных параметров.

```
$doc = Docs\Database::getDocumentById(557);
$doc_props = $doc->getProperties();
```

PHP, получение документа и его дополнительных параметров

Чтобы получить конкретный дополнительный параметр, надо воспользоваться методом «getItemByType» класса PropertyCollection.

```
$user_prop = $doc_props->getPropByType("USER");
```

PHP, получение конкретного дополнительного параметра документа

После того, как необходимый дополнительный параметр был получен, его можно изменить и сохранить содержащий его документ, чтобы обновить информацию в базе данных.

```
$user_prop->setValue("732");
$doc->save();
```

PHP, пример задания нового значения дополнительного параметра и сохранения изменений в базе данных

2.10. Javascript-функции модуля КриптоАРМ Документы

Большая часть взаимодействия с пользователем в модуле организована в виде javascript-функций. Такие операции, как подписание документов, проверка подписи, запросы на удаление документов выполняются запуском javascript-функций, которые разработчик может назначить на произвольные элементы интерфейса. Для того чтобы не затронуть другие объявленные функции, все функции модуля собраны в одном объекте **trustedCA**.

2.10.1. Отправка документов на подпись

Отправка документа на подпись вызывается запуском javascript-функции. Для этого на компьютере пользователя должна быть установлена и запущена программа КриптоАРМ ГОСТ.

```
trustedCA.sign(ids, extra, onSuccess, onFailure);
```

Javascript, описание функции отправки документа на подпись

Тип	Аргумент	Описание
array	ids	Массив уникальных идентификаторов документа
object	extra	Дополнительные параметры подписи. Необязательный параметр.
function	onSuccess	Callback который будет вызван если функция успешно выполнится.

Необязательный параметр.

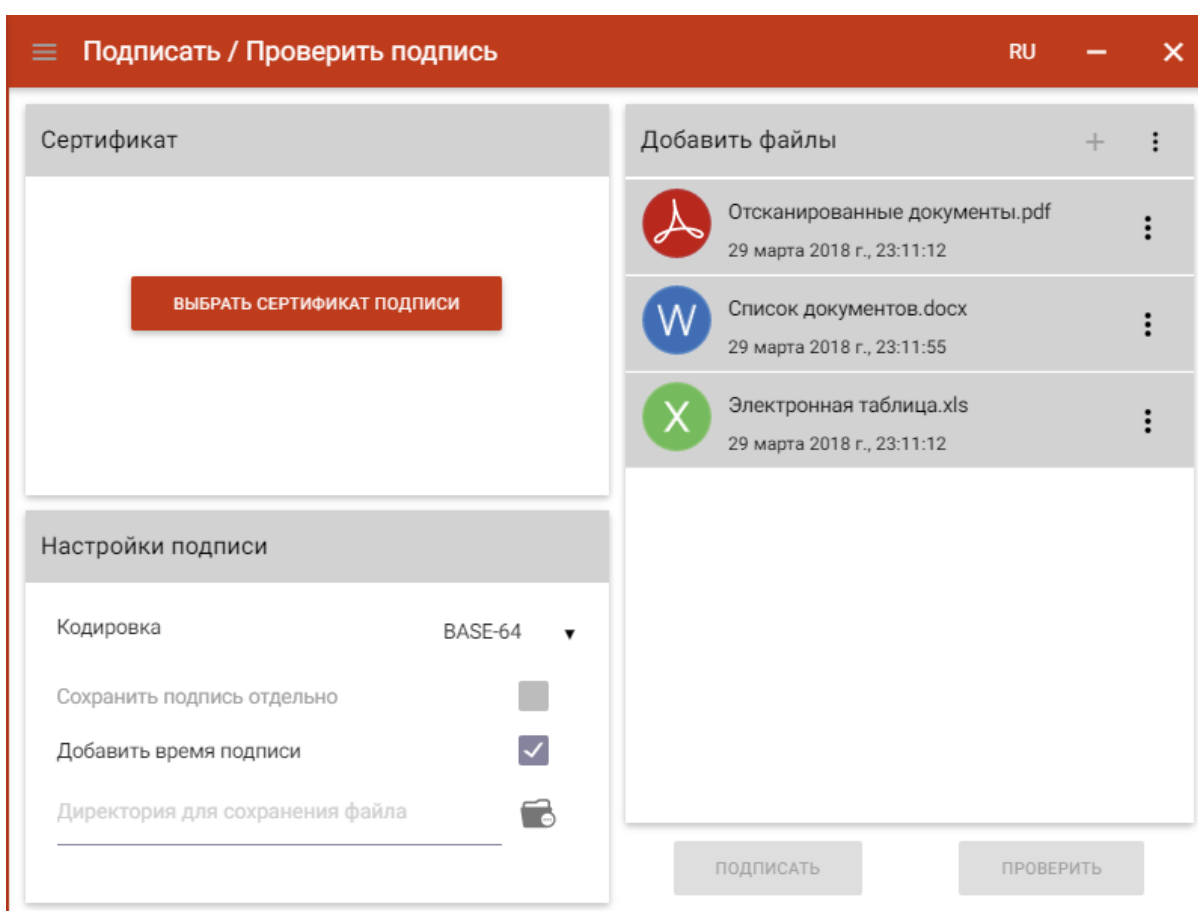
function	onFailure	Callback который будет вызван если функция завершится с ошибкой. Необязательный параметр.
----------	-----------	--

Для отправки документов на подпись достаточно знать их уникальные идентификаторы.

```
trustedCA.sign([545, 546, 547])
```

Javascript, пример отправки документов на подпись

Отправленные файлы появятся в окне запущенной программы КриптоАРМ ГОСТ.



Отправленным на подпись документам устанавливается статус DOC_STATUS_BLOCKED (см. [п.п. 3.3.](#)). Заблокированный документ нельзя повторно отправить на подпись, пока он не вернется с подписью или не будет разблокирован вручную (см. [п.п. 2.10.4.](#)).

Параметр extra является необязательным, и используется для передачи дополнительных свойств, например, когда надо указать роль подписывающего (см. [п.п. 2.11.](#)).

Функция отправки документа на подпись после запуска возвращает подробную информацию о результате выполнения. Эта информация может быть использована, чтобы

показывать пользователю предупреждения о невозможности отправить некоторые документы на подпись.

```
{
  "success":true,
  "message":"Some documents were sent for signing",
  "docsOk":[
    {
      "name":"test02.txt",

"url":"https:\\\\www.bitrix.org\\bitrix\\components\\trusted\\docs\\ajax.php?
command=content&id=71",
      "id":69
    }
  ],
  "docsNotFound":[
    70,
    71
  ],
  "docsFileNotFound":[
    {
      "filename":"test08.txt",
      "id":72
    }
  ],
  "docsBlocked":[
    {
      "filename":"test09.txt",
      "id":73
    }
  ]
}
```

Javascript, пример значения возвращаемого функцией подписи документа, включающий большое количество предупреждений о невозможности подписать некоторые из документов.

Возвращаемое значение	Описание
success	Общий результат выполнения функции. Равен <i>true</i> , если хотя бы один из файлов был отправлен на подпись.
message	Текстовое сообщение, описывающее результат выполнения функции
docsOk	Массив с информацией о документах, которые были отправлены на подпись
docsNotFound	Массив с идентификаторами документов, которые не были найдены в базе данных

docsNoAccess	Массив с идентификаторами документов, к которым у текущего пользователя нет доступа
docsFileNotFound	Массив с информацией о документах, файлы которых не удалось найти в хранилище
docsBlocked	Массив с информацией о документах, которые не удалось отправить на подпись, потому что они были заблокированы предыдущей операцией.
docsUnsigned	Массив с информацией о документах, которые не были подписаны

Для вывода интерфейса для подписи документов нужно выбрать необходимые идентификаторы документов и подставить их в вызов javascript-функции.

```
$all_docs = Docs\Database::getDocuments();
$all_docs_array = $all_docs->getList();
$all_ids = array();
foreach ($all_docs_array as $doc) {
    $doc_id = $doc->getId();
    $doc_name = $doc->getName();
    echo "<input type='button' value='" . $doc_id . " - " . $doc_name . "'
onclick='sign([" . $doc_id . "])'>";
    $all_ids[] = $doc_id;
}
echo "<input type='button' value = 'SIGN ALL' onclick='trustedCA.sign(" .
json_encode($all_ids) . ")'>"
```

PHP, пример вывода интерфейса для подписи документов

2.10.2. Отправка документов на проверку подписи

Отправка документов на проверку подписи вызывается запуском javascript-функции. Для этого на компьютере пользователя должна быть установлена и запущена программа КриптоАРМ ГОСТ.

```
trustedCA.verify(ids);
```

Javascript, описание функции отправки документов на проверку подписи

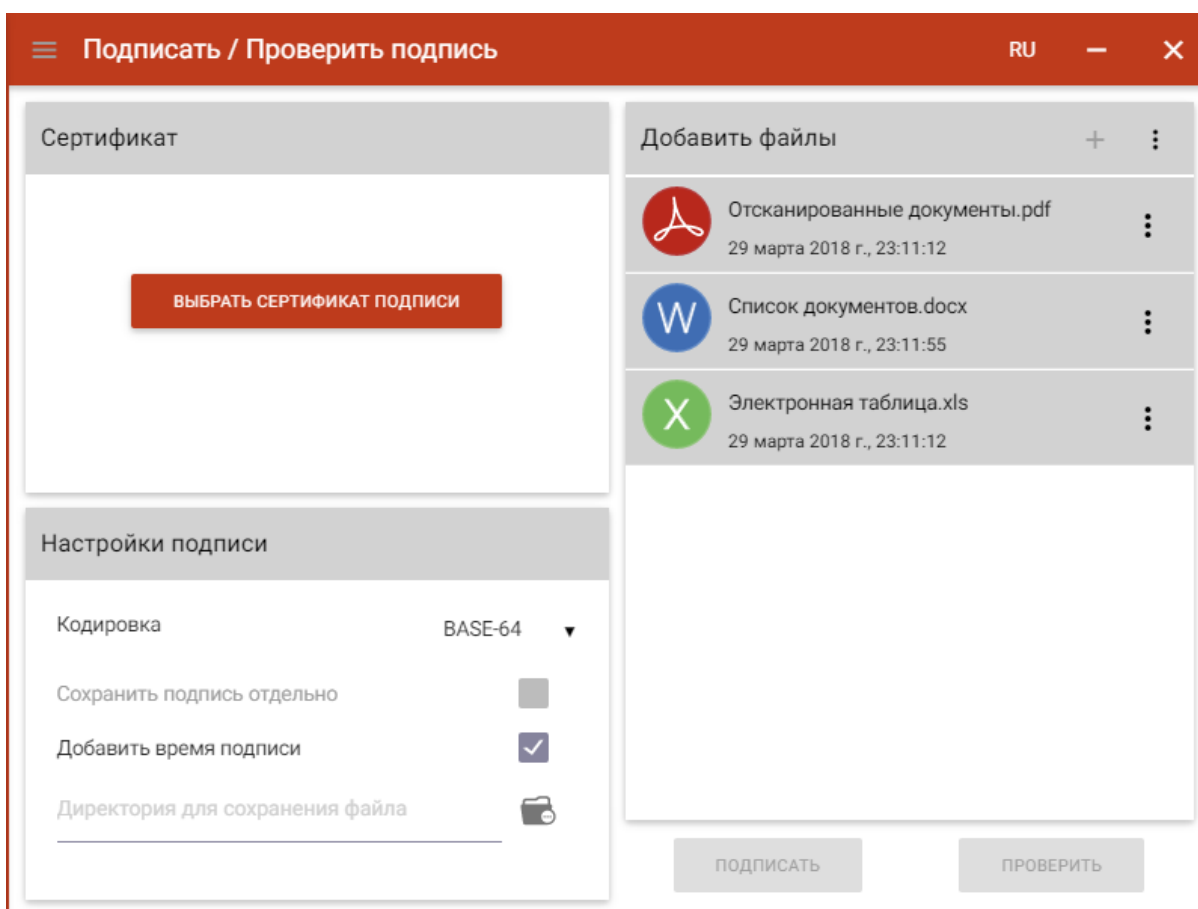
Тип	Аргумент	Описание
array	ids	Массив уникальных идентификаторов документа

Для отправки документов на проверку подписи достаточно знать их уникальные идентификаторы.

```
trustedCA.verify([545, 546, 547])
```

Javascript, пример отправки документов на проверку подписи

Отправленные файлы появятся в окне запущенной программы КриптоАРМ ГОСТ.



Кнопка «Подписать» будет неактивна для документов, отправленных на проверку подписи.

Для вывода интерфейса для отправки документов на проверку подписи нужно выбрать необходимые идентификаторы документов и подставить их в вызов javascript-функции.

```
$all_docs = Docs\Database::getDocuments();
$all_docs_array = $all_docs->getList();
$all_ids = array();
foreach($all_docs_array as $doc) {
    $doc_id = $doc->getId();
    $doc_name = $doc->getName();
    echo "<input type='button' value='" . $doc_id . " - " . $doc_name . "'
onclick='verify([" . $doc_id . "])'>";
    $all_ids[] = $doc_id;
}
echo "<input type='button' value='VERIFY ALL' onclick='trustedCA.verify(" .
json_encode($all_ids) . ")'>";
```

PHP, пример вывода интерфейса для отправки документов на проверку подписи

2.10.3. Снятие блокировки документа

Если документ по какой-то причине не удалось подписать (например из-за ошибки в программе КриптоАРМ ГОСТ), то блокировку файла можно снять самостоятельно.

```
trustedCA.unblock(ids);
```

Javascript, описание функции снятия блокировки документов

Тип	Аргумент	Описание
array	ids	Массив уникальных идентификаторов документа
function	onSuccess	Callback, который будет вызван, если функция успешно выполнится. Необязательный параметр.
function	onFailure	Callback, который будет вызван, если функция завершится с ошибкой. Необязательный параметр.

Для снятия блокировки документов достаточно знать их уникальные идентификаторы.

```
trustedCA.unblock([545, 546])
```

Javascript, пример вызова функции снятия блокировки документов

Для вывода на экран интерфейса для снятия блокировки документов нужно выбрать необходимые идентификаторы документов и подставить их в вызов javascript-функции.

```
$all_docs = Docs\Database::getDocuments();
$all_docs_array = $all_docs->getList();
$all_ids = array();
foreach($all_docs_array as $doc) {
    $doc_id = $doc->getId();
    $doc_name = $doc->getName();
    echo "<input type='button' value='" . $doc_id . " - " . $doc_name . "'
onclick='unblock([" . $doc_id . "])'>";
    $all_ids[] = $doc_id;
}
echo "<input type='button' value='UNBLOCK ALL' onclick='trustedCA.unblock(" .
json_encode($all_ids) . ")'>";
```

PHP, пример вывода интерфейса для снятия блокировки документов

2.10.4. Генерация протокола документа

Протокол документа содержит подробную информацию о документе (включая информацию о подписи документа) в пригодном для печати виде. Запрос на генерацию протокола документа выполняется вызовом javascript-функции.

```
trustedCA.protocol(id)
```

Javascript, описание функции генерации протокола документа

Тип	Аргумент	Описание
integer	id	Уникальный идентификатор документа

Для генерации протокола достаточно знать идентификатор документа.

```
trustedCA.protocol(545)
```

Javascript, пример вызова функции генерации протокола документа

Если функция генерации протокола завершится успешно, браузер пользователя начнет скачивание pdf-документа.

2.10.5. Функция «Поделиться документом»

Некоторые интерфейсы модуля показывают только личные документы текущего пользователя, т.е. такие, у которых установлен дополнительный параметр USER. Пользователи могут поделиться своими личными документами с другими пользователями Битрикс при помощи вызова функции share.

```
trustedCA.share(ids, email, level)
```

javascript, описание функции «поделиться документом»

Тип	Аргумент	Описание
array	ids	Массив уникальных идентификаторов документа
string	email	Email пользователя Битрикс, который получит доступ к документу
string	level	Уровень доступа, на котором пользователь получит доступ к документу. Необязательный параметр, по умолчанию SHARE_READ .

Предусмотрено два уровня доступа - SHARE_READ и SHARE_SIGN. Первый уровень просто дает другому пользователю просматривать документ, второй позволяет ему отправлять документ на подпись.

```
trustedCA.share([12], "mail@example.com", "SHARE_SIGN")
```

javascript, пример вызова функции «поделиться документом»

Для удобства работы с данным функционалом модуль предоставляет еще одну функцию, которая предварительно выводит на экран запрос от пользователя на указание адреса электронной почты.

```
trustedCA.promptAndShare(ids, level)
```

javascript, пример вызова функции, запрашивающей у пользователя адрес электронной почты

2.10.6. Функция отправки документа по почте

Функция позволяет отправить документ на произвольный адрес электронной почты.

```
trustedCA.sendEmail(ids, event, eventFields, templateId)
```

javascript, описание функции отправки документа по почте

Тип	Аргумент	Описание
array	ids	Массив уникальных идентификаторов документа
string	event	Идентификатор почтового события Битрикс
object	eventFields	Ассоциативный массив параметров, которые будут переданы в шаблон письма
integer	templateId	Идентификатор шаблона письма

Модуль автоматически устанавливает три почтовых события и шаблона (с префиксом `TR_CA_DOCS`).

```
let fields = {}
fields.EMAIL = "mail@example.com"
trustedCA.sendEmail([12], "TR_CA_DOCS_MAIL_SHARE", fields, 160)
```

javascript, пример вызова функции отправки документа по почте

Для удобства работы с данным функционалом модуль предоставляет еще одну функцию, которая предварительно выводит на экран запрос от пользователя на указание адреса электронной почты.

```
trustedCA.promptAndSendEmail(ids, event, eventFields, templateId)
```

javascript, пример вызова функции, запрашивающей у пользователя адрес электронной почты

2.10.7. Удаление документа

Запрос на удаление документа осуществляется вызовом javascript-функции. При удалении документа удаляется запись о документе в базе данных и файл документа на диске. Если документ имеет подпись, будут удалены и оригинальный файл и подписанный.

```
trustedCA.remove(ids, force, onSuccess, onFailure);
```

javascript, описание функции удаления документа

Тип	Аргумент	Описание
array	ids	Массив уникальных идентификаторов документа
boolean	force	Удалять документ без вывода запроса на подтверждение. Необязательный параметр. По умолчанию <i>false</i> .

function	onSuccess	Callback, который будет вызван, если функция успешно выполнится. Необязательный параметр.
----------	-----------	--

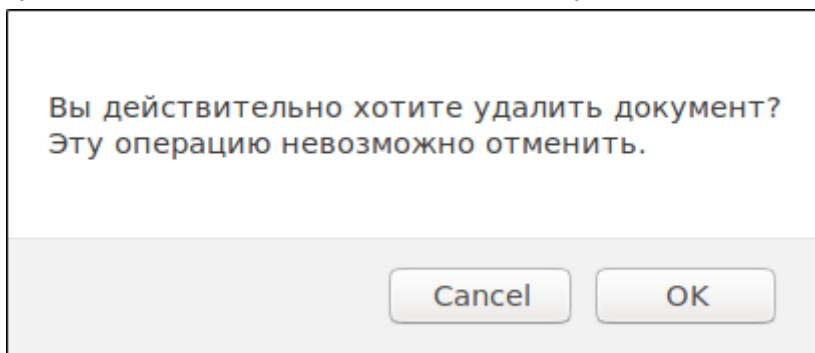
function	onFailure	Callback, который будет вызван, если функция завершится с ошибкой. Необязательный параметр.
----------	-----------	--

Для установки блокировки документов достаточно знать их уникальные идентификаторы.

```
remove([561, 562])
```

javascript, пример вызова функции удаления документа

На экран будет выведен запрос на подтверждение удаления.



Документы будут удалены, если пользователь выберет «OK».

Для вывода на экран интерфейса для снятия блокировки документов нужно выбрать необходимые идентификаторы документов и подставить их в вызов javascript-функции.

```
$all_docs = Docs\Database::getDocuments();
$all_docs_array = $all_docs->getList();
$all_ids = array();
foreach($all_docs_array as $doc) {
    $doc_id = $doc->getId();
    $doc_name = $doc->getName();
    echo "<input type='button' value='" . $doc_id . " - " . $doc_name . "'
onclick='trustedCA.remove([" . $doc_id . "])'>";
    $all_ids[] = $doc_id;
}
echo "<input type='button' value='REMOVE ALL' onclick='remove(" .
json_encode($all_ids) . ")'>";
```

PHP, пример вывода интерфейса для удаления документов

2.11. Пример организации бизнес-процесса с привязкой к заказу

В модуле встроен один пример реализации бизнес-процесса — привязка документа к заказу в интернет-магазине CMS 1С-Битрикс и последовательная подпись документа клиентом и продавцом.

Чтобы привязать документ к заказу, пользователь загружает документ, указывая три дополнительных параметра. Первый параметр типа «ORDER» содержит значение, содержащее номер заказа. Второй параметр типа «ROLES» содержит информацию о наличии подписи клиента и продавца. Третий параметр типа «USER» привязывает документ к пользователю, оформившему данный заказ.

```
$props = new Docs\PropertyCollection();
$props->add(new Docs\Property("ORDER", "54"));
$props->add(new Docs\Property("ROLES", "NONE"));
$props->add(new Docs\Property("USER", "42"));
Docs\Utils::createDocument("/docs/document.doc", $props);
```

PHP, пример вызова функции создания документа с привязкой к заказу в магазине CMS 1С-Битрикс

При подписании продавцом значение параметра «ROLES» устанавливается как «SELLER», при подписании клиентом — «CLIENT». Когда оба участника процесса подписали документ, у него устанавливается значение «BOTH».

Для того чтобы отличить подпись клиента от подписи продавца, используется аргумент «extra» функции «sign».

```
sign([545, 546, 547], {"role":"SELLER"})
```

Javascript, пример отправки документов на подпись от лица продавца

```
$all_docs = Docs\Database::getDocuments();
$all_docs_array = $all_docs->getList();
$all_ids = array();
foreach ($all_docs_array as $doc) {
    $doc_id = $doc->getId();
    $doc_name = $doc->getName();
    echo "<input type='button' value='" . $doc_id . " - " . $doc_name . "'
onclick='sign([" . $doc_id . "], {\\"role\\":\\"CLIENT\\"})'>";
    $all_ids[] = $doc_id;
}
echo "<input type='button' value = 'SIGN ALL AS CLIENT'
onclick='trustedCA.sign(" . json_encode($all_ids) . ",
{\\"role\\":\\"CLIENT\\"})'>"
```

PHP, пример вывода интерфейса отправки документов на подпись от лица клиента

3. Константы модуля КриптоАРМ Документы

3.1. Таблицы в базе данных

Константа	Значение	Описание
DB_TABLE_DOCUMENTS	tr_ca_docs	Название таблицы в базе данных, хранящей записи о загруженных документах.
DB_TABLE_PROPERTY	tr_ca_docs_property	Название таблицы в базе данных, хранящей записи о дополнительных свойствах загруженных документов.

3.2. Типы документов

Константа	Значение	Описание
DOC_TYPE_FILE	0	Неподписанный документ. Значение по умолчанию.
DOC_TYPE_SIGNED_FILE	1	Подписанный документ.

3.3. Статусы документов

Константа	Значение	Описание
DOC_STATUS_NONE	0	Отсутствие статуса. Значение по умолчанию.
DOC_STATUS_BLOCKED	1	Документ заблокирован в связи с отправкой на подпись.
DOC_STATUS_CANCELED	2	Подпись документа была отменена.
DOC_STATUS_ERROR	3	Ошибка во время подписи документа.

4. Работа с API модуля КриптоАРМ Документы

В модуле начиная с версии 2.1.1 есть возможность работать с API напрямую. Все контроллеры находятся по адресу: “https://адрес_сайта/bitrix/components/trusted/api/”. Авторизация на контроллерах происходит посредством логина и пароля с системы CMS Битрикс или посредством токена с сервиса id.trusted.plus. Подробное описание находится на [swagger](#). Все параметры можно передавать через POST(желательнее всего) или же GET запрос

Существуют несколько контроллеров, которые позволяют:

- 1) Получить список ID документов к которым пользователь имеет доступ (п.п.4.2)
- 2) Получить информацию о документах (п.п.4.3)
- 3) Получить ссылки на скачку документов (п.п.4.4)
- 4) Удалить документы (п.п.4.5)
- 5) Отправить документы на почту (п.п.4.6)
- 6) Поделиться документом (п.п.4.7)
- 7) Отправить запрос на подпись (п.п.4.8)
- 8) Добавить новый документ в систему (п.п.4.9)
- 9) Отказаться от доступа к документу (п.п.4.10)
- 10) Разблокировать документ отправленный на подпись (п.п.4.11)
- 11) Отправить на подпись документ (п.п.4.12)
- 12) Загрузить подписанный файл в систему (п.п.4.13)

4.1. Авторизация на контроллерах

Авторизацию происходит при помощи параметров передаваемых в POST запросе. В случае, если данные подходят, то произойдет авторизация под данным пользователем на одну операцию. Оба способа являются равноправными. Далее будут приведены примеры с token авторизацией. Авторизация является важным звеном каждого контроллера. Существуют два способа авторизации:

- 1) Авторизация при помощи логина и пароля с CMS Битрикс.

Константа	Значение	Описание
grandType	password	Тип авторизации. В данном примере авторизация идет по логину и паролю.
login	admin	Логин пользователя в системе CMS Битрикс
password	qwerty	Пароль пользователя в системе CMS Битрикс

- 2) Авторизация при помощи токена с сервиса <https://id.trusted.plus>

Константа	Значение	Описание
-----------	----------	----------

grandType	token	Тип авторизации. В данном примере авторизация идет по токenu
token	2f1df24b-3b28-48d4-92c7-6a6150e4a11d	Токен пользователя полученного с сервиса id.trusted.plus

4.2. Получение списка ID документов

Контроллер позволяет получить список ID документов, доступных данному пользователю. Описание авторизации на контроллере описано в (п.п.4.1). Контроллер находится по адресу: "https://адрес_сайта/bitrix/components/trusted/api/getList.php". Ответом данного контроллера будет JSON содержащий: код ответа, сообщение

Константа	Значение	Описание
grandType	token	Тип авторизации. В данном примере авторизация идет по токenu
token	2f1df24b-3b28-48d4-92c7-6a6150e4a11d	Токен пользователя полученного с сервиса id.trusted.plus

Ответ вида:

```
{
  "code": 200,
  "message": "ok",
  "data": [
    "2,3,4"
  ]
}
```

4.3. Получение информации о документах

Получение информации о документах по их ID, в случае, если пользователь имеет доступ к документу. Описание авторизации на контроллере описано в (п.п.4.1). Контроллер находится по адресу: "https://адрес_сайта/bitrix/components/trusted/api/getDocs.php".

Константа	Значение	Описание
grandType	token	Тип авторизации. В данном примере авторизация идет по токenu
token	2f1df24b-3b28-48d4-92c7-6a6150e4a11d	Токен пользователя полученного с сервиса id.trusted.plus

ids [2,3,7] Массив ID документов к которым нужно получить подробную информацию

Ответ вида:

```
{
  "code": 200,
  "message": "ok",
  "data": [
    {
      "id": 1,
      "code": 900,
      "message": "ok",
      "name": "Какой-то документ.doc",
      "type": 912,
      "status": 921,
      "hash": "60d5d24dad97596430e96c538273cd0",
      "access": 932,
      "date": "2019-12-12 11:58:30",
      "require": 940,
      "owner": "Женя Спицин"
    }
  ]
}
```

4.4. Получение ссылки на скачивание документов

Получение ссылки на скачивание документов по их ID в случае, если пользователь имеет доступ к документу. Описание авторизации на контроллере описано в (п.п.4.1). Контроллер находится по адресу: "https://адрес_сайта/bitrix/components/trusted/api/downloadDocs.php".

Константа	Значение	Описание
grandType	token	Тип авторизации. В данном примере авторизация идет по токenu
token	2f1df24b-3b28-48d4-92c7-6a6150e4a11d	Токен пользователя полученного с сервиса id.trusted.plus
ids	[2,3,7]	Массив ID документов к которым нужно получить ссылки на скачивание

Ответ вида:

```
{
  "code": 200,
  "message": "ok",
```

```

{
  "data": [
    {
      "id": 1,
      "code": 900,
      "message": "ok",
      "url": "https://localhost/docs/5df8d907c91a0/61Ywypnuw9Y.jpg"
    }
  ]
}

```

4.5. Удаление документов

Удаление документов по их ID в случае, если пользователь имеет доступ к документу. Описание авторизации на контроллере описано в (п.п.4.1). Контроллер находится по адресу: "https://адрес_сайта/bitrix/components/trusted/api/removeDocs.php".

Константа	Значение	Описание
grandType	token	Тип авторизации. В данном примере авторизация идет по токenu
token	2f1df24b-3b28-48d4-92c7-6a6150e4a11d	Токен пользователя полученного с сервиса id.trusted.plus
ids	[2,3,7]	Массив ID документов которые нужно удалить

Ответ вида:

```

{
  "code": 200,
  "message": "ok",
  "data": [
    {
      "id": 5,
      "code": 901,
      "message": "Нет прав"
    }
  ]
}

```

4.6. Отправление документа на почту

Отправление документов по их ID на почту. Почтовый шаблон настраивается на сайте. Отправка документов будет производиться от лица сайта, но с пометкой от какого пользователя, в случае, если стандартный почтовый шаблон не был изменен. Описание

авторизации на контроллере описано в (п.п.4.1). Контроллер находится по адресу: "https://адрес_сайта/bitrix/components/trusted/api/sendDocs.php".

Константа	Значение	Описание
grandType	token	Тип авторизации. В данном примере авторизация идет по токену
token	2f1df24b-3b28-48d4-92c7-6a6150e4a11d	Токен пользователя полученного с сервиса id.trusted.plus
ids	[2,3,7]	Массив ID документов которые нужно удалить
email	admin@admin.ru	Почтовый адрес на который предстоит отправить данные документы

Ответ вида:

```
{
  "code": 200,
  "message": "ok",
  "data": [
    1
  ]
}
```

4.7. Предоставление доступа к документу

Предоставление доступа к документам по их ID пользователю с данным email адресом. Email адрес должен быть зарегистрированного пользователя в CMS Битрикс вашего сайта, в противном случае вы получите соответствующий ответ. При получении доступа к файлу, пользователь получит email письмо с информации, что получен доступ к файлу. Описание авторизации на контроллере описано в (п.п.4.1). Контроллер находится по адресу: "https://адрес_сайта/bitrix/components/trusted/api/shareDocs.php".

Константа	Значение	Описание
grandType	token	Тип авторизации. В данном примере авторизация идет по токену
token	2f1df24b-3b28-48d4-92c7-6a6150e4a11d	Токен пользователя полученного с сервиса id.trusted.plus
ids	[2,3,7]	Массив ID документов которые нужно удалить

email	admin@admin.ru	Почтовый адрес на который предстоит отправить данные документы
-------	----------------	--

Ответ вида:

```
{
  "code": 200,
  "message": "ok",
  "data": [
    {
      "id": 1,
      "success": true
    }
  ]
}
```

4.8. Отправить документ на подпись

Отправление запроса на подпись документов по их ID пользователю с указанным email. Email адрес должен быть зарегистрированного пользователя в CMS Битрикс вашего сайта. В противном случае вы получите соответствующий ответ. При получения запроса на подпись, пользователь получит email письмо с информацией, что вы ждете подпись на этом документе, а также поменяется статус документа у пользователя в компоненте “Документы пользователя” на “Требуется подпись”. Описание авторизации на контроллере описано в (п.п.4.1). Контроллер находится по адресу: “https://адрес_сайта/bitrix/components/trusted/api/requireToSign.php”.

Константа	Значение	Описание
grandType	token	Тип авторизации. В данном примере авторизация идет по токenu
token	2f1df24b-3b28-48d4-92c7-6a6150e4a11d	Токен пользователя полученного с сервиса id.trusted.plus
ids	[2,3,7]	Массив ID документов которые нужно удалить
email	admin@admin.ru	Почтовый адрес пользователя которому надо отправить запрос на подпись

Ответ вида:

```
{
  "code": 200,
  "message": "ok",
  "data": [
```

```
{
  "id": 1,
  "success": true
}
]
```

4.9. Загрузить новый документ

Данный контроллер позволяет загрузить новый документ. В случае успешной загрузки контроллер вернет вам ID документа. При загрузке, документ привязывается к пользователю которым вы авторизовываетесь. Описание авторизации на контроллере описано в (п.п.4.1). Контроллер находится по адресу: "https://адрес_сайта/bitrix/components/trusted/api/addDoc.php".

Константа	Значение	Описание
grandType	token	Тип авторизации. В данном примере авторизация идет по токену
token	2f1df24b-3b28-48d4-92c7-6a6150e4a11d	Токен пользователя полученного с сервиса id.trusted.plus
file	binary file	Сам файл для загрузки.

Примечание: На сервере существуют ограничения по максимально допустимому размеру файла. В случае, если файл слишком большой - он не загрузится.

Ответ вида:

```
{
  "code": 200,
  "message": "ok",
  "data": {
    "id": 23,
    "name": "Какой-то документ.docx"
  }
}
```

4.10. Отказаться от доступа к документу

Данный контроллер позволяет отказаться от документа, в случае, если ранее к этому документу получили доступ от другого лица. Если вы владелец или у вас нет доступа, то выводятся соответствующие ответы. Описание авторизации на контроллере описано в

(п.п.4.1). Контроллер находится по адресу:
 “https://адрес_сайта/bitrix/components/trusted/api/unShareDocs.php”.

Константа	Значение	Описание
grandType	token	Тип авторизации. В данном примере авторизация идет по токenu
token	2f1df24b-3b28-48d4-92c7-6a6150e4a11d	Токен пользователя полученного с сервиса id.trusted.plus
ids	[2,3,7]	Массив ID документов от которых надо отказаться

Ответ вида:

```
{
  "code": 200,
  "message": "ok",
  "data": [
    {
      "code": 901,
      "message": "Нет прав"
    }
  ]
}
```

4.11. Разблокировать документы

Разблокировка документа, если он был заблокирован(отправлен на подпись) и нужно получить доступ для подписывания. Описание авторизации на контроллере описано в (п.п.4.1). Контроллер находится по адресу:
 “https://адрес_сайта/bitrix/components/trusted/api/unBlockDocs.php”.

Константа	Значение	Описание
grandType	token	Тип авторизации. В данном примере авторизация идет по токenu
token	2f1df24b-3b28-48d4-92c7-6a6150e4a11d	Токен пользователя полученного с сервиса id.trusted.plus
ids	[2,3,7]	Массив ID документов, которые надо разблокировать

Ответ вида:

```
{
  "code": 200,
  "message": "ok",
  "data": [
    {
      "code": 901,
      "message": "Нет прав"
    }
  ]
}
```

4.12. Подписать документы

Позволяет отправить документ на подпись, при этом возвращается информация для подписи документа, а также документ блокируется с целью невозможности каких либо изменений в период подписи. Документ можно разблокировать с помощью интерфейса на сайте, а также через функционал “Разблокировать документы”(п.п.4.11). Описание авторизации на контроллере описано в (п.п.4.1). Контроллер находится по адресу: “https://адрес_сайта/bitrix/components/trusted/api/signDocs.php”.

Константа	Значение	Описание
grandType	token	Тип авторизации. В данном примере авторизация идет по токену
token	2f1df24b-3b28-48d4-92c7-6a6150e4a11d	Токен пользователя полученного с сервиса id.trusted.plus
ids	[2,3,7]	Массив ID документов, которые надо подписать

Ответ вида:

```
{
  "code": 200,
  "message": "ok",
  "data": [
    {
      "id": 1,
      "code": 900,
      "message": "Все хорошо",
      "name": "КакойтоДокумент.docs",
      "hash": "60d5d24dad97596430e96c538273cd0",
      "token": "239c5b9f-dc04-42ac-89f7-b716d3965610",
      "license": "string",
      "url": "https://localhost/docs/5df8d907c91a0/КакойтоДокумент.docs"
    }
  ]
}
```

4.13. Загрузить подписанный документ

Данный контроллер позволяет загрузить подписанный документ, который был получен при помощи контроллера “Подписать документы”(п.п.4.12). Токен подписи и ID документа, который подписывают, должны обязательно совпадать, иначе подпись не произойдет и вернется соответствующий ответ. Описание авторизации на контроллере описано в (п.п.4.1). Контроллер находится по адресу: “https://адрес_сайта/bitrix/components/trusted/api/uploadDoc.php”.

Константа	Значение	Описание
grandType	token	Тип авторизации. В данном примере авторизация идет по токену
token	2f1df24b-3b28-48d4-92c7-6a6150e4a11d	Токен пользователя полученного с сервиса id.trusted.plus
id	2	ID документа, которые надо подписать
file	binary file	Подписанный файл
signToken	2f1df24b-3b28-48d4-92c7-6a6150e4a11d	Токен операции подписи
signers	[{"serialNumber": "", "digestAlgorithm": "", "issuerName": "", "notAfter": "", "notBefore": "", "organizationName": "", "signingTime": "", "subjectFriendlyName": "", "subjectName": "", "x509": ""}]	Массив информации о цепочки подписи

Ответ вида:

```
{
  "code": 200,
  "message": "ok",
  "data": [
    {
      "id": 1,
      "code": 900,
      "message": "Все хорошо"
    }
  ]
}
```



```
]
}
```

4.14. Описание кодов ответов

Код	Сообщение	Описание
200	ok	Все хорошо
800	trusted.id is not installed	Модуль "trusted.id" не установлен. Авторизация с помощью токена не возможна
801	token is not find	Параметр token не был передан
802	user did not give permission	Пользователь не дал разрешение на передачу персональных данных сайту
803	user is not find	Данный пользователь не найден в CMS Битрикс
804	something wrong	Что-то пошло не так с модулем "trusted.id"
900	ok	Все хорошо
901	have not permission	Не хватает полномочий для получения доступа. Документ не принадлежит вам
902	document does not exist	Документ не существует на сервере
903	document already unblocked	Документ уже разблокирован
904	some documents not sent	Некоторые документы не были отправлены по почте
905	documents not send	Все документы не отправлены
906	user is not find	Пользователь с данной почтой(в поле email) не найден
907	email is not find	Параметр email не был передан
908	ids is not find	Параметр ids не был передан

909	something wrong	Что-то пошло не так с модулем "trusted.cryptoarmdocscrp"
910	910	Документ готов для взаимодействий
911	911	Документ заблокирован
912	912	Документ отклонен(внутренняя ошибка связанная с типами)
913	913	Ошибка документа(внутренняя ошибка связанная с типами)
920	920	Документ не имеет подписи
921	921	Документ имеет подпись
930	930	Вы автор документа
931	931	С вами поделились документом и вы имеете право на: чтение документа, скачивание, проверка подписи, подпись документа, отправка по почте, отказ от доступа
932	932	С вами поделились документом и вы имеете право на: чтение документа, скачивание, проверка подписи, отправка по почте, отказ от доступа
940	940	Не запроса на подпись
941	941	Есть запрос на подпись
950	email is not valid	Почта не соответствует формату a@a.a
951	incorrect signers parameter	Параметр signers не был передан
952	incorrect sign token parameter	Параметр signToken не был передан
953	document does not exist	В параметре file отсутствует файл
954	document already has child	Документ не последний в цепочке. Подпись невозможна
955	wrong token	Параметр signToken не соответствует выписанному токеноу подписи

