



# Руководство программиста. API

## Оглавление

Описание API КриптоАРМ .....	4
1. Описание запросов и ответов .....	6
2. Команда signAndEncrypt. Запросы на подпись и шифрование документов, выполнение обратных операций .....	7
2.1. Формат ссылки .....	8
2.2. Описание запросов и ответов .....	8
2.2.1. Получение параметров операции .....	8
2.2.2. Отправка результата прямых операций .....	10
2.2.3. Отправка результата обратных операций .....	11
2.2.4. Отправка результата проверки подписи .....	12
2.3. Типы данных .....	13
2.3.1. Интерфейс ISignAndEncryptParameters .....	13
2.3.2. Тип ISignAndEncryptOperationDirect .....	13
2.3.3. Тип ISignAndEncryptOperationReverse .....	13
2.3.4. Тип ISignAndEncryptOperationVerify .....	13
2.3.5. Интерфейс ISignAndEncryptOperationProps .....	13
2.3.6. Интерфейс IFile .....	14
2.3.7. Интерфейс IExtra .....	14
2.3.8. Интерфейс IDirectResults .....	16
2.3.9. Интерфейс IDirectResultOut .....	16
2.3.10. Интерфейс IReverseResults .....	16
2.3.11. Интерфейс IReverseResultOut .....	17
2.3.12. Интерфейс IVerifySignResults .....	17
2.3.13. Интерфейс IVerifySignResult .....	17
2.3.14. Интерфейс ISignerStatus .....	17
2.3.15. Интерфейс ILocalResultParams .....	18
2.3.16. Интерфейс ISignStampAppearance .....	18
2.3.17. Интерфейс IMockupSettings .....	18
2.3.18. Интерфейс IRequisitesSettings .....	19
2.3.19. Интерфейс IPdfCertRequisite .....	19
2.3.20. Интерфейс IPdfMarkedArea .....	20
2.4. Интерфейс КриптоАРМ при подписи документов .....	20
3. Команда certificates. Запросы на экспорт или импорт сертификатов и просмотр информации о сертификате .....	21
3.1. Формат ссылки .....	22
3.2. Описание запросов и ответов .....	22
3.2.1. Получение параметров операции .....	22
3.2.2. Отправка сертификата .....	24

3.2.3. Отправка списка сертификатов.....	25
3.2.4. Отправка сведений о сертификате .....	26
3.3. Типы данных.....	27
3.3.1. Интерфейс ICertificatesParameters .....	27
3.3.2. Интерфейс ICertificatesOperationProps .....	27
3.3.3. Интерфейс ICertificateBase64Params .....	27
3.3.4. Интерфейс ICertificateInfo .....	27
3.3.5. Интерфейс ICertificateIdentityInfo .....	28
3.4. Интерфейс КриптоАРМ при выборе и отправке сертификатов .....	28
4. Команда certrequests. Генерация запросов на сертификат .....	29
4.1. Формат ссылки.....	30
4.2. Описание запросов и ответов.....	30
4.2.1. Получение параметров операции.....	30
4.2.2. Отправка запроса на сертификат.....	32
4.3. Типы данных.....	33
4.3.1. Интерфейс ICertrequestsParameters .....	33
4.3.2. Тип CertrequestsOperation .....	33
4.3.3. Интерфейс ICertrequestsOperationGenerateProps .....	33
4.3.4. Интерфейс IJSONTemplate .....	34
4.3.5. Интерфейс IRDN .....	34
4.3.6. Интерфейс IRequestExtension.....	34
4.3.7. Интерфейс IKeyUsage.....	35
4.3.8. Интерфейс IExtendedKeyUsage.....	35
4.3.9. Интерфейс ICertificateTemplate .....	35
4.3.10. Интерфейс ICertificaterequestBase64Params .....	35
5. Команда diagnostics. Запросы на диагностику рабочего места .....	36
5.1. Формат ссылки.....	37
5.2. Описание запросов и ответов.....	37
5.2.1. Получение параметров операции.....	37
5.2.2. Отправка сведений о рабочем месте.....	38
5.3. Типы данных.....	39
5.3.1. Интерфейс IDiagnosticsParameters .....	39
5.3.2. Тип IDiagnosticOperation .....	40
5.3.3. Интерфейс IDiagnosticsOperationProps.....	40
5.3.4. Интерфейс IDiagnosticsInformation .....	40
5.3.5. Интерфейс ISystemInformation.....	41
5.3.6. Интерфейс IVersions.....	41
5.3.7. Интерфейс IProviders .....	41
5.3.8. Интерфейс ILicenses .....	41

5.3.9. Интерфейс ILicenseInfo .....	42
5.3.10. LicenseType Enum .....	42
6. Команда startView. Открытие окна приложения .....	43
6.1. Формат ссылки.....	44
6.2. Описание запросов и ответов.....	44
6.2.1. Получение параметров операции.....	44
6.3. Типы данных.....	45
6.3.1. Интерфейс IStartViewParameters .....	45
6.3.2. Интерфейс IStartViewOperationProps .....	46
7. Команда sendMail. Действия с электронными письмами .....	47
7.1. Формат ссылки.....	48
7.2. Описание запросов и ответов.....	48
7.2.1. Получение параметров операции.....	48
7.3. Типы данных.....	50
7.3.1. Интерфейс IMailParameters .....	50
7.3.2. Интерфейс IMailOperationProps .....	50
7.3.3. Интерфейс IMailProps .....	51

## Описание API КристоАРМ

КристоАРМ поддерживает API с двумя возможными точками вызова: локальное приложение на рабочем месте и web-приложение (сервис). При вызове операции через API КристоАРМ будет запущен и отобразит необходимый раздел с переданными параметрами. Команду можно ввести через адресную строку браузера (вы можете размещать их так же, как ссылки на веб-страницы) или в терминале. Для взаимодействия используется зарегистрированный протокол **cryptoarm://**

В текущей редакции доступны команды:

- signAndEncrypt – выполнение криптографических операций над документами (подпись, шифрование, проверка подписи, расшифрование)
- certificates – экспорт или импорт сертификатов, просмотр свойств сертификата
- certrequests – генерация запросов на сертификат
- diagnostics – диагностика рабочего места
- startView – открыть окно или вкладку
- senMail – действия с электронными письмами (отправка)

Общий сценарий выполнения команд (для взаимодействия с web-приложениями):

1. Пользователь заходит на портал (web-приложение).
2. Выбирает объекты (например список документов) и действие (например подпись).
3. Портал генерирует и отображает (или сразу переходит) ссылку с протоколом **cryptoarm://**
4. Если КристоАРМ не запущен, то запускается. Затем обращается к portalу за JSON с набором параметров, нужных для выполнения конкретной операции. JSON генерируется на сервере, где располагается web-приложение.
5. Полученный JSON обрабатывается и в зависимости от команды выполняются нужные дополнительные запросы к web-приложению.
6. Пользователь подтверждает саму запрошенную операцию (остальной функционал приложения блокируется).
7. Результаты отправляются на сервер.

Общий сценарий выполнения команд (для взаимодействия с локальным приложением):

1. Пользователь открывает стороннее приложение, в котором есть поддержка КристоАРМ API.
2. Выбирает объекты (например список документов) и действие (например подпись).
3. Приложение генерирует объект (JSON) и передает на исполнение параметром в КристоАРМ.
4. Если КристоАРМ не запущен, то запускается. Затем обращается к portalу за JSON с набором параметров, нужных для выполнения конкретной операции. JSON генерируется на сервере, где располагается web-приложение.

5. Полученный JSON обрабатывается.
6. Пользователь подтверждает саму запрошенную операцию (остальной функционал приложения блокируется).
7. Результаты сохраняются в локальную папку, переданную в параметрах операции.

Общий формат ссылки для web КриптоАРМ API:

**cryptoarm://<command>/<URL>/?id=<id>**

Здесь:

- **cryptoarm://** - зарегистрированный протокол
- **<command>** - выполняемая команда
- **<URL>** - ссылка на получение JSON с параметрами, нужными для выполнения команды
- **?id=<id>** - обязательный параметр. Идентификатор транзакции.

Общий формат команды для локального варианта КриптоАРМ API:

Вариант 1: **cryptoarm://<command>/<URI>**

Здесь:

- **cryptoarm://** - зарегистрированный протокол
- **<command>** - выполняемая команда
- **<URI>** - путь к файлу JSON с параметрами, нужными для выполнения команды

Вариант 2: **cryptoarm://<command>/<JSON>**

Здесь:

- **cryptoarm://** - зарегистрированный протокол
- **<command>** - выполняемая команда
- **<JSON>** - JSON с параметрами, нужными для выполнения команды

## 1. Описание запросов и ответов

Все запросы между КристоАРМ и сервером ДОЛЖНЫ соответствовать спецификации протокола JSON-RPC 2.0. В качестве транспорта используется HTTP. ДОЛЖЕН использоваться TLS, незащищенные соединения КристоАРМ отклоняет.

При включенной настройке «Использовать только набор алгоритмов ГОСТ для подключений по API» КристоАРМ будет принимать только ГОСТ TLS, остальные шифросюиты будут отклоняться.

### POST запрос

КристоАРМ выполняет HTTP POST запросы, которые содержат заголовки:

- Content-Type: ДОЛЖЕН быть application/json.
- Content-Length: ДОЛЖЕН содержать правильную длину в соответствии с HTTP-спецификацией.
- Accept: ДОЛЖЕН быть application/json.

### GET запрос

Не используются.

### Ответ

HTTP ответ сервера ДОЛЖЕН содержать заголовки:

- Content-Type: ДОЛЖЕН быть application/json.
- Content-Length: ДОЛЖЕН содержать правильную длину в соответствии с HTTP-спецификацией.

### Объект Error

В случае ошибки сервер ДОЛЖЕН отправить ответ следующей структуры:

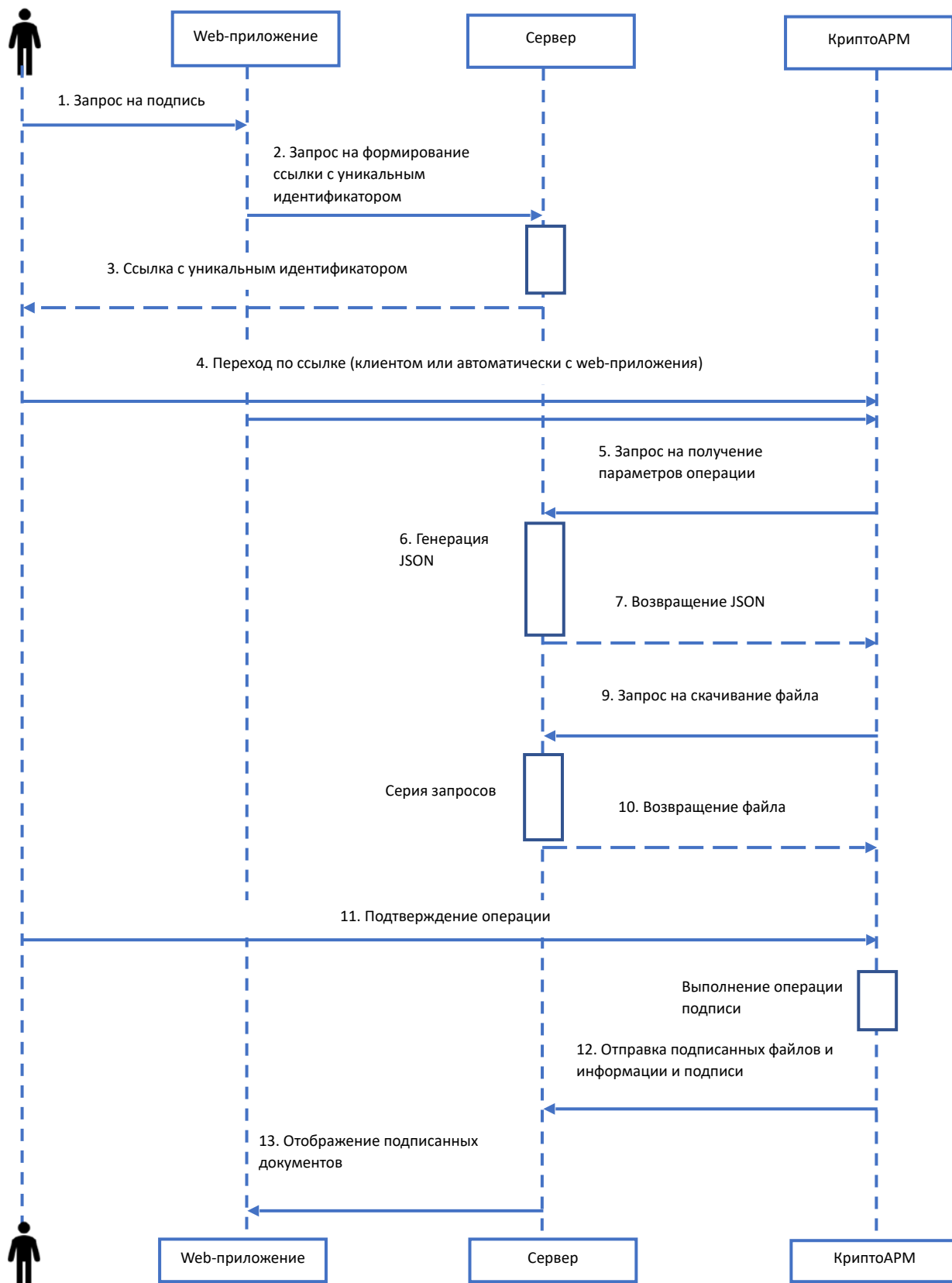
Ключ	Тип	Описание
code	number	Код ошибки
message	string	Короткое описание ошибки
data	string   Object	Необязательное поле. Может содержать дополнительные сведения об ошибке

### HTTP-коды

Код	Ошибка	Описание
200	OK	И для ответов, и для ошибок
204	No Response	Для пустых запросов (нотификация)
405	Method Not Allowed	Метод не доступен
415	Unsupported Media Type	Если Content-Type не application/json

## 2. Команда signAndEncrypt. Запросы на подпись и шифрование документов, выполнение обратных операций

Команда **signAndEncrypt** (подпись и шифрование) используется для запроса на подпись документа или пакета документов. Может использоваться в качестве аутентификатора. Выполнение операции требует действующей лицензии на КриптоАРМ ГОСТ. Схема:





## 2.1. Формат ссылки

Для выполнения команды `signAndEncrypt` должна быть сформирована ссылка вида:

**cryptoarm://signAndEncrypt/<URL>/?id=<id>**

Здесь:

- **cryptoarm://** - зарегистрированный протокол
- **signAndEncrypt** - выполняемая команда
- **<URL>** - ссылка, на которую КристоАРМ будет слать запросы
- **id** – уникальный идентификатор транзакции

Пример:

cryptoarm://signAndEncrypt/https://example.com/json?id=2c48eb32-a0a8-405c-ade9-eed130605cba

## 2.2. Описание запросов и ответов

Все запросы между КристоАРМ и сервером ДОЛЖНЫ соответствовать спецификации протокола JSON-RPC 2.0. В качестве транспорта используется HTTP. Общее описание указано в разделе [1. Описание запросов и ответов](#).

### 2.2.1. Получение параметров операции

После получения команды **signAndEncrypt** КристоАРМ отправляет запрос на **<URL>** для получения параметров операции.

**Формат запроса:**

Ключ	Значение	Описание
jsonrpc	«2.0»	Версия JSON-RPC протокола. Всегда «2.0»
method	«signAndEncrypt.parameters»	Используемый метод. Всегда «signAndEncrypt.parameters»
id	Уникальный идентификатор	Используется идентификатор, который указан в ссылке на операцию ( <a href="#">«Формат ссылки»</a> )
diagnostic	<a href="#">IDiagnosticInformaton</a>	Диагностическая информация о рабочем месте

Пример запроса:

```
Content-Type: application/json
Content-Length: ...
Accept: application/json

{
  "jsonrpc": "2.0",
  "method": "signAndEncrypt.parameters",
```

```

    "id": "2c48eb32-a0a8-405c-ade9-eed130605cba",
    "diagnostic": {
    }
}

```

#### Формат ответа:

Ключ	Значение	Описание
jsonrpc	2.0	Версия JSON-RPC протокола. Всегда «2.0»
result	<a href="#">ISignAndEncryptParameters</a>	Объект со сведениями о параметрах операции подписи
id	Уникальный идентификатор	Используется идентификатор, который указан в ссылке на операцию ( <a href="#">«Формат ссылки»</a> )
appName?	string	Необязательный параметр. Используется только для локального API. Если значение передано, то КриптоАРМ отобразит это название в окне подтверждения операции как название стороннего приложения
apiKey?	string	Необязательный параметр. Используется только для локального API. Если значение передано, то КриптоАРМ отобразит это значение в окне подтверждения операции как проверочный код

#### Пример ответа:

```

HTTP/1.1 200 OK
Connection: close
Content-Length: ...
Content-Type: application/json
Date: Sat, 08 Jul 2020 12:04:08 GMT

{
  "jsonrpc": "2.0",
  "result": {
    "operation": [
      "SIGN",
      "ARCHIVE",
      "ENCRYPT"
    ],
    "props": {
      "headerText": "Подпись документов cryptoarm.ru",
      "license": "",
      "files": [{
        "name": "file1.txt",

```

```

        "url": "http://localhost:8080/public/files/file1.txt",
        "id": 1,
        "urlDetached": ""
    },
    {
        "name": "file2.txt",
        "url": "http://localhost:8080/public/files/file2.txt",
        "id": 2,
        "urlDetached": ""
    },
    {
        "name": "file4.pdf",
        "url": "http://localhost:8080/public/files/file4.pdf",
        "id": 4,
        "urlDetached": ""
    }
],
"extra": {
    "token": "9c7101f7-9c47-4481-b4da-a6a497abde08",
    "signType": "1",
    "signStandart": "1"
},
"uploader": "http://localhost:8080/upload"
}
},
"id": "2c48eb32-a0a8-405c-ade9-eed130605cba"
}

```

### 2.2.2. Отправка результата прямых операций

После того, как пользователь выберет нужные сертификаты КриптоАРМ выполняет операцию. Полученные файлы отправляется POST запросом. Используются нотификации (уведомления), для которых не требуется ответ сервера.

#### Формат запроса:

Ключ	Значение	Описание
jsonrpc	«2.0»	Версия JSON-RPC протокола. Всегда «2.0»
method	«signAndEncrypt.outDirectResults»	Используемый метод. Всегда «signAndEncrypt.outDirectResults»
params	Объект типа <a href="#">IDirectResults</a>	Сведения о результатах прямой операции

#### Пример запроса:

```

Content-Type: application/json
Content-Length: ...
Accept: application/json

```

```
{
  "jsonrpc": "2.0",
  "method": "signAndEncrypt.outDirectResults",
  "params": {
    "id": "2c48eb32-a0a8-405c-ade9-eed130605cba",
    "directResults": [
      {
        "id": 2
        "out": "MIIWgQYJKoZIhvcNAQcCoIIWcjCCFm4CA...cN/aHmA="
      }
    ]
  }
}
```

### 2.2.3. Отправка результата обратных операций

Результаты отправляется POST запросом. Используются нотификации (уведомления), для которых не требуется ответ сервера.

#### Формат запроса:

Ключ	Значение	Описание
jsonrpc	«2.0»	Версия JSON-RPC протокола. Всегда «2.0»
method	«signAndEncrypt.outReverseResults»	Используемый метод. Всегда «signAndEncrypt.outReverseResults»
params	Объект типа <a href="#">IReverseResults</a>	Сведения о результатах обратной операции

#### Пример запроса:

Content-Type: application/json

Content-Length: ...

Accept: application/json

```
{
  "jsonrpc": "2.0",
  "method": "signAndEncrypt.outReverseResults",
  "params": {
    "id": "2c48eb32-a0a8-405c-ade9-eed130605cba",
    "reverseResults": [
      {
        "out": "MIIWgQYJKoZIhvcNAQcCoIIWcjCCFm4CA...cN/aHmA="
      }
    ]
  }
}
```

```
}  
}
```

#### 2.2.4. Отправка результата проверки подписи

Результаты проверки подписи отправляется POST запросом. Используются нотификации (уведомления), для которых не требуется ответ сервера.

##### Формат запроса:

Ключ	Значение	Описание
jsonrpc	«2.0»	Версия JSON-RPC протокола. Всегда «2.0»
method	«signAndEncrypt.verifySignResults»	Используемый метод. Всегда «signAndEncrypt.verifySignResults»
params	Объект типа <a href="#">VerifySignResults</a>	Сведения о проверке подписи

##### Пример запроса:

Content-Type: application/json

Content-Length: ...

Accept: application/json

```
{  
  "jsonrpc": "2.0",  
  "method": "signAndEncrypt.verifySignResults",  
  "params": {  
    "id": "2c48eb32-a0a8-405c-ade9-eed130605cba",  
    "verifyResults": [  
      {  
        "id": 2,  
        "status": false,  
        "signers": [  
          {  
            "hash": "MIIFDCCBMGgAwIBAgIQTm1HiybyfWV",  
            "issuerFriendlyName": "Минкомсвязь России",  
            "issuerName": "Минкомсвязь России",  
            "subjectFriendlyName": "Минкомсвязь России",  
            "subjectName": "Минкомсвязь России",  
            "status": true  
          }  
        ]  
      }  
    ]  
  }  
}
```

## 2.3. Типы данных

В данном разделе представлены типы данных, специфичные для команды **signAndEncrypt**.

### 2.3.1. Интерфейс ISignAndEncryptParameters

Объекты данного типа описывают вид операции и её параметры

Свойство	Тип	Описание
operation	string[]	Тип операции. Доступные значения типов (комбинировать различные типы нельзя): ISignAndEncryptOperationDirect, ISignAndEncryptOperationReverse, ISignAndEncryptOperationVerify
props	<a href="#">ISignAndEncryptOperationProps</a>	Параметры операции

### 2.3.2. Тип ISignAndEncryptOperationDirect

Возможные прямые операции.

Значение	Описание
SIGN	Подпись
ARCHIVE	Архивирование
ENCRYPT	Шифрование

### 2.3.3. Тип ISignAndEncryptOperationReverse

Возможные обратные операции.

Значение	Описание
UNSIGN	Снятие подписи
DECRYPT	Расшифрование
UNZIP	Разархивирование

### 2.3.4. Тип ISignAndEncryptOperationVerify

Проверка подписи.

Значение	Описание
VERIFYSIGN	Проверка подписи

### 2.3.5. Интерфейс ISignAndEncryptOperationProps

Интерфейс ISignAndEncryptOperationProps описывает параметры операции.

Свойство	Тип	Описание
headerText?	string	Необязательный параметр. Используется для отображения в заголовке окна. Максимальная длина: 40 символов
descriptionText?	string	Необязательный параметр. Используется для отображения в сведениях об операции. Максимальная длина: 120 символов
license?	string	Необязательное свойство. Содержит временную лицензию, которая будет использоваться для выполнения операции в КриптоАРМ
uploader	string	Ссылка, на которую будут отправлены результаты операции
files	Массив типа <a href="#">IFile[]</a>	Массив файлов на подпись. Если не передан, то должен быть передан параметр «archive»
archive	<a href="#">IFile</a>	Архив файлов на подпись. Распаковываются после получения. Параметр может использоваться вместо «files». Перед отправкой результатов операции на сервер, они будут упакованы в архив.
extra	Объект типа <a href="#">IExtra</a>	Настройки операции
localResultParams	Объект типа <a href="#">ILocalResultParams</a>	Необязательное свойство. Используется только для локального API. Указывает пути для сохранения результатов

### 2.3.6. Интерфейс IFile

Интерфейс IFile описывает файлы и ссылки на них.

Свойство	Тип	Описание
name	string	Имя файла (с расширением)
url	string	Ссылка на скачивание файла
id	string	Уникальный идентификатор файла
urlDetached?	string	Необязательный параметр. Используется для откреплённой подписи

### 2.3.7. Интерфейс IExtra

Интерфейс IExtra описывает настройки операции. Если параметр не задан, то пользователю доступен выбор из всех доступных в приложении значений.

Свойство	Тип	Описание
----------	-----	----------

signType?	number	Необязательный параметр. Возможные значения: 0 - присоединенная подпись 1 - отсоединённая подпись
signStandart?	number	Необязательный параметр. Стандарт подписи. Возможные значения: 0 - CMS 1 - CAdES-X Long Type 1 2 - CAdES-T 3 - CAdES-A
signEncoding?	number	Необязательный параметр. Кодировка. Возможные значения: 0 – BASE-64 1 – DER
timestampOnSign?	string	Необязательный параметр. Штамп времени на подпись. Возможные значения: True - будет добавлен штамп времени False – не будет добавлен штамп времени
timestampOnData?	string	Необязательный параметр. Штамп времени на подписываемые данные. Возможные значения: True - будет добавлен штамп времени False – не будет добавлен штамп времени
encryptEncoding?	number	Необязательный параметр. Кодировка. Возможные значения: 0 – BASE-64 1 – DER
encryptAlgorithm?	number	Необязательный параметр. Алгоритм шифрования. Возможные значения: 0 – ГОСТ 28147-89 1 – ГОСТ 34.12-2015 Магма 2 – ГОСТ 34.12-2015 Кузнечик
token?	string	Необязательный параметр. Токен, который будет использоваться при скачивании файлов с сервиса (параметр запроса)
tspURL?	string	Необязательный параметр. Адрес службы штампов времени
ocspURL?	string	Необязательный параметр. Адрес службы актуальных статусов
encryptCertificates?	string[]	Необязательный параметр. Сертификаты шифрования
signCertificate?	Объект типа: { «x509»: string }	Необязательный параметр. Сертификат подписчика или параметры для поиска сертификата подписчика
signatureExtension?	string	Необязательный параметр. Расширение для подписанного файла



separateArchivesForSignResults?	boolean	Необязательный параметр. Создавать пакет (архив) для каждого подписываемого файла
isAddStampToPdf?	boolean	Необязательный параметр. Добавить печатную форму для PDF-файлов
signStampAppearance?	<a href="#">ISignStampAppearance</a>	Необязательный параметр. Параметры штампа PDF
signStamp?	<a href="#">IPdfMarkedArea</a>	Необязательный параметр. Расположение и размер области под штамп

### 2.3.8. Интерфейс IDirectResults

Объекты данного типа описывают результаты прямой операции.

Свойство	Тип	Описание
id	string	Используется идентификатор, который указан в ссылке на операцию ( <a href="#">«Формат ссылки»</a> )
status	string	Статус выполнения операции. В случае успеха «Completed»
directResults	<a href="#">IDirectResultOut[]</a>	Массив результатов прямых операций

### 2.3.9. Интерфейс IDirectResultOut

Объекты данного типа содержат результаты прямой операции для файла.

Свойство	Тип	Описание
id?	string	Необязательный параметр. Идентификатор исходного файла. Если включено архивирование, то данный параметр не используется, т.к. выходной файл один для всех исходных
out	string	Результат операции в BASE-64
outSignStamp?	string	Необязательный параметр. При включенной опции isAddStampToPdf будет содержать результирующий файл со штампом.
signers?	ISignerStatus[]	Необязательный параметр. Сведения о подписчиках
signValid?	boolean	Необязательный параметр. Общий статус подписи. Учитывается статус всех подписей файла

### 2.3.10. Интерфейс IReverseResults

Объекты данного типа описывают результаты обратной операции.

Свойство	Тип	Описание
id	string	Используется идентификатор, который указан в ссылке на операцию ( <a href="#">«Формат ссылки»</a> )
reverseResults	<a href="#">IReverseResultOut[]</a>	Массив результатов обратных операций

#### 2.3.11. Интерфейс IReverseResultOut

Объекты данного типа описывают результаты обратной операции.

Свойство	Тип	Описание
id	string	Необязательный параметр. Идентификатор исходного файла
out	string	Результат операции в BASE-64

#### 2.3.12. Интерфейс IVerifySignResults

Объекты данного типа описывают результаты проверки подписи.

Свойство	Тип	Описание
id	string	Используется идентификатор, который указан в ссылке на операцию ( <a href="#">«Формат ссылки»</a> )
status	string	Статус выполнения операции. В случае успеха «Completed»
verifySignResults	<a href="#">IVerifySignResult[]</a>	Массив результатов проверки

#### 2.3.13. Интерфейс IVerifySignResult

Объекты данного типа описывают результаты проверки подписи.

Свойство	Тип	Описание
id	string	Идентификатор исходного файла
signValid	boolean	Общий статус проверки подписи
signers	<a href="#">ISignerStatus[]</a>	Информация о подписчиках документа

#### 2.3.14. Интерфейс ISignerStatus

Объекты данного типа описывают сведения о подписчиках.

Свойство	Тип	Описание
signerCertificate	<a href="#">ICertificateInfo</a>	Сведения о сертификате подписчика

isValid	boolean	Статус подписи
signingTime	string	Время подписи

### 2.3.15. Интерфейс ILocalResultParams

Объекты данного типа используется только в параметрах локального API. Содержат пути для сохранения результатов операций.

Свойство	Тип	Описание
savePath?	string	Необязательный параметр. Путь для сохранения результатов операции
saveResultsSeparately?	boolean	Необязательный параметр. Определяет, сохранять ли результаты по каждому файлу в отдельный файл или сохранять в один JSON, один файл.

### 2.3.16. Интерфейс ISignStampAppearance

Интерфейс ISignStampAppearance описывает параметры для штампа PDF.

Свойство	Тип	Описание
mockupSettings	<a href="#">IMockupSettings</a>	Настройки внешнего вида штампа подписи
requisitesSettings	<a href="#">IRequisitesSettings</a>	Настройки содержимого штампа (реквизиты)
pageSelection	string	Определяет, на какой странице или страницах будет отображаться штамп. Возможные значения: «all» – штамп на всех страницах «last» – штамп на последней странице «some» – штамп на некоторых страницах
pageNumbers?	string	Необязательный параметр. Страница или диапазон страниц. Пример: «3» или «1-7»

### 2.3.17. Интерфейс IMockupSettings

Интерфейс IMockupSettings описывает параметры внешнего вида штампа.

Свойство	Тип	Описание
position	string	Положение штампа внутри области. Возможные значения: «fitArea» – вписать «actualSize» – реальный размер «fitMockup» – подогнать область под макет
addBackground	boolean	Определяет, будет ли добавлен фон на область.

addBorders	boolean	Определяет, будет ли добавлена граница на область.
logotype?	string	Необязательный параметр. Путь в файлу с изображением
isHiddenLogo?	boolean	Необязательный параметр. Использовать или нет логотип для штампа

### 2.3.18. Интерфейс IRequisitesSettings

Интерфейс IRequisitesSettings описывает содержимое (реквизиты) штампа подписи.

Свойство	Тип	Описание
position	string	Реквизиты в штампе. Возможные значения: «default» – штамп без дополнительных областей под реквизиты «left» – штамп с дополнительными реквизитами слева «right» – штамп с дополнительными реквизитами справа «leftAndRight» – штамп с дополнительными реквизитами слева и справа
centralDisplayType	string	Внешний вид штампа. Определяет доступность настроек дополнительных реквизитов. Возможные значения: «byGost» – внешний вид по ГОСТ «arbitrary» – произвольно
central	<a href="#">IPdfCertRequisite[]</a>	Массив значений реквизитов, для добавления в центральную часть штампа
left	<a href="#">IPdfCertRequisite[]</a>	Массив значений реквизитов, для добавления в левую часть штампа
right	<a href="#">IPdfCertRequisite[]</a>	Массив значений реквизитов, для добавления в правую часть штампа

### 2.3.19. Интерфейс IPdfCertRequisite

Интерфейс IPdfCertRequisite описывает свойства для реквизита штампа.

Свойство	Тип	Описание
title	string	Заголовок
dataKey?	string	Необязательный параметр. Указывает название поля сертификата, значение которого будет использовано.

isEditable	boolean	Указывает, доступно ли редактирование поля пользователем. Если false, то значение должно быть передано в поле editedValue или взято из сертификата по значению dataKey
editedValue?	string	Необязательный параметр. Данные для поля
editedValueCharLimit?	number	Необязательный параметр. Определяет максимальную длину строки

#### 2.3.20. Интерфейс IPdfMarkedArea

Интерфейс IPdfMarkedArea описывает параметры для расположения области штампа PDF (координаты и размеры).

Свойство	Тип	Описание
height	number	Высота области под штамп
width	number	Ширина области под штамп
horizontalPadding	number	Отступ от левого края документа
verticalPadding	number	Отступ от правого края документа

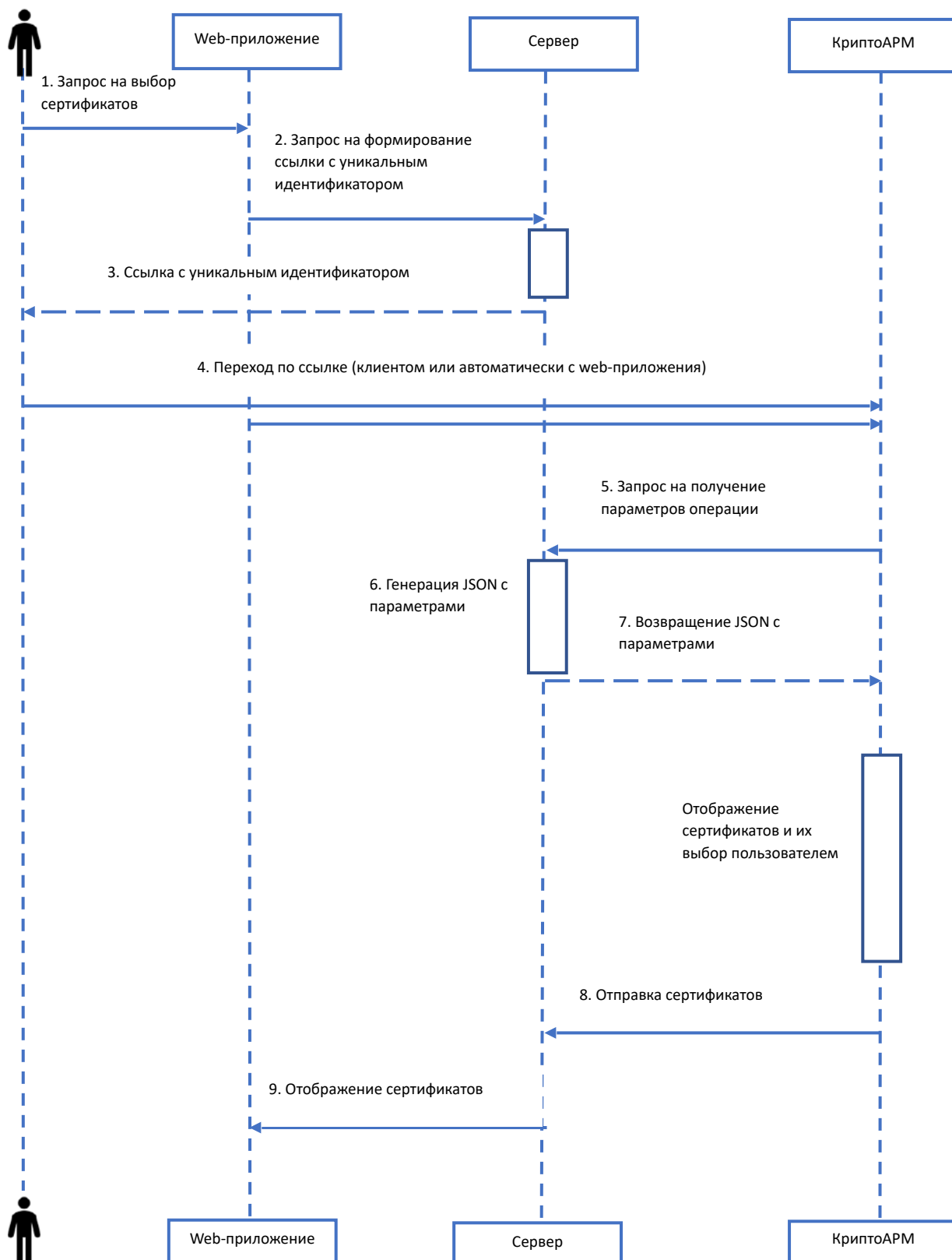
#### 2.4. Интерфейс КриптоАРМ при подписи документов

При выполнении операции подписи по ссылке, не относящийся к процедуре интерфейс блокируется. Пользователю доступны: выбор сертификата, часть настроек подписи. Кнопка «Выполнить» заменяется двумя: «Подпись» и «Отмена». После выполнения команды, приложение будет свернуто в системный трей.

### 3. Команда certificates. Запросы на экспорт или импорт сертификатов и просмотр информации о сертификате

Команда **certificates** используется для: экспорта сертификата или списка сертификатов, а импорта сертификата в локальное хранилище, просмотра информации о сертификате.

Схема взаимодействия (экспорт сертификатов):



### 3.1. Формат ссылки

Для выполнения команды `certificates` должна быть сформирована ссылка вида:

**cryptoarm://certificates/<URL>/?id=<id>**

Здесь:

- **cryptoarm://** - зарегистрированный протокол
- **certificates** - выполняемая команда
- **<URL>** - ссылка, на которую КриптоАРМ будет слать запросы
- **id** – уникальный идентификатор транзакции

Пример:

cryptoarm://certificates/https://example.com/json?id=2c48eb32-a0a8-405c-ade9-eed130605cba

### 3.2. Описание запросов и ответов

Все запросы между КриптоАРМ и сервером ДОЛЖНЫ соответствовать спецификации протокола JSON-RPC 2.0. В качестве транспорта используется HTTP. Общее описание указано в разделе [1. Описание запросов и ответов](#).

#### 3.2.1. Получение параметров операции

После получения команды **certificates** КриптоАРМ отправляет запрос на получение параметров операции.

**Формат запроса:**

Ключ	Значение	Описание
jsonrpc	«2.0»	Версия JSON-RPC протокола. Всегда «2.0»
method	«certificates.parameters»	Используемый метод. Всегда «certificate.parameters»
id	Уникальный идентификатор	Используется идентификатор, который указан в ссылке на операцию ( <a href="#">«Формат ссылки»</a> )
diagnostic	<a href="#">IDiagnosticInformaton</a>	Диагностическая информация о рабочем месте

Пример запроса:

```
Content-Type: application/json
Content-Length: ...
Accept: application/json
```

```
{
  "jsonrpc": "2.0",
  "method": "certificates.parameters",
```

```

    "id": "2c48eb32-a0a8-405c-ade9-eed130605cba",
    "diagnostic": {
    }
}

```

#### Формат ответа:

Ключ	Значение	Описание
jsonrpc	2.0	Версия JSON-RPC протокола. Всегда «2.0»
result	<a href="#">ICertificatesParameters</a>	Объект со сведениями о параметрах операции
id	Уникальный идентификатор	Используется идентификатор, который указан в ссылке на операцию ( <a href="#">«Формат ссылки»</a> )

#### Пример ответа для экспорта сертификата:

```

HTTP/1.1 200 OK
Connection: close
Content-Length: ...
Content-Type: application/json
Date: Sat, 08 Jul 2020 12:04:08 GMT

{
  "jsonrpc": "2.0",
  "result": {
    "operation": "export",
    "props": {
      "store": ["MY"],
      "multy": false
    }
  },
  "id": "2c48eb32-a0a8-405c-ade9-eed130605cba"
}

```

#### Пример ответа для импорта сертификата:

```

HTTP/1.1 200 OK
Connection: close
Content-Length: ...
Content-Type: application/json
Date: Sat, 08 Jul 2020 12:04:08 GMT

{

```



```

"jsonrpc": "2.0",
"result": {
  "operation": "import",
  "props": {
    "store": ["MY"],
    "certificateBase64": "MIIFDCCBMGgAwIBAgIQT...4VVkDWbX/n4="
  }
},
"id": "2c48eb32-a0a8-405c-ade9-eed130605cba"
}

```

Пример ответа для просмотра информации о сертификате:

```

HTTP/1.1 200 OK
Connection: close
Content-Length: ...
Content-Type: application/json
Date: Sat, 08 Jul 2020 12:04:08 GMT

{
  "jsonrpc": "2.0",
  "result": {
    "operation": "information",
    "props": {
      "certificateBase64": "MIIFDCCBMGgAwIBAgIQT...4VVkDWbX/n4="
    }
  },
  "id": "2c48eb32-a0a8-405c-ade9-eed130605cba"
}

```

### 3.2.2. Отправка сертификата

При экспорте сертификатов результат отправляются на сервер. После того, как пользователь выберет нужный сертификат КриптАРМ отправляет запрос, содержащий выбранные элементы (base64 без заголовков). Используются нотификации (уведомления), для которых не требуется ответ сервера.

#### Формат запроса:

Ключ	Значение	Описание
jsonrpc	«2.0»	Версия JSON-RPC протокола. Всегда «2.0»
method	«certificates.base64»	Используемый метод. Всегда «certificates.base64»
params	<a href="#">ICertificateBase64Params</a>	Параметры, содержащие объект сертификата

Пример запроса:

Content-Type: application/json

Content-Length: ...

Accept: application/json

```
{
  "jsonrpc": "2.0",
  "method": "certificates.base64",
  "params": {
    "id": "2c48eb32-a0a8-405c-ade9-eed130605cba",
    "certificateBase64": "MIIFDCCBMGgAwIBAgIQTm1HiybyfWV...4VVkDWbX/n4=",
    "friendlyName": "Минкомсвязь России"
  }
}
```

### 3.2.3. Отправка списка сертификатов

При экспорте сертификатов результат отправляются на сервер. Если параметры ICertRequestParameters содержат поле «multy» со значением «true», то пользователю в КриптоАРМ будет разрешён множественный выбор сертификатов. После того, как пользователь выберет нужные сертификаты и нажмет кнопку Готово, КриптоАРМ отправляет запрос, содержащий выбранные элементы (base64 без заголовков). Используются нотификации (уведомления), для которых не используется ответ сервера.

**Формат запроса:**

Ключ	Значение	Описание
jsonrpc	«2.0»	Версия JSON-RPC протокола. Всегда «2.0»
method	«certificates.base64»	Используемый метод. Всегда «certificates.base64»
params	<a href="#">ICertificateBase64Params[]</a>	Параметры запроса

Пример запроса:

Content-Type: application/json

Content-Length: ...

Accept: application/json

```
{
  "jsonrpc": "2.0",
  "method": "certificates.base64",
  "params": {
    "id": "2c48eb32-a0a8-405c-ade9-eed130605cba",
    "certificates": [{
      "certificateBase64":
"MIIFDCCBMGgAwIBAgIQTm1HiybyfWV...4VVkDWbX/n4=",
      "friendlyName": "Минкомсвязь России"
    }]
  }
}
```

```

    },
    {
      "certificateBase64":
"MIIFDCCBMGgAwIBAgIQTm1HiybyfWV...4VVkDwbX/n4=",
      "friendlyName": "Головной удостоверяющий центр"
    }
  ]
}
}

```

### 3.2.4. Отправка сведений о сертификате

После того, как пользователь выберет нужный сертификат КрипТАРМ отправляет запрос, содержащий выбранные элементы (base64 без заголовков). Используются нотификации (уведомления), для которых не требуется ответ сервера.

#### Формат запроса:

Ключ	Значение	Описание
jsonrpc	«2.0»	Версия JSON-RPC протокола. Всегда «2.0»
method	«certificates.information»	Используемый метод или вид команды. Всегда «certificates.information»
params	Объект типа <a href="#">ICertificateInfo</a>	Сведения о сертификате

#### Пример запроса:

```

Content-Type: application/json
Content-Length: ...
Accept: application/json

{
  "jsonrpc": "2.0",
  "method": "certificates.information",
  "params": {
    "id": "2c48eb32-a0a8-405c-ade9-eed130605cba",
    "hash": "MIIFDCCBMGgAwIBAgIQTm1HiybyfWV",
    "issuerFriendlyName": "Минкомсвязь России",
    "issuerName": "Минкомсвязь России",
    "subjectFriendlyName ": "Минкомсвязь России",
    "subjectName": "Минкомсвязь России",
    "status": true
  }
}

```

### 3.3. Типы данных

В данном разделе представлены типы данных, специфичные для команды **certificates**.

#### 3.3.1. Интерфейс ICertificatesParameters

Объекты данного типа описывают параметры команды запроса на получение сертификат

Свойство	Тип	Описание
operation	string	Тип операции импорт, экспорт, информация. Доступные значения: "import", "export", "information"
props	<a href="#">ICertificatesOperationProps</a>	Параметры операции

#### 3.3.2. Интерфейс ICertificatesOperationProps

Объекты данного типа описывают дополнительные свойства операции.

Свойство	Тип	Описание
headerText?	string	Необязательный параметр. Используется для отображения в заголовке окна. Максимальная длина: 40 символов
descriptionText?	string	Необязательный параметр. Используется для отображения в сведениях об операции. Максимальная длина: 120 символов
store?	string[]	Необязательный параметр. Массив имен хранилищ. Если не задан, то используется значение MY (личные сертификаты)
multy?	boolean	Необязательный параметр. Разрешен ли множественный выбор. По умолчанию false
certificateBase64?	string	Необязательный параметр. Сертификат в формате X.509 закодированный в Base64

#### 3.3.3. Интерфейс ICertificateBase64Params

Объекты данного типа описывают параметры запроса для метода certificates.base64.

Свойство	Тип	Описание
id	string	Идентификатор транзакции
certificateBase64	string	Сертификат в формате X.509 закодированный в Base64
friendlyName	string	Дружественное имя сертификата

#### 3.3.4. Интерфейс ICertificateInfo

Объекты данного типа описывают объекты, содержащие свойства сертификата

Свойство	Тип	Описание
hash	string	SHA1 отпечаток
issuerFriendlyName	string	Дружественное имя издателя (CN)
issuerName	string	Имя издателя
notAfter	string	Дата окончания действия сертификата
notBefore	string	Дата начала действия сертификата
rootCAMinComSvyaz	boolean	Флаг, обозначающий является ли владельцем корневого сертификата цепочки “Минкомсвязь России”
subjectFriendlyName	string	Дружественное имя субъекта (CN)
subjectName	string	Имя субъекта
status	boolean	Статус сертификата. Проверяется вся цепочка
serial	string	Серийный номер сертификата
x509?	string	Необязательный параметр. Сертификат в формате X.509 закодированный в Base64

### 3.3.5. Интерфейс ICertificateIdentityInfo

Объекты данного типа описывают объекты, содержащие параметры сертификата, позволяющие только идентифицировать его.

Свойство	Тип	Описание
id	string	Идентификатор транзакции
hash	string	SHA1 отпечаток
rootCAMinComSvyaz	boolean	Флаг, обозначающий является ли владельцем корневого сертификата цепочки “Минкомсвязь России”
status	boolean	Статус сертификата. Проверяется вся цепочка
pubKeyAlg	string	OID алгоритма открытого ключа сертификата

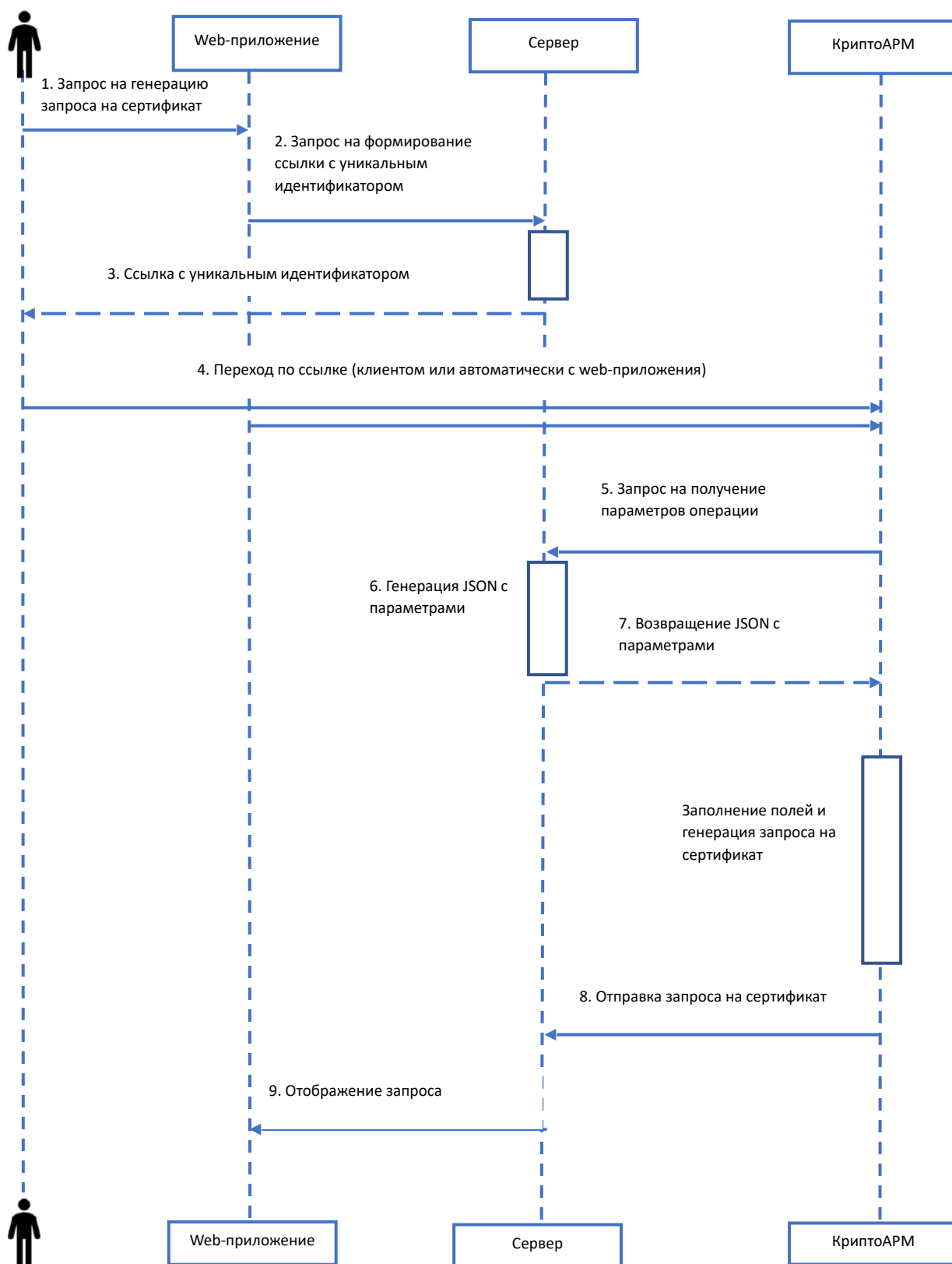
## 3.4. Интерфейс КриптоАРМ при выборе и отправке сертификатов

При выполнении команды запроса на сертификаты, не относящийся к процедуре интерфейс блокируется. Пользователю доступны: выбор сертификатов. Доступны две кнопки: «Готово» и «Отмена». После выполнения команды, приложение будет свернуто в системный трей.

## 4. Команда **certrequests**. Генерация запросов на сертификат

Команда **certrequests** используется для: генерации запроса на сертификат по шаблону или экспорт запросов

Схема взаимодействия (экспорт сертификатов):



## 4.1. Формат ссылки

Для выполнения команды **certrequests** должна быть сформирована ссылка вида:

**cryptoarm://certrequests/<URL>/?id=<id>**

Здесь:

- **cryptoarm://** - зарегистрированный протокол
- **certrequests** - выполняемая команда
- **<URL>** - ссылка, на которую КриптоАРМ будет слать запросы
- **id** – уникальный идентификатор транзакции

Пример:

cryptoarm://certrequests/https://example.com/json?id=2c48eb32-a0a8-405c-ade9-eed130605cba

## 4.2. Описание запросов и ответов

Все запросы между КриптоАРМ и сервером ДОЛЖНЫ соответствовать спецификации протокола JSON-RPC 2.0. В качестве транспорта используется HTTP. Общее описание указано в разделе [1. Описание запросов и ответов](#).

### 4.2.1. Получение параметров операции

После получения команды **certrequests** КриптоАРМ отправляет запрос на получение параметров операции.

**Формат запроса:**

Ключ	Значение	Описание
jsonrpc	«2.0»	Версия JSON-RPC протокола. Всегда «2.0»
method	«certrequests.parameters»	Используемый метод. Всегда «certrequests.parameters»
id	Уникальный идентификатор	Используется идентификатор, который указан в ссылке на операцию ( <a href="#">«Формат ссылки»</a> )
diagnostic	<a href="#">IDiagnosticInformaton</a>	Диагностическая информация о рабочем месте

Пример запроса:

```
Content-Type: application/json
Content-Length: ...
Accept: application/json
```

```
{
  "jsonrpc": "2.0",
  "method": "certrequests.parameters",
  "id": "2c48eb32-a0a8-405c-ade9-eed130605cba",
```

```

    "diagnostic": {
    }
}

```

#### Формат ответа:

Ключ	Значение	Описание
jsonrpc	2.0	Версия JSON-RPC протокола. Всегда «2.0»
result	ICertrequestsParameters	Объект со сведениями о параметрах операции
id	Уникальный идентификатор	Используется идентификатор, который указан в ссылке на операцию ( <a href="#">«Формат ссылки»</a> )

#### Пример ответа:

```

HTTP/1.1 200 OK
Connection: close
Content-Length: ...
Content-Type: application/json
Date: Sat, 08 Jul 2020 12:04:08 GMT

{
  "jsonrpc": "2.0",
  "result": {
    "operation": "GENERATE",
    "props": {
      "templateType": "JSONTemplate",
      "template": {
        "Description": "",
        "FriendlyName": "Пользователь",
        "RDN": [{
          "Oid": "2.5.4.3",
          "Name": "CN",
          "Length": 64,
          "LocalizedNames": "Общее имя",
          "SettingsValues": [],
          "DefaultValue": null,
          "ProhibitAnyValue": false,
          "ProhibitChange": false,
          "ProhibitEmpty": true
        }],
        "Extensions": {
          "KeyUsage": [{
            "Name": "cRLSign",

```



```

        "LocalizedName": "Автономное подписание списка отзыва
(CRL)",
        "DefaultValue": false,
        "ProhibitChange": true
    }],
    "ExtendedKeyUsage": [{
        "Name": "1.3.6.1.5.5.7.3.1",
        "LocalizedName": "Проверка подлинности сервера",
        "DefaultValue": false,
        "ProhibitChange": true
    }]
},
"MarkExportable": false
},
"id": "2c48eb32-a0a8-405c-ade9-eed130605cba"
}
}
}

```

#### 4.2.2. Отправка запроса на сертификат

После генерации запроса, он будет отправлен на сервер. Используются нотификации (уведомления), для которых не требуется ответ сервера.

##### Формат запроса:

Ключ	Значение	Описание
jsonrpc	«2.0»	Версия JSON-RPC протокола. Всегда «2.0»
method	«certrequests.base64»	Используемый метод. Всегда «certrequests.base64»
params	<a href="#">ICertificaterequestBase64Params</a>	Параметры, содержащие объект сертификата

##### Пример запроса:

Content-Type: application/json

Content-Length: ...

Accept: application/json

```

{
  "jsonrpc": "2.0",
  "method": "certrequests.base64",
  "params": {
    "id": "2c48eb32-a0a8-405c-ade9-eed130605cba",
    "certificaterequestBase64": "MIIFFDCCBMGgAwIBAgIQTm1HiybyfwV...4VVkDWbX/
n4=",
    "friendlyName": "000 Рога и Копыта"
  }
}

```

```
}
}
```

### 4.3. Типы данных

В данном разделе представлены типы данных, специфичные для команды **certrequests**.

#### 4.3.1. Интерфейс ICertrequestsParameters

Объекты данного типа описывают параметры команды.

Свойство	Тип	Описание
operation	string	Тип операции: генерация запроса на сертификат. Значение типа: <a href="#">CertrequestsOperation</a>
props	<a href="#">ICertrequestsOperationGenerateProps</a>	Параметры операции

#### 4.3.2. Тип CertrequestsOperation

Возможные операции с запросами на сертификат.

Значение	Описание
GENERATE	Генерация запроса на сертификат

#### 4.3.3. Интерфейс ICertrequestsOperationGenerateProps

Объекты данного типа описывают дополнительные свойства операции.

Свойство	Тип	Значение	Описание
headerText?	string	Необязательный параметр. Используется для отображения в заголовке окна. Максимальная длина: 40 символов	headerText?
descriptionText?	string	Необязательный параметр. Используется для отображения в сведениях об операции. Максимальная длина: 120 символов	descriptionText?
templateType	string	JSONTemplate	В качестве шаблона используется JSON типа IJSONTemplate
		CertificateTemplate	В качестве шаблона используется сертификат
template	<a href="#">IJSONTemplate</a> или <a href="#">ICertificateTemplate</a>	Шаблон для генерации запроса на сертификат	

#### 4.3.4. Интерфейс IJSONTemplate

Объекты данного типа описывают поля для генерации запроса на сертификат.

Свойство	Тип	Описание
Description	string	Описание шаблона
FriendlyName	string	Дружественное имя шаблона
RDN	<a href="#">IRDN[]</a>	Набор полей DN
Extensions	<a href="#">IRequestExtension</a>	Расширения
MarkExportable	boolean	Определяет экспортируемость ключей

#### 4.3.5. Интерфейс IRDN

Свойство	Тип	Описание
Oid	string	Описание шаблона
Name	string	Наименование OID
Length	number	Максимальная длина поля
LocalizedName	string	Локализованное наименование OID
SettingsValues	string[]	Список возможных значений
DefaultValue	string	Значение по умолчанию
ProhibitAnyValue	boolean	Флаг указывающий, что пользователю доступны только значения из массива SettingsValues
ProhibitChange	boolean	Флаг указывающий, что поле не может быть изменено
ProhibitEmpty	boolean	Флаг указывающий, что поле должно быть непустым

#### 4.3.6. Интерфейс IRequestExtension

Свойство	Тип	Описание
KeyUsage	<a href="#">IKeyUsage[]</a>	Описание шаблона
ExtendedKeyUsage	<a href="#">IExtendedKeyUsage[]</a>	Наименование OID

#### 4.3.7. Интерфейс IKeyUsage

Свойство	Тип	Описание
Name	string	Наименование OID
LocalizedName	string	Локализованное наименование OID
DefaultValue	boolean	Значение по умолчанию
ProhibitChange	boolean	Флаг указывающий, что поле не может быть изменено

#### 4.3.8. Интерфейс IExtendedKeyUsage

Свойство	Тип	Описание
Name	string	Наименование OID
LocalizedName	string	Локализованное наименование OID
DefaultValue	boolean	Значение по умолчанию
ProhibitChange	boolean	Флаг указывающий, что поле не может быть изменено

#### 4.3.9. Интерфейс ICertificateTemplate

Объекты данного типа используются для генерации запроса на сертификат, где запрос формируется по сертификату.

Свойство	Тип	Описание
certificateBase64	string	Сертификат в формате X.509 закодированный в Base64

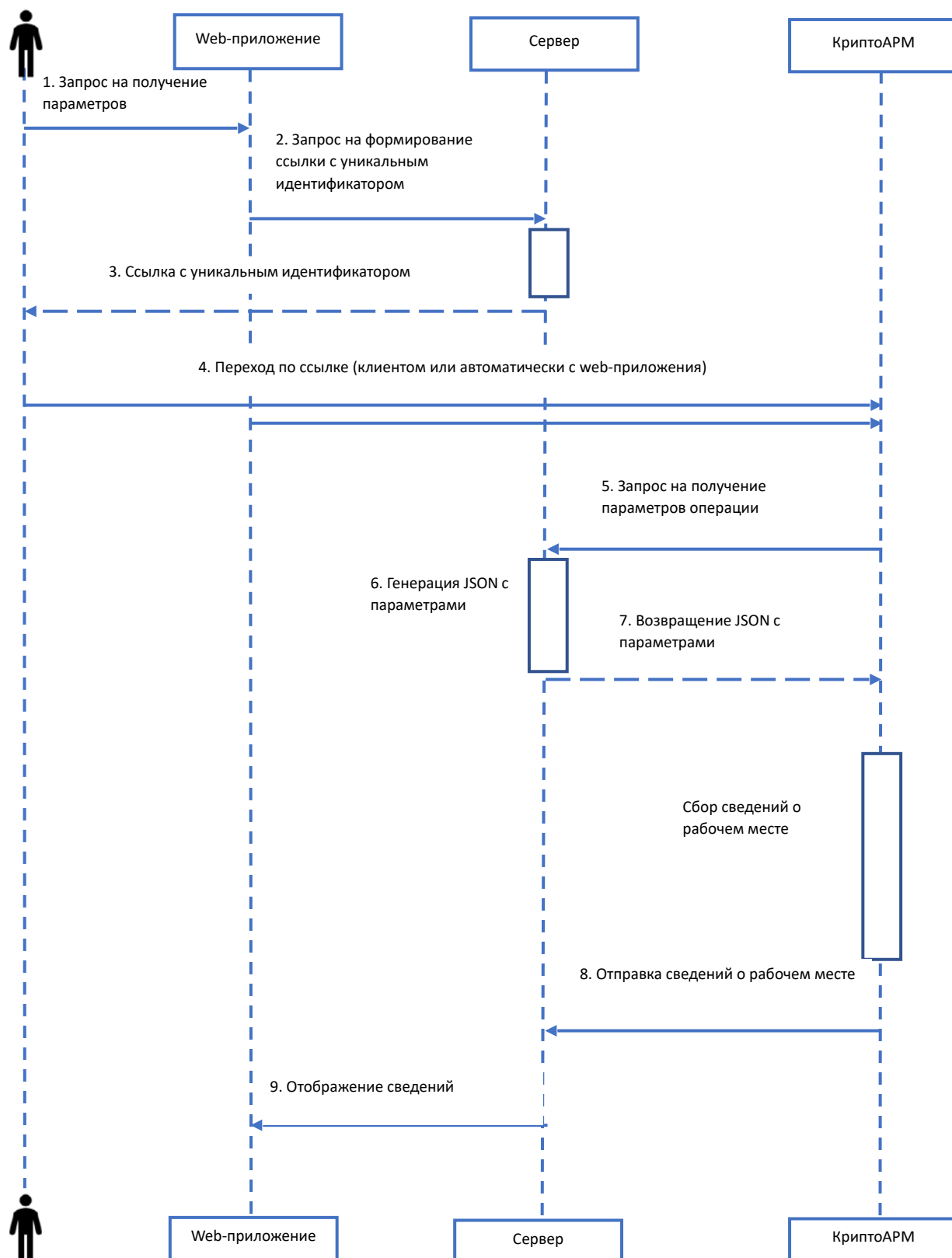
#### 4.3.10. Интерфейс ICertificaterequestBase64Params

Объекты данного типа описывают параметры запроса для метода certrequests.base64.

Свойство	Тип	Описание
id	string	Идентификатор транзакции
certificaterequestBase64	string	Запрос в формате X.509 закодированный в Base64
friendlyName	string	Дружественное имя субъекта

## 5. Команда diagnostics. Запросы на диагностику рабочего места

Команда **diagnostics** используется для диагностики рабочего места пользователя. Схема взаимодействия:



## 5.1. Формат ссылки

Для выполнения команды `diagnostics` должна быть сформирована ссылка вида:

**cryptoarm://diagnostics/<URL>/?id=<id>**

Здесь:

- **cryptoarm://** - зарегистрированный протокол
- **diagnostics** - выполняемая команда
- **<URL>** - ссылка, на которую КриптоАРМ будет слать запросы
- **id** – уникальный идентификатор транзакции

Пример:

cryptoarm://diagnostics/https://example.com/json?id=2c48eb32-a0a8-405c-ade9-eed130605cba

## 5.2. Описание запросов и ответов

Все запросы между КриптоАРМ и сервером ДОЛЖНЫ соответствовать спецификации протокола JSON-RPC 2.0. В качестве транспорта используется HTTP. Общее описание указано в разделе [1. Описание запросов и ответов](#).

### 5.2.1. Получение параметров операции

После получения команды **diagnostics** КриптоАРМ отправляет запрос на получение параметров операции.

**Формат запроса:**

Ключ	Значение	Описание
jsonrpc	«2.0»	Версия JSON-RPC протокола. Всегда «2.0»
method	«diagnostics.parameters»	Используемый метод. Всегда «diagnostics.parameters»
id	Уникальный идентификатор	Используется идентификатор, который указан в ссылке на операцию ( <a href="#">«Формат ссылки»</a> )
diagnostic	<a href="#">IDiagnosticInformaton</a>	Диагностическая информация о рабочем месте

Пример запроса:

```
Content-Type: application/json
```

```
Content-Length: ...
```

```
Accept: application/json
```

```
{
  "jsonrpc": "2.0",
  "method": "diagnostics.parameters",
  "id": "2c48eb32-a0a8-405c-ade9-eed130605cba",
```

```

    "diagnostic": {
    }
}

```

#### Формат ответа:

Ключ	Значение	Описание
jsonrpc	2.0	Версия JSON-RPC протокола. Всегда «2.0»
result	<a href="#">IDiagnosticsParameters</a>	Объект со сведениями о параметрах операции
id	Уникальный идентификатор	Используется идентификатор, который указан в ссылке на операцию ( <a href="#">«Формат ссылки»</a> )

#### Пример ответа:

```

HTTP/1.1 200 OK
Connection: close
Content-Length: ...
Content-Type: application/json
Date: Sat, 08 Jul 2020 12:04:08 GMT

```

```

{
  "jsonrpc": "2.0",
  "result": {
    "operation": ["CSP_ENABLED", "LICENSES", "VERSIONS"],
    "props": {
      "headerText": "Диагностика cryptoarm.ru",
      "descriptionText": "Выполняется диагностика рабочего места для
работы на портале"
    }
  },
  "id": "2c48eb32-a0a8-405c-ade9-eed130605cba"
}

```

#### 5.2.2. Отправка сведений о рабочем месте

Полученные сведения отправляется POST запросом. Используются нотификации (уведомления), для которых не требуется ответ сервера.

#### Формат запроса:

Ключ	Значение	Описание
jsonrpc	«2.0»	Версия JSON-RPC протокола. Всегда «2.0»

method	«diagnostics.information»	Используемый метод. Всегда «diagnostics.information»
params	Объект типа <a href="#">IDiagnosticsInformation</a>	Сведения о рабочем месте

Пример запроса:

```
Content-Type: application/json
Content-Length: ...
Accept: application/json
```

```
{
  "jsonrpc": "2.0",
  "method": "diagnostics.information",
  "params": {
    "id": "2c48eb32-a0a8-405c-ade9-eed130605cba",
    "CSP_ENABLED": true,
    "LICENSES": {
      "csp": {
        "status": true,
      },
      "cryptoarm": {
        "status": true,
        "type": "temporary",
        "expiration": "1591689487056",
      }
    },
    "VERSIONS": {
      "csp": "5.0.11753",
      "cryptoarm": "2.5.2",
    }
  }
}
```

## 5.3. Типы данных

В данном разделе представлены типы данных, специфичные для команды **diagnostics**.

### 5.3.1. Интерфейс IDiagnosticsParameters

Объекты данного типа описывают вид операции и её параметры

operation	string[]	Тип операции. Доступные значения типа: <a href="#">IDiagnosticOperation</a>
props	<a href="#">IDiagnosticsOperationProps</a>	Параметры операции



### 5.3.2. Тип IDiagnosticOperation

Возможные операции.

Значение	Описание
SYSTEMINFORMATION	Сведения о системе
CSP_ENABLED	Наличие КристоПро CSP
CADES_ENABLED	Доступность CADES
VERSIONS	Версии используемых компонентов (КристоАРМ, КристоПро)
PROVIDERS	Список криптопровайдеров
LICENSES	Статус лицензий
PERSONALCERTIFICATES	Наличие личных сертификатов

### 5.3.3. Интерфейс IDiagnosticsOperationProps

Интерфейс IDiagnosticsOperationProps описывает параметры операции.

Свойство	Тип	Описание
headerText?	string	Необязательный параметр. Используется для отображения в заголовке окна. Максимальная длина: 40 символов
descriptionText?	string	Необязательный параметр. Используется для отображения в сведениях об операции. Максимальная длина: 120 символов

### 5.3.4. Интерфейс IDiagnosticsInformation

Объекты данного типа описывают объекты, содержащие сведения о рабочем месте

Свойство	Тип	Описание
id	string	Идентификатор транзакции
API_VERSION	string	Версия поддерживаемого КристоАРМ API
SYSTEMINFORMATION	<a href="#">ISystemInformation</a>	Сведения о системе
CSP_ENABLED	boolean	Установлен или нет КристоПро CSP
CADES_ENABLED	boolean	Доступность CADES
VERSIONS	<a href="#">IVersions</a>	Версии компонентов
PROVIDERS	<a href="#">IProviders</a>	Сведения о провайдерах
LICENSES	<a href="#">ILicenses</a>	Сведения о лицензиях
PERSONALCERTIFICATES	<a href="#">ICertificateIdentityInfo[]</a>	Сведения о личных сертификатах (только идентификаторы)

### 5.3.5. Интерфейс ISystemInformation

Объекты данного типа содержат сведения о системе пользователя.

Свойство	Тип	Описание
type	string	Тип системы. Возможные значения: 'Linux', 'Darwin' и 'Windows_NT'
arch	string	Архитектура операционной системы. Возможные значения: 'arm', 'arm64', 'ia32', 'mips', 'mipsel', 'ppc', 'ppc64', 's390', 's390x', 'x32', и 'x64'
platform	string	Имя платформы. Возможные значения: 'aix', 'darwin', 'freebsd', 'linux', 'openbsd', 'sunos', и 'win32'
packageType?	string	Необязательный параметр. Тип поддерживаемого пакета (инсталлятора). Возможные значения: 'msi', 'pkg', 'rpm' или 'deb'

### 5.3.6. Интерфейс IVersions

Объекты данного типа содержат сведения о версиях.

Свойство	Тип	Описание
csp	string	Версия КристоПро CSP
cryptoarm	string	Версия КристоАРМ

### 5.3.7. Интерфейс IProviders

Объекты данного типа описывают доступность ГОСТ провайдеров.

Свойство	Тип	Описание
GOST2012_256	boolean	ГОСТ 2012-256
GOST2012_512	boolean	ГОСТ 2012-512

### 5.3.8. Интерфейс ILicenses

Объекты данного типа описывают статусы лицензии КристоАРМ и КристоПро CSP.

Свойство	Тип	Описание
csp	<a href="#">ILicenseInfo</a>	Сведения о лицензии на КристоПро CSP
cryptoarm	<a href="#">ILicenseInfo</a>	Сведения о лицензии на КристоАРМ

### 5.3.9. Интерфейс ILicenseInfo

Объекты данного типа описывают сведения о лицензии компонента.

Свойство	Тип	Описание
status	boolean	Действительна или нет лицензия
type	<a href="#">LicenseType</a>	Тип лицензии.
expiration?	string	Необязательный параметр. Дата истечения лицензии для триальных лицензий или подписок (в формате UTC)

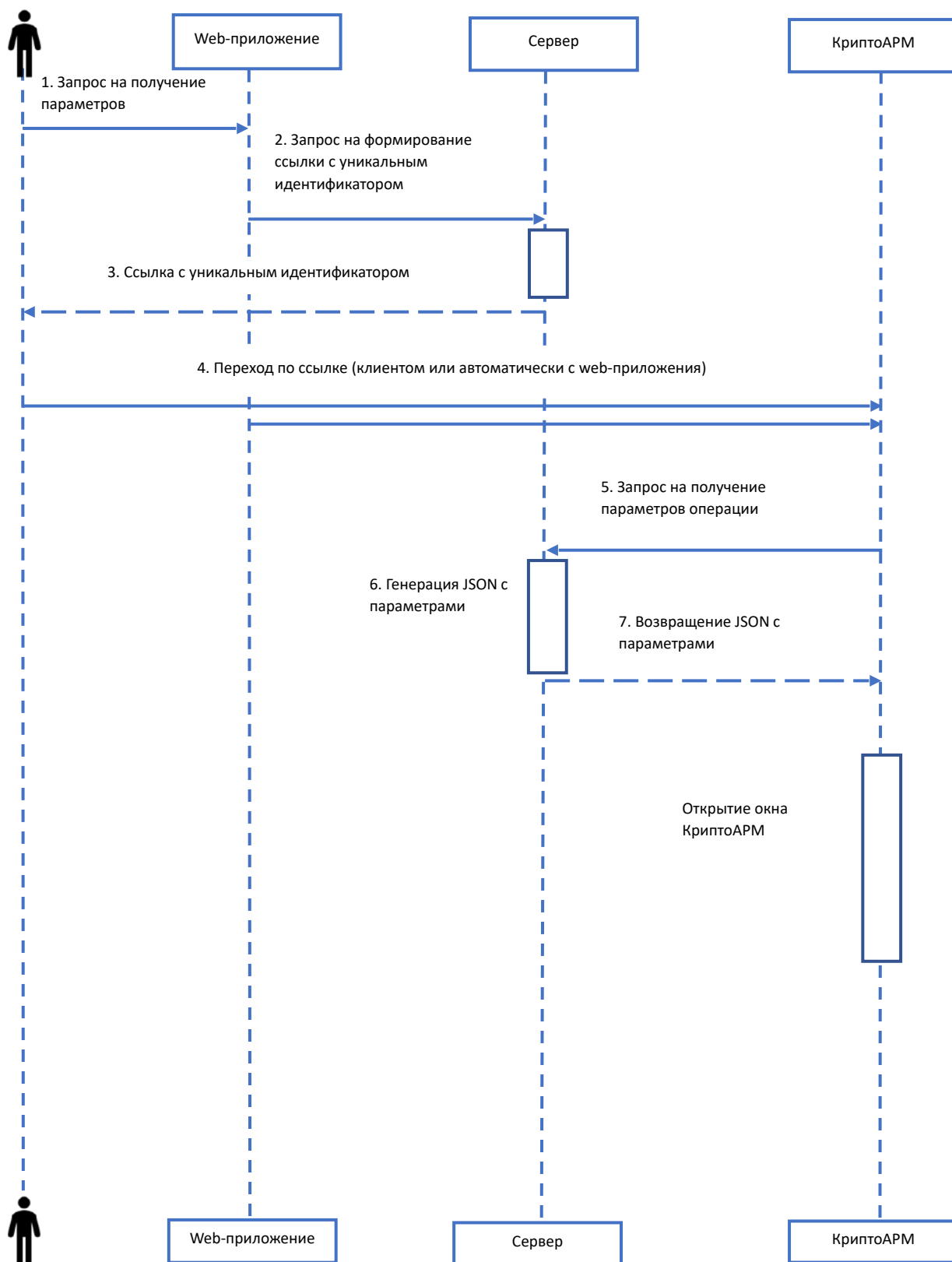
### 5.3.10. LicenseType Enum

Данное перечисление описывает возможные типы лицензий

Значение	Описание
Permanent	Постоянная
Subscription	Подписка
Daily	Дневная
Trial	Триальная

## 6. Команда startView. Открытие окна приложения

Команда **startView** используется для открытия конкретного окна приложения КристоАРМ ГОСТ. Схема взаимодействия:



## 6.1. Формат ссылки

Для выполнения команды **startView** должна быть сформирована ссылка вида:

**cryptoarm://startView/<URL>/?id=<id>**

Здесь:

- **cryptoarm://** - зарегистрированный протокол
- **startView** - выполняемая команда
- **<URL>** - ссылка, на которую КристоАРМ будет слать запросы
- **id** – уникальный идентификатор транзакции

Пример:

cryptoarm://startView/https://example.com/json?id=2c48eb32-a0a8-405c-ade9-eed130605cba

## 6.2. Описание запросов и ответов

Все запросы между КристоАРМ и сервером ДОЛЖНЫ соответствовать спецификации протокола JSON-RPC 2.0. В качестве транспорта используется HTTP. Общее описание указано в разделе [1. Описание запросов и ответов](#).

### 6.2.1. Получение параметров операции

После получения команды **startView** КристоАРМ отправляет запрос на получение параметров операции.

**Формат запроса:**

Ключ	Значение	Описание
jsonrpc	«2.0»	Версия JSON-RPC протокола. Всегда «2.0»
method	«startView.parameters»	Используемый метод. Всегда «startView.parameters»
id	Уникальный идентификатор	Используется идентификатор, который указан в ссылке на операцию ( <a href="#">«Формат ссылки»</a> )
diagnostic	<a href="#">IDiagnosticInformaton</a>	Диагностическая информация о рабочем месте

Пример запроса:

```
Content-Type: application/json
```

```
Content-Length: ...
```

```
Accept: application/json
```

```
{
```

```
  "jsonrpc": "2.0",
```

```
  "method": "startView.parameters",
```

```
  "id": "2c48eb32-a0a8-405c-ade9-eed130605cba",
```

```

    "diagnostic": {
    }
}

```

#### Формат ответа:

Ключ	Значение	Описание
jsonrpc	2.0	Версия JSON-RPC протокола. Всегда «2.0»
result	<a href="#">IStartViewParameters</a>	Объект со сведениями о параметрах операции
id	Уникальный идентификатор	Используется идентификатор, который указан в ссылке на операцию ( <a href="#">«Формат ссылки»</a> )

#### Пример ответа:

```

HTTP/1.1 200 OK
Connection: close
Content-Length: ...
Content-Type: application/json
Date: Sat, 08 Jul 2020 12:04:08 GMT

{
  "jsonrpc": "2.0",
  "result": {
    "uiView": "CERTIFICATES_MY",
    "props": {
      "headerText": "ИС cryptoarm.ru",
      "descriptionText": "Запрос на открытие окна"
    }
  },
  "id": "2c48eb32-a0a8-405c-ade9-eed130605cba"
}

```

## 6.3. Типы данных

В данном разделе представлены типы данных, специфичные для команды **startView**.

### 6.3.1. Интерфейс IStartViewParameters

Объекты данного типа описывают вид операции и её параметры

uiView	string	<p>Тип окна, которое нужно отобразить пользователю.</p> <p>Доступные значения:</p> <p>“MAIL” – окно почты</p> <p>“DOCUMENTS” – окно Документы</p> <p>“SIGN_AND_ENCRYPT” - окно подписи и шифрования</p> <p>“CERTIFICATES_MY” - хранилище личных сертификатов</p> <p>“CERTIFICATES_ADDRESS_BOOK”</p> <p>“CERTIFICATES_CA”</p> <p>“CERTIFICATES_ROOT”</p> <p>“KEYS” - окно ключей</p> <p>“ABOUT” - окно "О программе"</p> <p>“DIAGNOSTIC_PROBLEM_PERSONAL_CERTIFICATES” – окно диагностики, с проблемой отсутствия личных сертификатов</p>
props	<a href="#">IStartViewOperationProps</a>	Параметры операции

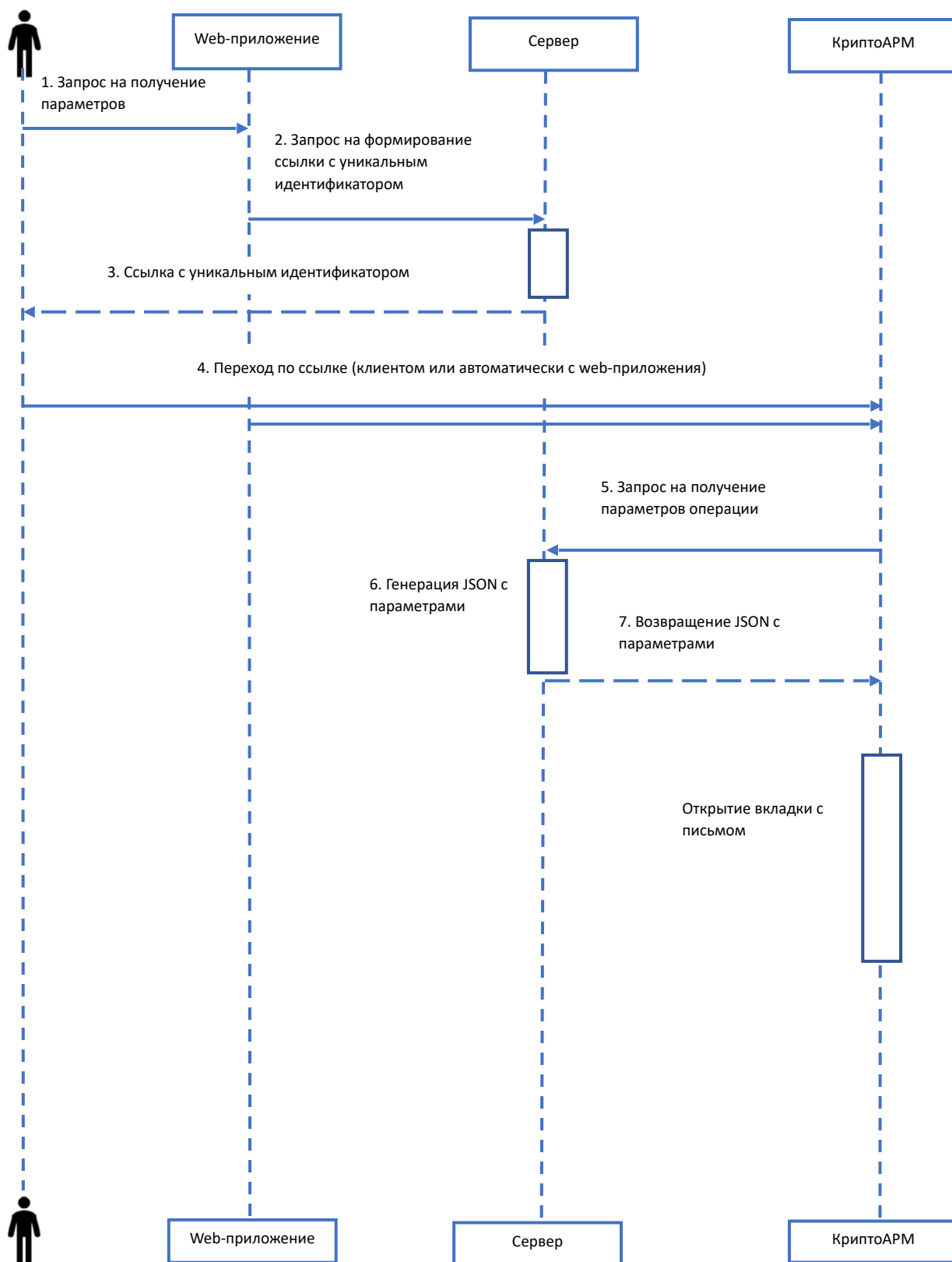
### 6.3.2. Интерфейс IStartViewOperationProps

Интерфейс IStartViewOperationProps описывает параметры операции.

Свойство	Тип	Описание
headerText?	string	Необязательный параметр. Используется для отображения в заголовке окна. Максимальная длина: 40 символов
descriptionText?	string	Необязательный параметр. Используется для отображения в сведениях об операции. Максимальная длина: 120 символов

## 7. Команда sendMail. Действия с электронными письмами

Команда **sendMail** используется для действий (отправка) с электронными письмами в КриптоАРМ ГОСТ. Схема взаимодействия:





## 7.1. Формат ссылки

Для выполнения команды **sendMail** должна быть сформирована ссылка вида:

**cryptoarm://sendMail/<URL>/?id=<id>**

Здесь:

- **cryptoarm://** - зарегистрированный протокол
- **sendMail**- выполняемая команда
- **<URL>** - ссылка, на которую КристоАРМ будет слать запросы
- **id** – уникальный идентификатор транзакции

Пример:

cryptoarm://sendMail/https://example.com/json?id=2c48eb32-a0a8-405c-ade9-eed130605cba

## 7.2. Описание запросов и ответов

Все запросы между КристоАРМ и сервером ДОЛЖНЫ соответствовать спецификации протокола JSON-RPC 2.0. В качестве транспорта используется HTTP. Общее описание указано в разделе [1. Описание запросов и ответов](#).

### 7.2.1. Получение параметров операции

После получения команды **sendMail** КристоАРМ отправляет запрос на получение параметров операции.

**Формат запроса:**

Ключ	Значение	Описание
jsonrpc	«2.0»	Версия JSON-RPC протокола. Всегда «2.0»
method	«sendMail.parameters»	Используемый метод. Всегда «sendMail.parameters»
id	Уникальный идентификатор	Используется идентификатор, который указан в ссылке на операцию ( <a href="#">«Формат ссылки»</a> )
diagnostic	<a href="#">IDiagnosticInformaton</a>	Диагностическая информация о рабочем месте

Пример запроса:

```
Content-Type: application/json
```

```
Content-Length: ...
```

```
Accept: application/json
```

```
{
  "jsonrpc": "2.0",
  "method": "sendMail.parameters",
  "id": "2c48eb32-a0a8-405c-ade9-eed130605cba",
```

```

    "diagnostic": {
    }
}

```

#### Формат ответа:

Ключ	Значение	Описание
jsonrpc	2.0	Версия JSON-RPC протокола. Всегда «2.0»
result	<a href="#">IMailParameters</a>	Объект со сведениями о параметрах операции
id	Уникальный идентификатор	Используется идентификатор, который указан в ссылке на операцию ( <a href="#">«Формат ссылки»</a> )

#### Пример ответа:

```

HTTP/1.1 200 OK
Connection: close
Content-Length: ...
Content-Type: application/json
Date: Sat, 08 Jul 2020 12:04:08 GMT

```

```

{
  "jsonrpc": "2.0",
  "result": {
    "operation": "SEND",
    "props": {
      "mailProps": {
        "to": ["test@example.com"],
        "cc": ["test1@example.com", "test2@example.com"],
        "bcc": ["test3@example.com"],
        "subject": "Message subject",
        "content": "text message content",
        "htmlContent": "<div>HTML message content</div>",
        "attachments": [{
          "name": "file1.txt",
          "url": "http://localhost:8080/public/files/file1.txt",
          "id": 1,
          "urlDetached": ""
        },
        {
          "name": "file2.txt",
          "url": "http://localhost:8080/public/files/file2.txt",
          "id": 2,

```

```

        "urlDetached": ""
    },
    {
        "name": "file4.pdf",
        "url": "http://localhost:8080/public/files/file4.pdf",
        "id": 4,
        "urlDetached": ""
    }
],
"sign": false,
"encrypt": false,
"deliveryReport": false,
"readReport": false
}
},
"id": "2c48eb32-a0a8-405c-ade9-eed130605cba"
}

```

### 7.3. Типы данных

В данном разделе представлены типы данных, специфичные для команды sendMail.

#### 7.3.1. Интерфейс IMailParameters

Объекты данного типа описывают вид операции и её параметры

Свойство	Тип	Описание
operation	string	Тип операции. Доступные значения: «SEND» – открыть новое сообщение (черновик) с переданными параметрами «OPEN» – открыть для чтения сообщение eml
props	<a href="#">IMailOperationProps</a>	Параметры операции

#### 7.3.2. Интерфейс IMailOperationProps

Объекты данного типа описывают вид операции и её параметры.

Свойство	Тип	Описание
eml?	<a href="#">IFile</a>	Необязательный параметр. Параметры для получения eml файла
mailProps?	<a href="#">IMailProps</a>	Необязательный параметр. Свойства для нового сообщения
extra?	Объект типа <a href="#">IExtra</a>	Необязательный параметр. Настройки операции. Для команды sendMail это свойство может содержать token, для скачивания eml с сервера

### 7.3.3. Интерфейс IMailProps

Объекты данного типа описывают параметры операции электронного сообщения

Свойство	Тип	Описание
to	string[]	Массив адресов получателей
cc?	string[]	Массив адресов получателей копии
bcc?	string[]	Массив адресов получателей скрытой копии
subject	string	Тема сообщения
content	string	Текстовое содержимое сообщения
htmlContent?	string	Форматированное содержимое сообщения в формате HTML
attachments?	<a href="#">IFile</a> []	Файлы для прикрепления к сообщению (вложения)
sign	boolean	Флаг указывающий на необходимость подписать сообщение
encrypt	boolean	Флаг указывающий на необходимость зашифровать сообщение
deliveryReport	boolean	Флаг включающий получение отчета о доставке
readReport	boolean	Флаг включающий получение отчета о прочтении