

## Краткая инструкция по быстрому старту для изучения API.

### ОБРАТИТЕ ВНИМАНИЕ:

Данная инструкция предназначена только для быстрого первоначального изучения работы API, и не является заменой основного документа по API, расположенного по ссылке [https://честныйзнак.пф/business/projects/medicines/#documents@for\\_developers](https://честныйзнак.пф/business/projects/medicines/#documents@for_developers)

1. Данные **Общество с ограниченной ответственностью "Типография для Типографий"**:
  - Учетная система:
    - client\_id: 22d12250-6cf3-4a87-b439-f698cfddc498,
    - client\_secret: 3deb0ba1-26f2-4516-b652-931fe832e3ff
  - Участник:
    - sys\_id: 57663471-2d09-4832-9b76-78095fbd0d43
2. Данные **Общество с ограниченной ответственностью "Автомойка-Чисто"**:
  - Учетная система:
    - client\_id: 2cabd9b7-6042-40d8-97c2-8627f5704aa1,
    - client\_secret: 1713da9a-2042-465c-80ba-4da4dca3323d
  - Участник:
    - sys\_id: 86325e0c-9a23-4547-ad8a-219b4fc7fd03

Для работы с API на тестовом стенде Песочница необходимо использовать тестовые сертификаты, выданные в дополнение к данной инструкции.

### Авторизация пользователей учетных систем на тестовом стенде Песочница

#### Авторизация тестовым участником

На тестовом стенде API необходимо авторизоваться тестовым участником и получить «Код аутентификации», используя метод API

POST <https://api.sb.mdlp.crpt.ru/api/v1/auth>

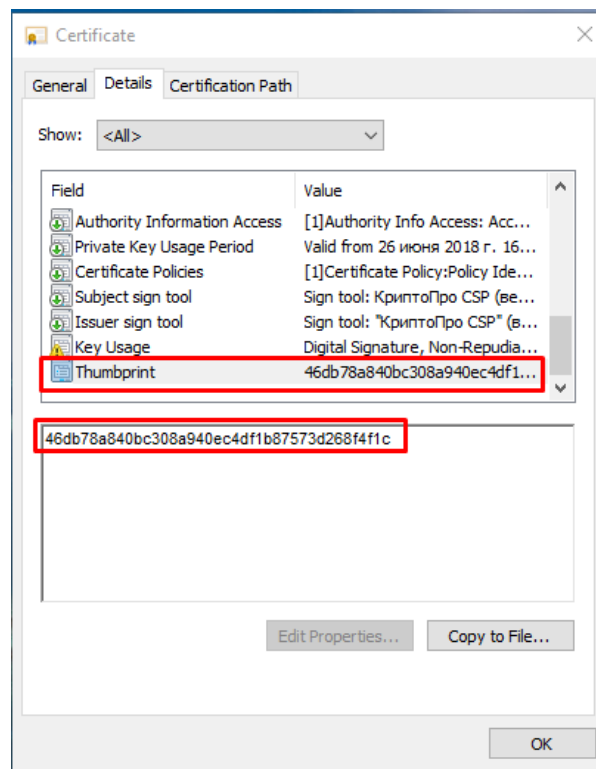
Content-Type: application/json; charset=UTF-8

```
{
  "client_secret": "3deb0ba1-26f2-4516-b652-931fe832e3ff",
  "client_id": "22d12250-6cf3-4a87-b439-f698cfddc498",
  "user_id": "46db78a840bc308a940ec4df1b87573d268f4f1c",
  "auth_type": "SIGNED_CODE"
}
```

Ответ:

```
{
  "code": "b74d04c3-e784-421e-84f9-dde0fc1bb3c9"
}
```

Подробнее про метод получения кода аутентификации описан в документации «Протокол обмена интерфейсного уровня» пункт 3.2



В параметры client\_secret и client\_id можно использовать GUID-ы из данного примера;

В параметр user\_id использовать либо отпечаток сертификата (thumbprint, см. скрин), либо серийный номер сертификата в десятичной форме. **Не пытайтесь использовать отпечаток (thumbprint) взятый из примера.**

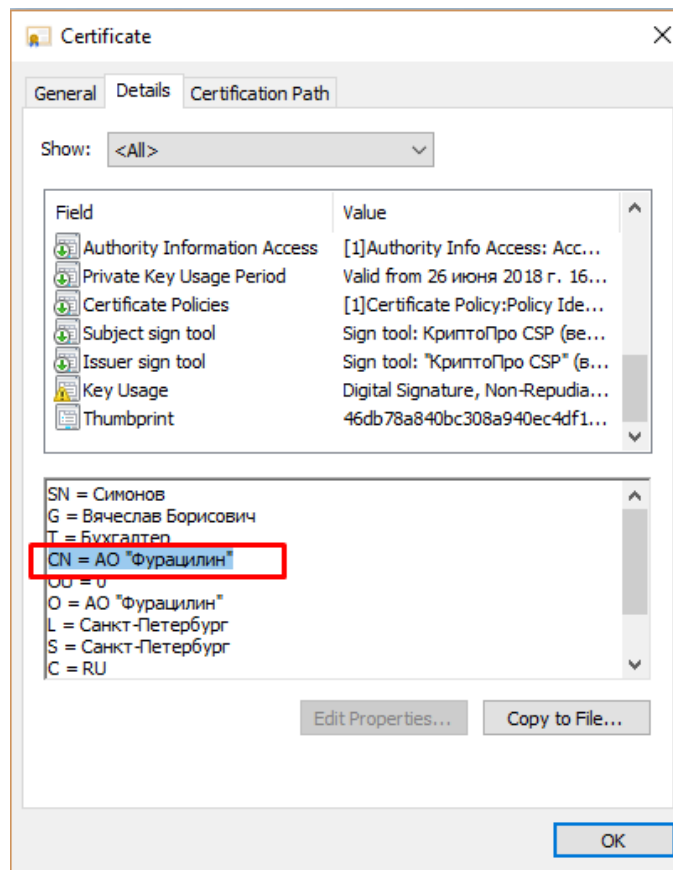
#### Подписать полученный код

В данном примере будет показан метод получения сигнатуры электронной подписи с использованием CSP Крипто Про версии 4.0

1. Необходимо создать файл с расширением txt.
2. В текстовый файл записать код полученный в ответе в методе аутентификации (обратите внимание, что никаких лишних символов не должно быть (пробелов, переводов строк, табуляций и т.п.).

```
1 b74d04c3-e784-421e-84f9-dde0fc1bb3c9
```

3. Подписать УКЭП текстовый файл:
  - а. Посмотреть CN своего сертификата. Если в наименовании сертификата присутствуют кавычки, то при вводе данных сертификата, кавычки вводить не нужно.



- b. Открыть консоль (cmd) в папке с установленным программным обеспечением CSP Кripto Про. (По умолчанию путь c:\Program Files\Crypto Pro\CSP)
- c. В консоли ввести следующую команду:

```
csptest -sfsign -sign -in <УКАЗАТЬ ПОЛНЫЙ ПУТЬ К ФАЙЛУ>test.txt -out
<УКАЗАТЬ ПОЛНЫЙ ПУТЬ К ФАЙЛУ>out.txt -my "АО Фурацилин" -detached -
base64 -add
```

Исходя из примера на выходе будет получен файл **out.txt**, в котором будет указана сигнатура подписи. **В сигнатуре подписи также необходимо убрать заголовок строки и символы окончания строки.** Подготовленную сигнатуру вставить в строку "signature" метода /api/v1/token

Получение ключа сессии для УКЭП.

Метод получения ключа сессии

```
POST https://api.sb.mdlp.crpt.ru/api/v1/token
```

```
Content-Type: application/json; charset=UTF-8
```

```
{
```

```
  "code": "b74d04c3-e784-421e-84f9-dde0fc1bb3c9",
```

```
  "signature":
```

```
    "MIAGCSqGSIB3DQEHAgCAMIACAQExDDAKBgYqhQMCAgkFADCABgkqhkiG9w0BBwEAAKCAMIIBjjCCAT2gAwIBAgIEWWJzHzAIBgYqhQMCAgMwMTELMakGA1UEBhMCU1UxEjAQBgNVBAoMCUNyeXB0b1BybzeOMAwGA1UEAwwFQWxpYXNwHhcnMTcxMTEzMTEzMjI4WhcNMTgxMTEzMTEzMjI4WjAQMQuwCQYDVQGEwJSVTESMBAGA1UECgwJQ3J5cHRvUHJvMQ4wDAYDVQQDDAVBbG1hczBjMBwGBiQFAwICEzASBgqhQMCAiQABgcqhQMCAh4BA0MABEAIWARZAiI81k4i4Gz8EC7Ic01653JX5PCUfvqCBTpLduYtbTwLowmGFcZzw9bwsxQpALqhcdRHxtx1UEeNKJuMozswOTA0BgNVHQ8BAf8EBAMCA+gwEwYDVR01BAwwCgYIKwYBBQUHAWIwEgYDVR0TAQH/BAgwBgEB/wIBBTAIBgYqhQMCAgMDQQL9CrIk0EgnMVR1J5dKbfXVFrhJxGxztFkTdmGkGJ6gHywB5Y9KpP67pv7I2bP1m1ej9hu+C17GSJrWgMgq+UZAAXggEFMIIBAQBATA5MDExCzAJBgNVBAYTA1JVMRIwEAYDVQQKDA1DcnlwdG9Qcm8xZjAM
```

```
BgNVBAMMBUFsaWFzAgRZYnMfMAoGBiqFAwICCQUAoGkwGAYJKoZIhvcNAQkDMQsGCSqG
SIb3DQEHAQcBgkqhkiG9w0BCQUxDxcNMTcxMTMwMDkyMTU5WjAvBgkqhkiG9w0BCQQx
IgQg+vo77XxU6eHIpoZxOwIk+gWY+JSv9lfa1TgOM/UW8aowCAYGKoUDAgIDBECJhWsL
OQ1Wn4KntnGNL8KNZSbpJ35E8NjyEio+Im5dLiLoqZotxGF1WmYvBaPy8FHcAsTqaE
EJna7+CecOi/AAAAA
```

```
}
```

Ответ:

```
{
```

```
  "token": "fd780e64-b753-45c0-a375-31218210eadf",
```

```
  "life_time": 30
```

```
}
```

**Не пытайтесь использовать сигнатуру из примера.**

Процедура Аутентификации и Авторизации с помощью УКЭП закончена.

Подпись и отправка схем с использованием УКЭП.

**Обратите внимание:** Все методы API должны

вызываться с использованием HTTPS протокола. Для взаимодействия по https используется ГОСТ сертификат (ГОСТ Р 34.10-2012 сертификат).

Для отправки документа используется метод API: /documents/send

Примеры схем и форматы описаны на сайте

<https://честныйзнак.рф/business/projects/medicines/#documents>

Рассмотрим пример отправки документа:

```
<documents xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
version="1.27"><register_end_packing action_id="311"><subject_id>
00000000100930</subject_id>
```

**Крайне важно соблюдать кодировку UTF-8. Убедиться, что символы пробелов, переводов строк (символы перевода строки в Unix и Windows различные) совпадают в подписываемом и отправляемом теле запроса. В противном случае при проверки подписи результат проверки будет отрицательным. Подпись должна производиться один-в-один того, что отправляется в поле document.**

Подписать документ

В данном примере будет показан метод получения сигнатуры электронной подписи с использованием CSP Кripto Про версии 4.0

1. Необходимо создать файл с расширением txt. Например, **doc.txt**
2. В текстовый файл вставить тело документа

```

1 <documents xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.19">CRLF
2   <register_end_packing action_id="311">CRLF
3     <subject_id>000000000000517</subject_id>CRLF
4     <operation_date>2018-08-22T15:00:00+05:00</operation_date>CRLF
5     <order_type>1</order_type>CRLF
6     <series_number>100000001</series_number>CRLF
7     <expiration_date>30.03.2020</expiration_date>CRLF
8     <gtin>11170012610151</gtin>CRLF
9     <tnved_code>3004</tnved_code>CRLF
10    <signs>CRLF
11      <sgtin>07091900400001TRANSF2000021</sgtin>CRLF
12    </signs>CRLF
13  </register_end_packing>CRLF
14 </documents>

```

### 3. Подписать УКЭП текстовый файл.

#### Пример подписи с использованием Крипто Про CSP версии 4.0.

Открыть консоль (cmd) в папке с установленным программным обеспечением CSP Крипто Про (по умолчанию путь c:\Program Files\Crypto Pro\CSP)

В консоли ввести следующую команду:

```
csptest -sfsign -sign -in <УКАЗАТЬ ПОЛНЫЙ ПУТЬ К ФАЙЛУ>doc.txt -
out <УКАЗАТЬ ПОЛНЫЙ ПУТЬ К ФАЙЛУ>signed_doc.txt -my "АО Фурацилин" -
detached -base64 -add
```

Замечание: если в наименовании сертификата присутствуют кавычки, то при вводе данных сертификата, кавычки вводить не нужно.

Замечание: указывайте сертификат которым вы прошли процедуру аутентификации.

Исходя из примера на выходе будет получен файл **signed\_doc.txt**, в котором будет указана сигнатура подписи. В сигнатуре подписи необходимо удалить символы переноса строк.

Полученную сигнатуру подписи использовать для отправки документа (см. ниже следующий шаг).

#### Перевести тело документа в Base64

```
PGRvY3VtZW50cyB4bWxuczp4c2k9Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvWE1MU2No
ZW1hLWluc3RhbmNlIiB2ZXJzaW9uPSIxLjE5Ij48cmVnaXN0ZXJfZW5kX3BhY2tpbmcg
YWN0aW9uX2lkPSIzMTEiPjxzZdWJqZWNOX2lkPjAwMDAwMDAwMDAwNTE3PC9zdWJqZWNO
X2lkPg==
```

#### Отправить документ

Сгенерировать новый идентификатор в формате UUID (для параметра request\_id)

```
POST https://api.sb.mdip.crpt.ru/api/v1/documents/send
```

```
Content-Type: application/json; charset=UTF-8
```

```
Authorization: token fd780e64-b753-45c0-a375-31218210eadf
```

```
{
```

"PGRvY3VtZW50cyB4bWxuczp4c2k9Imh0dHA6Ly93d3cudzMub3JnLzlwMDEvWE1MU2NoZWlhLW1uc3RhbmNlIiB2ZXJzaW9uPSIiXjE5Ij48cmVnaXN0ZXJfZW5kX3BhY2tpbmcyYWN0aW9uX2lkPSIzMTEiPjxzZdWJqZWNOX2lkPjAwMDAwMDAwMDAwNTE3PC9zdWJqZWNOX2lkPq==" ,

MIINHwYJKoZIhvcNAQcCoIINEDCCDQwCAQEEdjAMBggqhQMHAQEACgUAMAsGCSqGSIB3DQEHAaCCCQMwggj/MIIIRqADAgECAGPKStMAAAMAA0TkMagGBiqFAWICazCCAUGxGDABgUqhQNkARINMTAzNzcwMDA4NTQ0NDEaMBGCCqFawOBawEBEGwwMDC3MTcxMDc5OTExOTA3BgNVBAkeMARDBDsALGAgBCEEQWRJBFEEMGRBBDoEOAQ5ACAEMgQwBDsALAAGBDQA LgAgADEAODeHMB8GA1UECB4YADcANwAgBDMALgAgBBwEPgRBBDoEMgQwMRUwEwYDVQQH HgwEHAQ+EEEEOGqyBDAXIDAeBgkqhkiG9w0BCQEWEWluZm9AY3J5cHRvcHJvLnJlMQsw CQYDVQQGEwJSVTEpMCcGA1UECh4gBB4EHgQeACAAIgQaBCAEGAQfBCIEHgAtBB8EIAQe ACIXQTa/BgNVBAMEOAQiBDUEEQRCBD4EMgRLBDkAIAQjBCYAIAQeBB4EHgAgACIEGgQG BBgEHwQiBB4ALQQfBCAEHgAiMB4XDTE4MDYwNTE0NDQwMFoXDTTE4MDkwNTE0NTQwMFo ggIEMRGwFgYFKoUDZAESDTAwMDAwMDAwMDAwMDMXGjAYBggqhQMDgQMBARIMMDAwMDAw MDawMDAxMR4WHAYJKoZIhvcNAQkBFG9leGFtcGxlQG1haWwucnUxCzAJBgNVBAYTAlJV MSYwJAYDVQQIDB3QodCw0L3QutGCLdCf0LXRgtC10YDQsdGD0YDQszEmMCQGA1UEBwwd 0KHQsNC90LrRgi3qn9C10YLQtDGA0LHRg9GA0LMxNDAYBgNVBAOMK9Ci0LXQv9C70L7Q v9GA0LjQsdC+0YDRgdC10YDQstC40YEg0YLQtDGB0YIXcJAIBGNVBASMATExpZA9BgNV BAMMNtCe0YHQUNc/0L7QsiDQkNC70YzQsdC10YDRgiDQkNC70LXQutGB0LDQvdC00YDQ vtCy0LjRhzeuMCwGA1UECQw10YLQtDGB0YLQvtCy0YvQuSDQsNC00YDQtdGBINC00L7Q vCA0NDEFMB0GCSqGSIB3DQEJAGwQ0JTQUnga0LXQutGC0L7RgDEwMC4GA1UEDawn0JPQ tdc90LXRgNCw0LvRjNC90YvQuSDQtNC40YDQtdC60YLQvtGAMTIwMAYDVQQQCdnQkNC7 0YzQsdC10YDRgiDQkNC70LXQutGB0LDQvdC00YDQvtCy0LjRhzeVMBMGALUEBAWM0J7R gdC40L/QvtCyMGYwHwYIKoUDBWEBaqEWewYHKoUDAGIkAAAYIKoUDBWEBAGIDQwAEQPXR befMOOOIm39XX4hNPMDnlv9f4Ckf5QMOSBVcfTHiLx2uzD5ia1ll+GniVAloW13A+wvsh xpsiGoQoJc2vn1+jggSzMIIErzAOBgNVHQ8BAf8EBAMCBPAwJgYDVR01BB8wHQYIKwYB BQUHAWQGBByqFAWICIGYGCCS GAQUFBwMCMB0GA1UdDgQWBBrCH97RoSiItjcIGebTvQbQk CEhwYDCCAYkGA1UdIwSCAYAwggF8gBQRsha0ZoICrPDhqkCGeAFxRZ0z46GCAVCkggFM MIIBSDEYMBYGBSqFA2QBEg0xMDM3NzAwMDg1NDQ0MRowGAYIKoUDA4EDAQESDDAwNzcz NzEwNzk5MTE5MDcGA1UECR4wBEMEOWAuACAeIQRBBeKEUQQyBEEEOGQ4BDkAIAQyBDAE OwAsACAENAAuACAAMQA4MSEwHwYDVQQIHhgANWA3ACAEMwAuACAeHAQ+EEEEOGqyBDAX FTATBgNVBAceDAQcBD4EQQQ6BDIEMDEgMB4GCSqGSIB3DQEJARyRaW5mb0BjcnldwG9w cm8ucnUxCzAJBgNVBAYTAlJVMskwJwYDVQQKHIAEHgQeBB4AIAAiBB0EIAQYBB8EIgQe AC0EHwQgBB4AIjFBMD8GA1UEAx44BCIENQRBBEIEPgQyBEseOQAgBCMEJgAgBB4EHgQe ACAAIGQaBCAEGAQfBCIEHgAtBB8EIAQeACKCEESH21dJk2CeShb2gnNE/xcwXAYDVR0f BFUwUzBRoe+gTYZLaHR0cDovL3d3dy5jcnldwG9wcM8ucnUvcMevY2RwLzJiYjIxMDM0 NjY4MjAyYWNmMGUXYWE0MDg2NzgWMTcxNDU5ZDMzMzTMuY3JsMIGxBggRBgEFBQcBAQSB pDCBoTA0BggrBgEFBQcwAYYoAHR0cDovL3d3dy5jcnldwG9wcM8ucnUvb2NzcG5jMi9v Y3NWLnNyZjAyBggrBgEFBQcwAYYmaHR0cDovL3d3dy5jcnldwG9wcM8ucnUvb2NzcDIv b2NzcC5zcmYwNQYIKwYBBQUHMAKGKh0dHA6Ly93d3cuY3J5cHRvcHJvLnJlL3JhL2Nk cC9jYWNlcjMuY3J0MCSGA1UdEAQkMCKADzIwMTGwnJA1MTQ0NDawWoEPMjAxODA5MDUX NDQ0MDBaMB0GA1UdIAQWMBQwCAYGKoUDZHEBMAGGBiqFA2Rxaja0BgUqhQNkbwQrdCnQ mtGA0LjQv9GC0L7Qn9GA0L4gQ1NQICjQstC10YDRgdC40Y8gMy42KTCCATMGBSqFA2Rw BIIBKDCCASQMKyLQmtGA0LjQv9GC0L7Qn9GA0L4gQ1NQIia0LLQtdGA0YHQUNGPIDMu NikMUyLQo9C00L7RgdGC0L7QstC10YDRj9GO0YnQuNC5INGGOLXQvdGC0YaGitCa0YDQ uNC/0YLQvtCf0YDQviDQo9CmIIdQstC10YDRgdC40LggMS41DE/QodC10YDRgtC40YTQ uNC60LDRgiDRgdC+0L7RgtCy0LXRgtGB0YLQstC40Y8g4oSWINCh0KQvMTI0LTi3Mzgg 0L7RgiAwMS4wNy4ymDE1DE/QodC10YDRgtC40YTQuNC60LDRgiDRgdC+0L7RgtCy0LXR gtGB0YLQstC40Y8g4oSWINCh0KQvMTI4LTi3Njgg0L7RgiAzMS4xmi4ymDE1MAgGBiqF AwICAwnBAooGYI1/8v4f0Di2ih0Q1AP7X2U9DnsfpY+IZABpBI6/8Iksnt/JBmRgBe6u uF3wtewkgD8YKJkUqXbblylRd+0xggPhMIID3QIBATCCAvgwgG FIMRGwFgYFKoUDZAES DTEwMzc3MDAwODU0NDQxGjAYBggqhQMDgQMBARIMMDA3NZE3MTA30TKxMTknWYDVQQJ HjAEQwQ7AC4AIAQHBMESQRRBDIEQQQ6BDgEQOAgBDIEMAQ7ACWAIAQ0AC4AIAAXAdgx ITAfBgNVBAGEGAA3ADcAIAQZAC4AIAQCBD4EQQQ6BDIEMDEVMBMGALUEBX4MBBwEPgRB BDOEMqQwMSAwHqYJKoZIhvcNAQkBfhFpbmZvQGNyeXB0b3Byby5vdTELMAkGA1UEBHMCM

ULUxKTAnBgNVBAoeIAQeBB4EHgAgACIEGgQgBBgEHwQiBB4ALQQfBCAEHgAiMUEwPwYD  
VQQDHjgEIgQ1BEEEQgQ+BDIESwQ5ACAEIwQmACAEHgQeBB4AIAAiBB0EIAQYBB8EIgQe  
AC0EHwQgBB4AIgIKSkrtAAADAANE5DAMBggqhQMHAQECAgUAoIICHDAYBgkqhkiG9w0B  
CQMxCwYJKoZIhvcNAQcBMBwGCSqGSIb3DQEJBTEPFw0xODA4MjKxNDElMjRaMCMGCSqG  
SIb3DQEJBDEiBCB9An2pcj1/MOfuHX8cjgNNzDp7wG+DWLhkqIeeSY04fjCCAa8GCyqG  
SIb3DQEJEAIVMYIBNjCCAZowggGWMIIIBKjAKBggqhQMHAQECAgQg2kW4PyhwpD0j1ZI9  
AHkdyAY1j6oDX6EFrMufeO/L/fYwggFgMIIBUKSCAUwwggFIMRgwFgYFKoUDZAESDTEw  
Mzc3MDAwODU0NDQxGjAYBgqhQMDgQMBARIMMDA3NzE3MTA3OTkxMTkwNwYDVQQJHjAE  
QwQ7AC4AIAQhBEMESQRRBDIEQQQ6BDgEOQAqBDIEMAQ7ACwAIAQ0AC4AIAAxADgxITAf  
BgNVBAgeGAA3ADcAIAQzAC4AIAQcBD4EQQQ6BDIEMDEVMBMGA1UEBx4MBBwEPgRBBDoE  
MgQwMSAwHgYJKoZIhvcNAQkBFhFpbmZvQGNyeXB0b3Byby5ydTELMAkGA1UEBhMCU1Ux  
KTAnBgNVBAoeIAQeBB4EHgAgACIEGgQgBBgEHwQiBB4ALQQfBCAEHgAiMUEwPwYDVQQD  
HjgEIgQ1BEEEQgQ+BDIESwQ5ACAEIwQmACAEHgQeBB4AIAAiBB0EIAQYBB8EIgQeAC0E  
HwQgBB4AIgIKSkrtAAADAANE5DAMBggqhQMHAQEBAQUABECGawmGd26uEblqctxuxmy6  
K8+WAx148Bksn0ENrxm8Zbb8PAT2yU8IDARWmPIUAMGPENn85anoRasZJqpyYI4y",

"request\_id ": "2ba0919b-6dbc-4729-b886-df19dd60566d"

(request\_id: должен быть всегда уникальным)

}

Ответ:

{

"document\_id": "cc7890a8-39ac-4ba9-ae9a-b20f406a781c"

}

Не используйте данные взятые из данного примера.

Получить ссылку для просмотра документа

GET

https://api.sb.mdpl.crpt.ru/api/v1/documents/download/{document\_id}

Content-Type: application/json; charset=UTF-8

Authorization: token fd780e64-b753-45c0-a375-31218210eadf

Ответ

{

"link": "https://api.sb.mdpl.crpt.ru/webdav/upload/  
cc7890a8-39ac-4ba9-ae9a-b20f406a781c/cc7890a8-39ac-4ba9-ae9a-  
b20f406a781c"

}

Скачать документ

GET https://api.sb.mdpl.crpt.ru/webdav/upload/cc7890a8-39ac-4ba9-  
ae9a-b20f406a781c/cc7890a8-39ac-4ba9-ae9a-b20f406a781c

Authorization: token fd780e64-b753-45c0-a375-31218210eadf

Получить ссылку на квитанцию

GET https://api.sb.mdpl.crpt.ru/api/v1/  
documents/040fb1c3-69e4-43f7-b656-259a742eb944/ticket

Authorization: token d780e64-b753-45c0-a375-31218210eadf

Ответ

```
{  
  "link": "https://api.sb.mdlp.crpt.ru/webdav/  
upload/040fb1c3-69e4-43f7-b656-259a742eb944/  
ticket_040fb1c3-69e4-43f7-b656-259a742eb944"  
}
```

[Скачать квитанцию](#)

GET **https://api.sb.mdlp.crpt.ru/webdav/upload/040fb1c3-69e4-43f7-b656-259a742eb944/ticket\_040fb1c3-69e4-43f7-b656-259a742eb944**

Authorization: token **d780e64-b753-45c0-a375-31218210eadf**