

Quick start for API

NOTE:

The following manual is meant for a quick initial review of API work and is no substitute for the basic API document available at the following link:

https://честныйзнак.рф/business/projects/medicines/#documents@for_developers

To connect to API test environment the following environment domain must be added to operating system hosts

```
185.196.171.27 api.stage.mdlp.crpt.ru
```

API test environment registration of an accounting system user

The system doesn't imply regular operations with participants' test data samples **starter_resident_1** and **starter_resident_2** published in the document "Interface level communication protocol". In particular, they should not be considered as full-featured accounts due to a dummy mark "non-resident" applied precisely for test environment to facilitate the use. It allows to log in with a username and password. In fact, user data belongs to participants-residents.

1. **starter_resident_1** data:
 - user_id (aka login): starter_resident_1
 - Accounting system:
 - client_id: 01db16f2-9a4e-4d9f-b5e8-c68f12566fd5,
 - client_secret: 9199fe04-42c3-4e81-83b5-120eb5f129f2
 - Participant:
 - sys_id: 9dedee17-e43a-47f1-910e-3a88ff6bc81b
2. **starter_resident_2** data:
 - user_id (aka login): starter_resident_2
 - Accounting system:
 - client_id: c9c307fd-dcb0-4505-8178-13ba2f362339,
 - client_secret: 4d3a2f91-992f-4604-a8a1-71378a1eb75e
 - Participant:
 - sys_id: 6f6fa779-b637-4234-9117-8ac4c1a9a81c

For correct work and for testing API methods it's necessary to create native users that you are going to work with in future.

1. Issue a test certificate of a higher-level qualified e-signature for an accounting system user.

(Comment: Registration of foreign users is not anticipated on API test environment)

A test certificate can be issued on the following page:

<https://www.cryptopro.ru/ui/Register/RegGetSubject.asp>

It requires prior installation of CryptoPro CSP from 4.0 version.

Samples how to fill in the required fields of a certificate:

- 1) For a legal person

Welcome to the self-registration page!

You need to create a registration request in the system. To do this, please fill in the fo

Surname:	Test Surname
Name:	Test User
Position/title:	Position name
Unstructured name:	Unstructured name
Address:	Test address street
Common name:	Common name test
Unit:	Unit name
Organization:	Organization name
City:	Test City
Region:	Regoin Test
Country/region:	CD ▼
Email:	example@mail.ru
INN:	001234567890
Bin:	1234567890123
SNILS:	
OGRN:	
The key phrase:	
Additional information:	

Fig.1 Sample how to fill in the required fields of a certificate for legal persons

2) For an individual entrepreneur

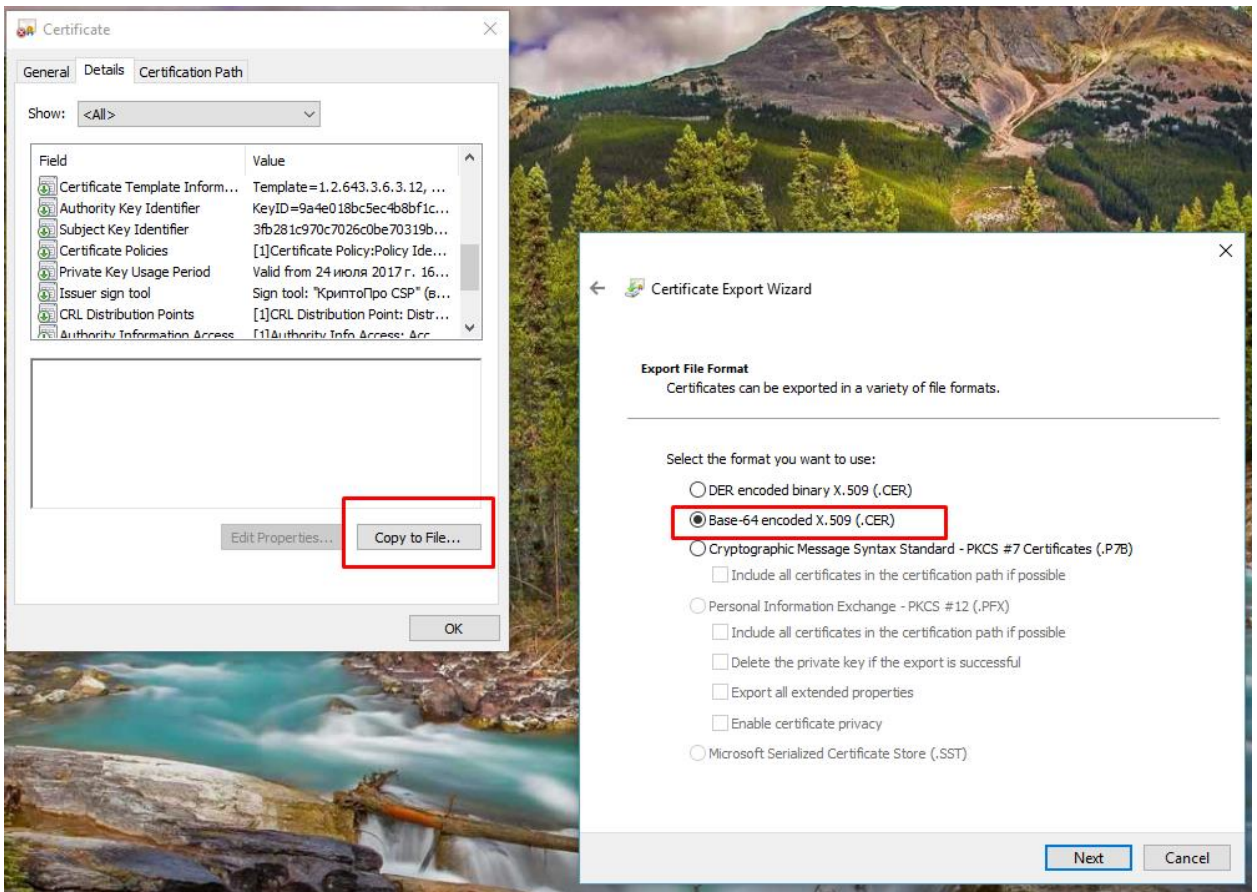
Welcome to the self-registration page!

You need to create a registration request in the system. To do this, please fill in the fo

Surname:	Test
Name:	Test User
Position/title:	
Unstructured name:	Unstructured name
Address:	Test address
Common name:	Test UKP
Unit:	
Organization:	
City:	Test City
Region:	Regoin Test
Country/region:	CD ▼
Email:	example@mail.ru
INN:	123456789012
Bin:	
SNILS:	12345678901
OGRN:	123456789012345
The key phrase:	
Additional information:	

Fig.2 Sample how to fill in the required fields of a certificate for individual entrepreneurs

An issued test certificate should be prepared for future work. Therefore it's necessary to convert it into *Base64* format.



2. Registration of accounting systems' users on API test environment

Step 1. Test participant authorization

After test participant authorization on API test environment it's required to get an "Authentication code" using the following API method:

```
POST http://api.stage.mdlp.crpt.ru/api/v1/auth
Content-Type: application/json;charset=UTF-8
{
  "client_id": "01db16f2-9a4e-4d9f-b5e8-c68f12566fd5",
  "client_secret": "9199fe04-42c3-4e81-83b5-120eb5f129f2",
  "user_id": "starter_resident_1",
  "auth_type": "PASSWORD"
}
```

Response:

```
{
  "code": "7386a68f-c1e5-42c6-8ed5-5b933017c66c"
}
```

client_secret, *client_id* here are test participant keys

The received code must be used for getting a session key (don't change password in the string *password*).

```
POST http://api.stage.mdlp.crpt.ru/api/v1/token
Content-Type: application/json;charset=UTF-8
```

```
{
  "code": "7386a68f-c1e5-42c6-8ed5-5b933017c66c",
  "password": "password"
}
```

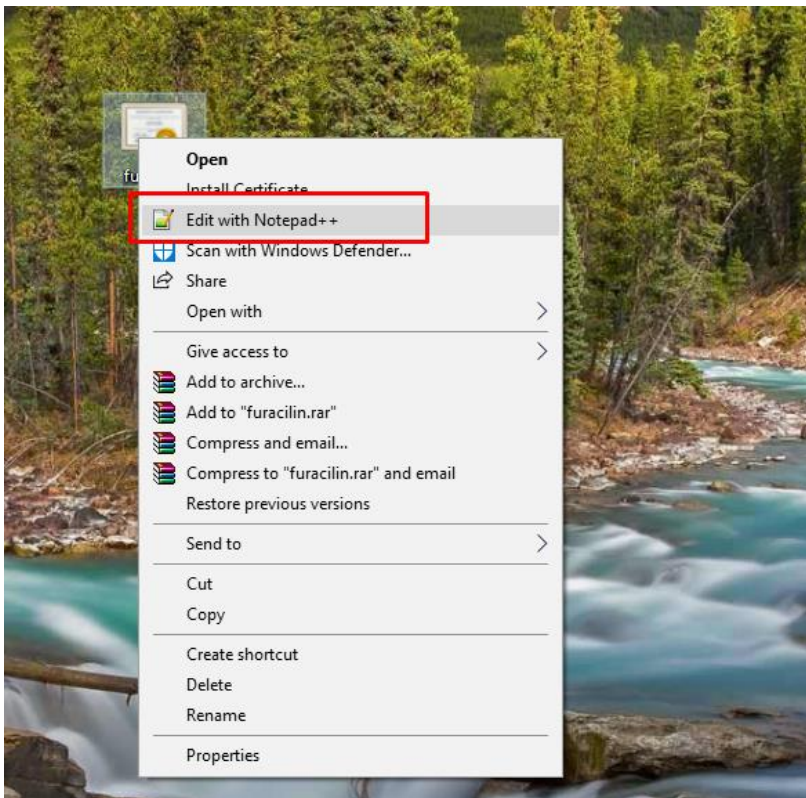
Response:

```
{
  "token": "13b5b046-0cd7-4e1c-8409-da9541986d1c",
  "life_time": 30
}
```

In response to the request you get a session key (token is always different) and its lifetime (life_time).

Step 2. Registration of an accounting system user – Russian Federation resident.

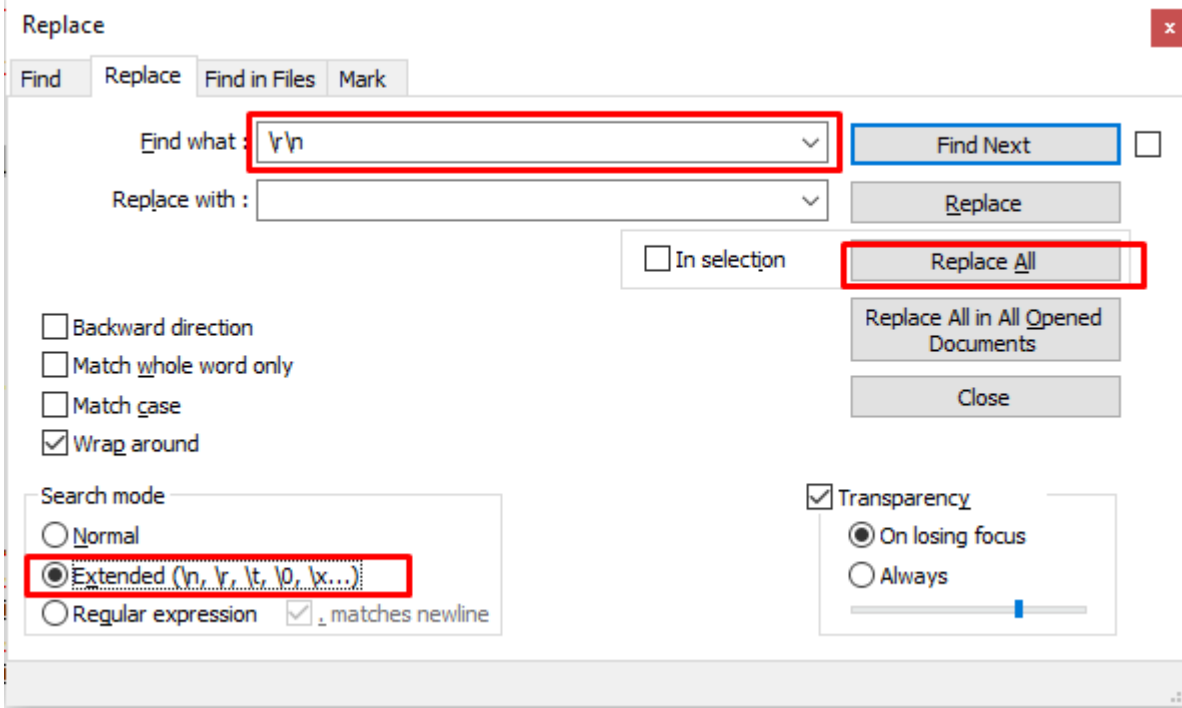
Open the certificate received from the step [Issue a test certificate of higher-level qualified e-signature for an accounting system user](#) that has already been converted into *Base64* format (as a sample the text editor *Notepad++* is used)



Remove a string header and a string end from an open certificate

```
1 -----BEGIN CERTIFICATE-----  
2 MIIITgDCCCFegAwIBAgIKNjBobQADAANG6zaIBgYqhQMCAGMwggFIMRgwFgYFKoUD  
3 ZAESDTEwMzc3MDAwODU0NDQxGjAYBgqhQMDgQMBARIMMDA3NzE3MTA3OTkxMTkw  
4 NwYDVQQJHjAEQwQ7AC4AIAQhBEMESQRRBDIEQQQ6BDgEQQAgBDIEMAQ7ACwATAQ0  
5 AC4AIAAxAdGxITAFBgNVBAgeGAA3ADcAIAQZAC4AIAQcBD4EQQQ6BDIEMDEVMBMG  
6 A1UEBx4MBBWEpGRBBDoEMgQwMSAwHgYJKoZIhvcNAQkBFhFpbmZvQGNyeXB0b3By  
7 by5ydTElMAkGALUEBhMCU1UxKTAnBgNVBAAeIAQeBB4EHgAgACIEGgQgBBgEHwQi  
8 BB4ALQQfBCAEHgAiMUEwPwYDVQQDHjgEIGQ1BEEEQgQ+BDIE.SwQ5ACAEIwQmACAE  
9 HgQeBB4AIAAIBBoEIAQYBB8EIGQeAC0EHwQgBB4AIjAeFw0xODA2MjYxMzZwMwMdBa  
10 Fw0xODA5MjYxMzZwMwMdBaMIIBADEYMBYGBSgFA2QBEG02MDY3NTAOMTAwMzcyMRo  
11 GAYIKoUDA4EDAQESDDAwODg0MjYxMzZwMwMdBaMIIBADEYMBYGBSgFA2QBEG02MDY3  
12 XN0LnJlMQswCQYDVQQGEwJVTEEmMCQGA1UECAw0KHQsNC90LzRmR3Q3Qm90LzRmR3Q3  
13 tdGAOLHRg9GAOLMxJjAkBgNVBACMHdCh0LDQvdC60Yi0J/QtdGC0LXRgNCx0YPR  
14 gNCzMSIwIAYDVQQKBNQkNcEICLQpNGD0YDQsNGG0LjQu9C40L0iMQowCAYDVQQQ  
15 DAEwMSIwIAYDVQQDBBNQkNcEICLQpNGD0YDQsNGG0LjQu9C40L0iMRswGQYDVQQM  
16 DBLQkdGD0YXQs9cW0LVRgtC10YaxLDAgBgNVBCoMI9CS0Y/Rh9C10YHQu9Cw0LIg  
17 0JHQvtGAOLjRgdC+0LLQuNGHMRcwFQYDVQED7QodC40LzQvtC90L7QsjCBqjAh  
18 BggqhQMHAQEBAjAVBgqhQMHAQIBAgEGCCqFAwCBAQIDA4GEAASBgE7eEQT9NK2J  
19 q/ikln7LlzRsNoqXQh77CbLwsPDLBc705vpUL+7F94W1ATdycQ61wC2wP9EFMESW  
20 qCbK39I1Tpv3G4cf8jiZg86AsLlnFgvOyAxIoR0/csBTnAnTKm9Ls+LDeEJQa8MQ  
21 yVbBpGz+HcH8BBD5BhisQoIvCpoAuIgro4IEszCCBK8wDgYDVR0PAQH/BAQDAgTw  
22 MCVCA1UAZGQEMCCGCGIABEBAQwIBAgEgGCqFAwCBAQIDA4GEAASBgE7eEQT9NK2J  
36 L29jc3AyL29jc3Auc3JmMDUGCCsGAQUFBzAChilodHRwOi8vd3d3LmNyeXB0b3By  
37 by5ydS9yYS9jZHVhY2FjZlZlZmNyeDARBgNVHRAEJDA1gA8yMDE4MDYyNjEzZmZw  
38 MFqBdzIwMTgwOTI2MTZMDAwWjAdBgNVHSAEFjAUMAgGBiqFA2RxAATAIBgYqhQk  
39 cQ1wNAYFKoUDZG8EKwpp0JzRgNC40L/RgtC+0J/RgNC+IENTUCAo0LLQtdGAOYH  
40 uNGPIDMuNikwggEzBgUqhQkASCASgwgGkDcs10JzRgNC40L/RgtC+0J/RgNC+  
41 IENTUCIgKNCy0LXRgNGBoLjRjyAzLjYpDFM10KPQeNC+0YHRgtC+0LLQtdGAOY/  
42 jtG70LjQuSDRhtC10L3RgtGAICLQmtGAOLjQv9GC0L7Qn9GA0L4g0KQpIiG0LLQ  
43 tdGAOYHQuNC4IDeUNQxP0KHQtdGA0YLQuNGE0LjQutCw0YIg0YHQtC+0YLQstC1  
44 0YLRgdGC0LLQuNGPIOKE1iDQodCkLzEYnCOyNz4INC+0YIgmDEuMDCuMjAxN0xP  
45 0KHQtGA0YLQuNGE0LjQutCw0YIg0YHQtC+0YLQstC10YLRgdGC0LLQuNGPIOKE  
46 1iDQodCkLzEYnCOyNz4INC+0YIgmDEuMDCuMjAxN0xP0KHQtGA0YLQuNGE0LjQ  
47 SpXSQdd1jNUjv1I6bsjjArLQ1eQRh6FpoKJKw/1RGhm8TwAGYR9YfjV4yGRgU8K  
48 T8Zwfjlp2ChZ2FA  
49 -----END CERTIFICATE-----  
50
```

Press **CTRL+H** in the text editor. Click on "Replace" to remove line breaks and line spacing. Put `\r\n` in "Find what", select "Extended" from "Search mode", click on "Replace all".



Certificate data is converted into a string.

```
MIIIIqDCCCFegAwIBAgIKNjBobQADAANG6zAIBgYqhQMCAGMwggFIMRgwFgYFKoUDZAESDTEwMzc3MDAwODUONDQxGjAYI
OTkxMTkwNwYDVQJHjAEQwQ7AC4AIAQhBEMESQRRBDIEQQQ6BDgEOQAQgBDIEMAQ7ACwAIAQ0AC4AIAAxADgxITAfBgNVI
QQQ6BDIEMDEVMBMGALUEBx4MBBwEPgRBBD0EMgQwMSAwHgYJKoZIHvcNAQkBFhFpbmZvQGNYeXB0b3Byby5ydTELMakGI
BB4EHgAgACIEGgQgBBgEHwQiBB4ALQQfBCAEHGAiMUEwPwYDVQDHjgEIGQ1BEEEQgQ+BDIESwQ5ACAEIwQmACAEHGAeI
HwQgBB4AIjAeFw0xODA2MjYxMzMwMDBaFw0xODA5MjYxMzQwMDBaMIIBADEYMBYGBSFA2QBEg02MDY3NTA0MTAwMzcyl
MjQxOTg4NzEhMBkGCSqGSIB3DQEJARYMdGVzdEB0ZXN0LnJlMQswCQYDVQGEwJSVTEmMCQGA1UECAw0KHQsNC90LrRgI
JjAkBgNVBAcMHdCh0LDQvdC60YIt0J/QtdGC0LXRgNCx0YPRgNCzMSIwIAYDVQQKBnQkNCEICLQpNGD0YDQsNGG0LjQ
IAYDVQQDBnQkNCEICLQpNGD0YDQsNGG0LjQu9C40L0iMRswGQYDVQQMDBLQkdGD0YXQs9Cw0LVRgtC10YAxLDAqBgNVB
0JHQvtGA0LjRgdC+0LLQuNGHMRcwFYDVQQEDA7QodC40LzQvtC90L7QsjCBqjAhBggqghQMHAQEBAjAVBgkqhQMHAQIBI
EQT9NK2Jq/ik1n7LlzRsNoqXQh77CbLwsPDLBc705vpUL+7F94WiAtdycQ61wC2wP9EFMESWqCbk39I1Tpv3G4cf8jiZ(
Km9Ls+LDcEJQa8MQyVbBPgZ+HcH8BBD5BhisQoIvCpoAuIgro4IESzCCBK8wDgYDVR0PAQH/BAQDAgTwMTCYGA1UdJQQfI
BggrBgEFBQcDAjAdBgNVHQ4EFgQUUNlh86WiQ4GFDt3qbAvSyQkxxv5IwggGJBgNVHSMEggGAMIIBfIAUK7IQNGaCAqzw
AUgxGDAWBgUqhQNkARINMTAZNzcwMDA4NTQ0NDEaMBGcCCqFAwOBAwEBEGwwMDc3MTcxMDc50TExOTA3BgNVBAkeMARDI
OAQ5ACAEMGQwBDsALAAgBDQALgAgADEAODEHMB8GALUECB4YADcANwAgBDMALgAgBBwEPgRBBD0EMgQwMRUwEwYDVQQHI
hkiG9w0BCQEWEluZm9AY3J5cHRvcHJvLnJlMQswCQYDVQGEwJSVTEpMCCGALUECh4gBB4EHgQeACAAIqQaBCAEGAQFI
BAMEOAQIBDUEQRCBD4EMgRLBDkAIAQjBCYAIAQeBB4EHgAgACIEGgQgBBgEHwQiBB4ALQQfBCAEHGAiGhBEH9pXSZNq
UaBPoE2GS2h0dHA6Ly93d3cuY3J5cHRvcHJvLnJlL2NkC8yYmIyMTAzNDY2ODIwMmFjZjZjB1MWFhNDA4Njc4MDE3I
BQUHAQEEdGgaEwNAyIKwYBBQUHMAGGKgH0dHA6Ly93d3cuY3J5cHRvcHJvLnJlL29jc3BuYzIvbnZncz5zcmYwMgYII
Y3J5cHRvcHJvLnJlL29jc3AyL29jc3Auc3JmMDUGCCsGAUUFBzAChilodHRwOi8vd3d3LmNyeXB0b3Byby5ydS9yYS9j
JDAigA8yMDE4MDYyNjEzZmZAwMFqBDzIwMTgwOTI2MTMzMdAAWjAdBgNVHSAEFjAUMAgBiqFA2RxATAIBgYqhQkCQIwI
gtC+0J/RgNC+IENTUCAo0LLQtdGA0YHQUNGPIDMuNikwggEzBgUqhQkCASCASgwgGkDCsi0JrRgNC40L/RgtC+0J/R
jyAzLjYpDFMi0KPQtNC+0YHRgtC+0LLQtdGA0Y/RjtGJ0LjQuSdRhtC10L3RgtGAICLQmtGA0LjQv9GC0L7Qn9GA0L4g
NQxP0KHQtdGA0YLQuNGE0LjQutCw0YIq0YHQvtC+0YLQstC10YLrGdGC0LLQuNGPIOKELiDQodCkLzEYn0C0yNz4INC+
0YLQuNGE0LjQutCw0YIq0YHQvtC+0YLQstC10YLrGdGC0LLQuNGPIOKELiDQodCkLzEYn0C0yNz4INC+0YIgmZEuMTIul
5pXSQddljNUjv1I6bsjjArLQleQRh6FpoKJKw/1RGhm8TwAGYR9Yfjv4uyGRqU8KT8ZwFj1bp2ChZ2FF
```

The received value will be used for a new user registration (*public_cert* field).

The method for registration and description of fields is described in a more detailed way in the documentation "Interface level communication protocol" paragraph 3.1.2. - the Method for registration of accounting system users (for residents).

```
POST http://api.stage.mdlp.crpt.ru/api/v1/registration/user_resident
Content-Type: application/json;charset=UTF-8
Authorization: token 13b5b046-0cd7-4e1c-8409-da9541986d1c
```

```
{
  "sys_id": "9dedee17-e43a-47f1-910e-3a88ff6bc81b",
  "public_cert": "
```

```
MIIIIqDCCCFegAwIBAgIKNjBobQADAANG6zAIBgYqhQMCAGMwggFIMRgwFgYFKoUDZAES
DTEwMzc3MDAwODUONDQxGjAYBggqghQMDgQMBARIMMDA3NzE3MTA3OTkxMTkwNwYDVQJH
HjAEQwQ7AC4AIAQhBEMESQRRBDIEQQQ6BDgEOQAQgBDIEMAQ7ACwAIAQ0AC4AIAAxADgx
ITAfBgNVBAgeGAA3ADcAIAQzAC4AIAQcBD4EQQQ6BDIEMDEVMBMGALUEBx4MBBwEPgRB
BD0EMgQwMSAwHgYJKoZIHvcNAQkBFhFpbmZvQGNYeXB0b3Byby5ydTELMakGALUEBhMC
U1UxKTAhBgNVBAoeIAQeBB4EHgAgACIEGgQgBBgEHwQiBB4ALQQfBCAEHGAiMUEwPwYD
VQDHjgEIGQ1BEEEQgQ+BDIESwQ5ACAEIwQmACAEHGAeBB4AIAAIBBoEIAQYBB8EIGQe
AC0EHwQgBB4AIjAeFw0xODA2MjYxMzMwMDBaFw0xODA5MjYxMzQwMDBaMIIBADEYMBY
BSqFA2QBEg02MDY3NTA0MTAwMzcylMR0wGAYIKoUDA4EDAQESDDAwODg0MjQxOTg4NzE
hMBkGCSqGSIB3DQEJARYMdGVzdEB0ZXN0LnJlMQswCQYDVQGEwJSVTEmMCQGA1UECAw
0KHQsNC90LrRgi3Qn9C10YLQtdGA0LHRg9GA0LMxJjAkBgNVBAcMHdCh0LDQvdC60YIt
0J/QtdGC0LXRgNCx0YPRgNCzMSIwIAYDVQQKBnQkNCEICLQpNGD0YDQsNGG0LjQu9C4
0L0iMQowCAYDVQQQLDAEwMSIwIAYDVQQDBnQkNCEICLQpNGD0YDQsNGG0LjQu9C40L0i
MRswGQYDVQQMDBLQkdGD0YXQs9Cw0LVRgtC10YAxLDAqBgNVBComI9CS0Y/Rh9C10YHQ
u9Cw0LIg0JHQvtGA0LjRgdC+0LLQuNGHMRcwFYDVQQEDA7QodC40LzQvtC90L7QsjCB
qjAhBggqghQMHAQEBAjAVBgkqhQMHAQIBAgEGCCqFAwCBAQIDA4GEEAASBgE7eEQT9NK2J
q/ik1n7LlzRsNoqXQh77CbLwsPDLBc705vpUL+7F94WiAtdycQ61wC2wP9EFMESWqCbk
39I1Tpv3G4cf8jiZg86AsLLnFgvOyAxIoR0/csBTnAntK9Ls+LDcEJQa8MQyVbBPgZ+
HcH8BBD5BhisQoIvCpoAuIgro4IESzCCBK8wDgYDVR0PAQH/BAQDAgTwMTCYGA1UdJQQf
MB0GCCsGAUUFBwMEBgcqhQMCAiIGBggrBgEFBQcDAjAdBgNVHQ4EFgQUUNlh86WiQ4GFD
t3qbAvSyQkxxv5IwggGJBgNVHSMEggGAMIIBfIAUK7IQNGaCAqzw4apAhngBcUWdM+Oh
ggFQpIIBTDCCAUGxGDAWBgUqhQNkARINMTAZNzcwMDA4NTQ0NDEaMBGcCCqFAwOBAwEB
EGwwMDc3MTcxMDc50TExOTA3BgNVBAkeMARDBDsALgAgBCEEQwRJBFEEMgRBBD0EOAQ5
```


The Method for creating a group of user rights is described in a more detailed way in the documentation “Interface level communication protocol” paragraph 5.6.3.

The full list of accounting system user rights for the field “rights” is described in the documentation “Interface level communication protocol” paragraph 3.10.

Step 4. Adding a user to the group of rights

```
POST http://api.stage.mdlp.crpt.ru/api/v1/rights/a5d134fe-43cb-42a1-85eb-61a5cfcffef9/user_add
Content-Type: application/json;charset=UTF-8
Authorization: token 13b5b046-0cd7-4e1c-8409-da9541986d1c

{
    "user_id": "e0509105-78b5-419f-b0f1-7edc86434f6d"
}
```

Response:

```
HTTP 200
```

Add user identifier (received from **Step 2**) to the request body string “user_id”.

Add user group identifier (received from **Step 3**) to request URI.

Thus, as shown by the sample, a user will be added to the group of rights called “Test group” and will have the following rights: “Accounting systems control” and “Viewing accounts”.

That concludes the process of creating a native test user of an accounting system.

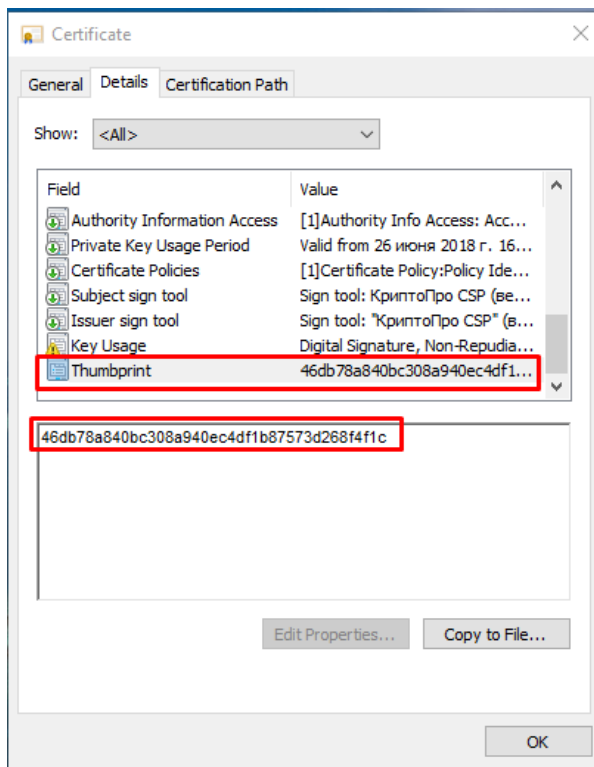
For creating the second user you need to repeat these steps, it will allow you to have an opportunity for testing interaction between two participants.

Authentication of a created user by the higher-level qualified e-signature certificate.

Thus, we’ve registered a user that belongs to a participant – resident of the Russian Federation. For further work a created user is required. Test users **starter_resident_1** and **starter_resident_2** are no longer necessary. Created users belong to the same accounting system as test users **starter_resident_1** and **starter_resident_2**, respectively.

Authorization

You can learn about your certificate thumbprint in the following way:



GUIDs from this sample can be used for *client_secret* and *client_id* parameters;

For *user_id* parameter it's possible to use either a certificate thumbprint (see the screenshot) or a certificate decimal serial number. **Don't try to use the thumbprint from the sample.**

```
POST http://api.stage.mdlp.crpt.ru/api/v1/auth
Content-Type: application/json;charset=UTF-8
{
  "client_secret": "9199fe04-42c3-4e81-83b5-120eb5f129f2",
  "client_id": "01db16f2-9a4e-4d9f-b5e8-c68f12566fd5",
  "user_id": "46db78a840bc308a940ec4df1b87573d268f4f1c",
  "auth_type": "SIGNED_CODE"
}
```

Response:

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

The Method for getting an authentication code is described in a more detailed way in the documentation "Interface level communication protocol" paragraph 5.2.2.

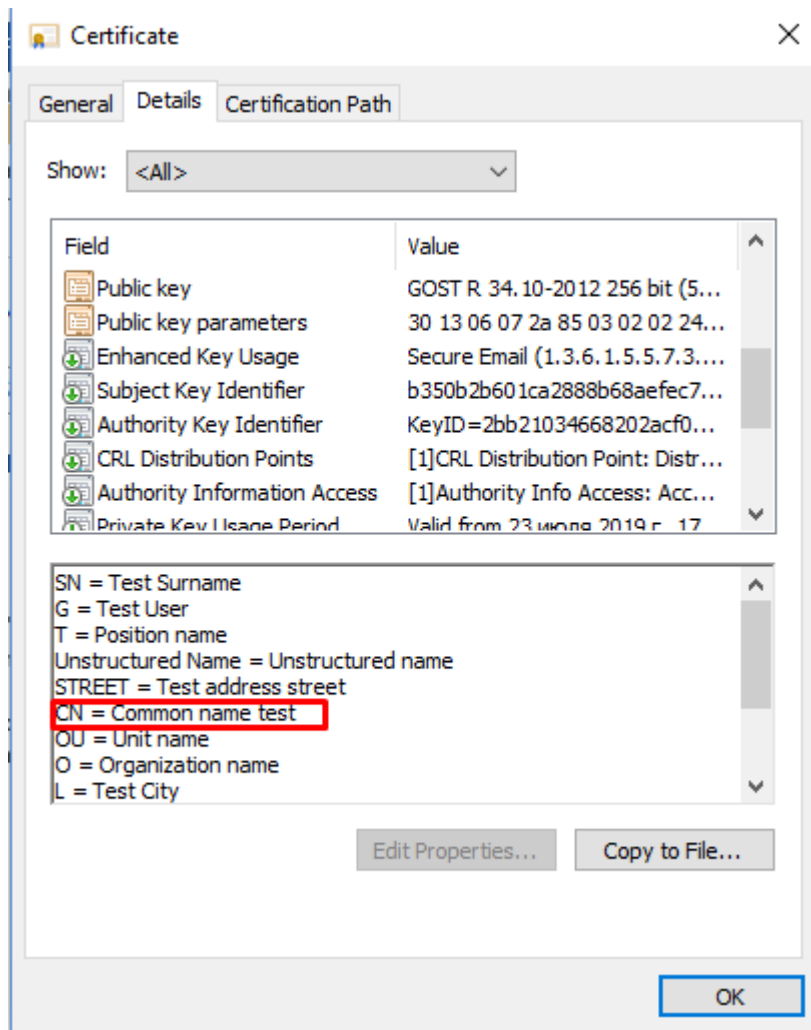
Signing a received code

The method for getting an e-signature with the help of CSP Crypto Pro version 4.0 is shown in the following sample:

1. Create a file with *txt* extension.
2. Add the code received in the authentication method to the text file. Note that there should not be any extra characters (spaces, line breaks, tabs etc).

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

3. Sign the text file with a higher-level qualified e-signature:
 - a. Check the CN of your certificate. If there are quotation marks in the certificate name, don't use them while entering the data.



- b. Open a `cmd` console in the folder with installed CSP Crypto Pro software. (The default path: `\Program Files\Crypto Pro\CSP`).
- c. Enter the following command in the console:

```
csptest -sfsign -sign -in <FULL PATH TO THE FILE>test.txt -out <FULL PATH TO THE FILE>out.txt -my "Common name test" -detached -base64 -add
```

As shown by the sample at the output you get an `out.txt` file with an e-signature. **Remove a string header and string end characters from the e-signature.** Then put the e-signature in "signature" string of the `/api/v1/token` method.

Getting a session key for a higher-level qualified e-signature

The method for getting a session key

```
POST http://api.stage.mdlp.crpt.ru/api/v1/token
```

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

```
{
  "code": "b74d04c3-e784-421e-84f9-dde0fc1bb3c9",
  "signature":
    "MIAGCSqGS Ib3DQEHAqCAMIACAQEExDDAKBgYqhQMCAgkFADCABgkqhkiG9w0BBwEAAKCAMIIBjjCCAT2gAwIBAgIEWWJzHzAIBgYqhQMCAgMwMTELMakGA1UEBhMCU1UxEjAQBgNVBAoMCUNyeXB0b1BybzEOMAwGA1UEAwwFQWxpYXNwYXN0b1BhcnVzMTczMjI4WhcNMTg0MTEzMTczMjI4WjAxMQswCQYDVQGEwJSVTESMBAGA1UECgwJQ3J5cHRvUHJvMQ4wDAYDVQQDDAVBbG1hc2BjMBwGBiQFAwICEzASBgqhQMCAiQABgcqhQMAh4BA0MABEAI
```

```
WARzAiI81k4i4Gz8EC7Ic01653JX5PCUfvgCBTpLduYtbTwLOwmGFcZzw9bwsxQpALqh
cdRHxtx1UEeNKJuMozswOTAObgNVHQ8BAf8EBAMCA+gwEwYDVR01BAwwCgYIKwYBBQUH
AwIwEgYDVR0TAQH/BAgwBgEB/wIBBTAIBgYqhQMCAgMDQQL9CrIk0EgnMVr1J5dKbfX
VFrhJxGxztFkTdmGkGJ6gHywB5Y9KpP67pv7I2bP1m1ej9hu+C17GSJrWgMgq+UZAAAx
ggEFMIIBAQBATA5MDExCzAJBgNVBAYTA1JVMRIwEAYDVQQKDA1DcnlwdG9Qcm8xDjAM
BgNVBAMMBUFsaWFzAgRZYnMfMAoGBiqFAwICCQUAoGkwGAYJKoZIhvcNAQkDMQsGCSqG
SIb3DQEHAQAcBgkqhkiG9w0BCQUxDxcNMTcxMTMwMDkyMTU5WjAvBgkqhkiG9w0BCQQx
IqQg+vo77XxU6eHIpoZxOwIk+gWY+JSv91fa1TgOM/UW8aowCAYGKoUDAgIDBECJhWsl
OQ1Wn4KntnGNL8KNZSpbpJ35E8NjyEio+Im5dLiLoqZotxGF1WMmYvBaPy8FHcAsTqaE
EJna7+CecOi/AAAAAAA"
```

```
}
```

Response:

```
{
```

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

```
  "life_time": 30
```

```
}
```

Don't try to use the e-signature from the sample.

The procedure of authentication and authorization using a higher-level qualified e-signature is finished.

[Signing and sending schemes with a higher-level qualified e-signature](#)

Note: After authentication and authorization procedures all API methods must be called using HTTPS protocol. For https interaction the certificate GOST R 34.10-2012 is used.

(Comment: for sending schemes a user must have the rights UPLOAD_DOCUMENT)

For sending a document the following API method is used: /documents/send

Scheme samples and formats are described on the website

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

Look at the sample of sending a document:

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

It's extremely important to use UTF-8 encoding. Make sure that the characters of spaces, line breaks (line breaks characters in Unix and Windows are different) are the same in the request body that is going to be signed and the one that is going to be sent. Otherwise while checking the signature the result will be negative. The signature must be identical to the one that is sent to the field *document*.

[Signing a document](#)

The following sample demonstrates the method of getting an e-signature using CSP Crypto Pro version 4.0.

1. Create a file with txt extension. E.g. **doc.txt**
2. Insert document body into the text file

```

1 <documents xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.19">CRLE
2   <register_end_packing action_id="311">CRLE
3     <subject_id>00000000000517</subject_id>CRLE
4     <operation_date>2018-08-22T15:00:00+05:00</operation_date>CRLE
5     <order_type>1</order_type>CRLE
6     <series_number>100000001</series_number>CRLE
7     <expiration_date>30.03.2020</expiration_date>CRLE
8     <gtin>11170012610151</gtin>CRLE
9     <tnved_code>3004</tnved_code>CRLE
10    <signs>CRLE
11      <sgtin>07091900400001TRANSF2000021</sgtin>CRLE
12    </signs>CRLE
13  </register_end_packing>CRLE
14 </documents>

```

3. Sign the text file with a higher-level qualified e-signature

Signature sample using Crypto Pro CSP version 4.0.

Open a *cmd* console in the folder with installed CSP Crypto Pro software. (The default path:\Program Files\Crypto Pro\CSP)

Enter the following command in the console:

```

csptest -sfsign -sign -in <FULL PATH TO THE FILE>doc.txt -out
<FULL PATH TO THE FILE>signed_doc.txt -my "Common name test" -
detached -base64 -add

```

Comment: if there are quotation marks in the certificate name, don't use them while entering the data.

Comment: specify the certificate that was used for an authentication procedure.

As shown by the sample at the output you get a *signed_doc.txt* file with an e-signature. Remove line wrap characters from the e-signature.

Use the e-signature for sending a document (see the next step below).

Convert a document body into Base64

```

PGRvY3VtZW50cyB4bWxuczp4c2k9Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvWE1MU2No
ZW1hLWluc3RhbmNlIiB2ZXJzaW9uPSIxLjE5Ij48cmVnaXN0ZXJfZW5kX3BhY2tpbmcg
YWN0aW9uX2lkPSIzMTEiPjxzZWJqZW50X2lkPjAwMDAwMDAwMDAwNTE3PC9zdWJqZW50
X2lkPg==

```

Send a document

Generate a new identifier in UUID format (for *request_id* parameter)

```
POST https://api.stage.mdlp.crpt.ru/api/v1/documents/send
```

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

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

```

{
  "document":
  "PGRvY3VtZW50cyB4bWxuczp4c2k9Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvWE1MU2No

```

oZW1hLWluc3RhbmNlIiB2ZXJzaW9uPSIxLjE5Ij48cmVnaXN0ZXJfZW5kX3BhY2tpbmc
gYWN0aW9uX2lkPSIzMTEiPjxzdwJqZWN0X2lkPjAwMDAwMDAwMDAwNTE3PC9zdWJqZWN
0X2lkPg==",

"sign": "

MIINHwYJKoZiHvcNAQcCoIINEDCCDQwCAQExDjAMBggqhQMHAQECAGUAMAsGCSqGSib3
DQEHAaCCCQMwggj/MIIIrqADAgECAGpKStMAAAMAA0TkMAgGBiqFAwICAzCCAUGxGDAW
BgUqhQNkARINMTAzNzcwMDA4NTQ0NDEaMBGqCCqFAwOBAwEBEGwwMDc3MTcxMDc5OTEx
OTA3BgNVBAkeMARDBDsALgAgBCEEQwRJBFEEMgRBBDoEOAQ5ACAEMgQwBDsALAAgBDQA
LgAgADEAODEhM8GA1UECB4YADcANwAgBDMALgAgBBwEPgRBBDoEMgQwMRUwEwYDVQQH
HgwEHAQ+BEEOgQyBDAXIDAeBgkqhkiG9w0BCQEWEluZm9AY3J5CHRvcHJvLnJlMQsw
CQYDVQQGEwJSVTEpMCcGA1UECh4gBB4EHgQeACAAIqQaBCAEGAQfBCIEHgAtBB8EIAQe
ACIXQTA/BgNVBAMeOAQiBDUEQQRCBD4EMgRLBDkAIAQjBCYAIAQeBB4EHgAgACIEGgQg
BBgEHwQiBB4ALQQfBCAEGhAiMB4XDTE4MDYwNTE0NDQwMFoXDTE4MDkwNTE0NTQwMFow
ggIEMRgwFgYFKoUDZAESDTAwMDAwMDAwMDAwMDMxGjAYBggqhQMDgQMBARIMMDAwMDAw
MDAwMDAwMR4wHAYJKoZiHvcNAQkBFg9leGftcGx1QG1haWwucnUxCzAJBgNVBAYTALJV
MSYwJAYDVQQIDB3QodCw0L3QutGCLdCf0LXRgtC10YDQsdGD0YDQsZEmMCQGA1UEBwwd
0KHQsNC90LrRgi3Qn9C10YLQtdGA0LHRg9GA0LMxNDAyBgNVBAoMK9Ci0LXQv9C70L7Q
v9GA0LjQsdC+0YDRgdC10YDQstC40YEg0YLQtdGB0YLxjAIBgNVBAsMATEXPzA9BgNV
BAMNtCe0YHQUNc/0L7QsiDQkNC70YzQsdC10YDRgiDQkNC70LXQutGB0LDQvdc00YDQ
vtCy0LjRhZEuMCwGA1UECQw10YLQtdGB0YLQvtCy0YvQuSDQsNC00YDQtdGBINC00L7Q
vCA0NDEfMB0GCSqGSib3DQEJAgwQ0JTQuNGA0LXQutGC0L7RgDEwMC4GA1UEDAwn0JPQ
tdC90LXRgNCw0LvRjNC90YvQuSDQtNC40YDQtdC60YLQvtGAMTIwMAYDVQQqDcnQkNC7
0YzQsdC10YDRgiDQkNC70LXQutGB0LDQvdc00YDQvtCy0LjRhZEVMBMGA1UEBAwM0J7R
gdC40L/QvtCyMGYwHwYIKoUDBwEBAQEwEwYHKoUDAgIkaAYIKoUDBwEBAgIDQwAEQPXR
befMOOOIm39XX4hNPMdn1v9f4Ckf5QMOSBvcfTHiLx2uZD5ia11+GniVA1oW13A+vwsh
xpsiGoQoJc2vn1+jggSzMIIErzaOBgNVHQ8BAf8EBAMCBPAwJgYDVR0LBB8wHQYIKwYB
BQUHAWQByqFAwICIGYCCsGAQUFBwMCMB0GA1UdDgQWBRRCH97RoSiTjciGebTvQbQk
CEhwYDCCAYkGA1UdIwSCAYAwggF8gBQRshA0ZoICrPDhqcCGeAFxRZ0z46GCAVCkggFM
MIIBSDEYMBYGBSFA2QBEg0xMDM3NzAwMDg1NDQ0MR0wGAYIKoUDA4EDAQESDDAwNzcX
NzEwNzk5MTE5MDcGA1UECR4wBEMEOwAuACAEIQRDBEkeEUQQyBEEEOgQ4BDkAIAQyBDAE
OwAsACAENAAuACAAMQA4MSEwHwYDVQQIHHgANwA3ACAEMwAuACAEHAQ+BEEOgQyBDAX
FTATBgNVBAceDAQcBD4EQQQ6BDIEMDEgMB4GCSqGSib3DQEJARYRaW5mb0BjcnlwdG9w
cm8ucnUxCzAJBgNVBAYTALJVMskwJwYDVQQKHIAEHgQeBB4AIAAIBBoEIAQYBB8EIGQe
AC0EHwQgBB4AIjFBMD8GA1UEAx44BCIENQRBBEIEPqQyBEsEOQAqBCMEJgAgBB4EHgQe
ACAAIqQaBCAEGAQfBCIEHgAtBB8EIAQeACKCEESH21dJk2CeSHb2gnNE/xcwXAYDVR0f
BFUwUzBR0E+gTYZLaHR0cDovL3d3dy5jcnlwdG9wcm8ucnUvcnUvY2RwLzJiYjIxMDM0
NjY4MjAyYWNmMGUxYWE0MDg2NzgwMTcxNDU5ZDMzZTMuY3JSMIGxGggrBgEFBQcBAQSB
pDCBoTA0BggrBgEFBQcwAYYoHR0cDovL3d3dy5jcnlwdG9wcm8ucnUvbn2NzcG5jMi9v
Y3NwLnNyZjAyBggrBgEFBQcwAYYmaHR0cDovL3d3dy5jcnlwdG9wcm8ucnUvbn2NzcDIv
b2NzcC5zcmYwNQYIKwYBBQUHMAKkWh0dHA6Ly93d3cuY3J5CHRvcHJvLnJlL3JhL2Nk
cC9jYWNlcjMuY3J0MCSGA1UdEAQkMCKADzIwMTgwnjA1MTQ0NDAwWoEPMjAxODA5MDUx
NDQ0MDBaMB0GA1UdIAQWMBQwCAYGKoUDZHEBMAgGBiqFA2RxAJA0BgUqhQNkbwQrDCnQ
mtGA0LjQv9GC0L7Qn9GA0L4gQ1NQICjQstC10YDRgdC40Y8gMy42KTCCATMGBSFA2Rw
BIIBKDCASQMkYLQmtGA0LjQv9GC0L7Qn9GA0L4gQ1NQIiAo0LLQtdGA0YHQUNGPIDMu
NikMUyLQo9C00L7RgdGC0L7QstC10YDRj9G00YnQuNC5INGG0LXQvdcGC0YAgItCa0YDQ
uNC/0YLQvtCf0YDQviDQo9CmIiDQstC10YDRgdC40LggMS41DE/QodC10YDRgtC40YTQ
uNC60LDRgiDRgdC+0L7RgtCy0LXRgtGB0YLQstC40Y8g4oSWINCh0KQvMTI0LTI3Mzgg
0L7RgiAwMS4wNy4yMDE1DE/QodC10YDRgtC40YTQuNC60LDRgiDRgdC+0L7RgtCy0LXR
gtGB0YLQstC40Y8g4oSWINCh0KQvMTI4LTI3Njgg0L7RgiAzMS4xMi4yMDE1MAGGBiqF
AwICAwNBA0OgYI1/8v4f0Di2iH0Q1AP7X2U9DnSfpy+IZABpBI6/8IksnT/JBmRgBe6u
uF3wTewkgD8YKJkUqXbbly1Rd+0xggPhMIID3QIBATCCAAGwggFIMRgwFgYFKoUDZAES
DTEwMzc3MDAwODU0NDQxGjAYBggqhQMDgQMBARIMMDA3Nze3MTA30TkxMTkwNwYDVQQJ
HjAEQwQ7AC4AIAQhBEMESQRRBDIEQQQ6BDgEQQAqBDIEMAQ7ACwAIAQ0AC4AIAAaxADgx
ITAFBgNVBAgeGAA3ADcAIAQzAC4AIAQcBD4EQQQ6BDIEMDEVMBMGA1UEBx4MBBwEPgRB
BDoEMgQwMSAwHgYJKoZiHvcNAQkBFhFpbmZvQGNYeXB0b3Byby5ydTELMakGA1UEBhMC
ULUxKTAnBgNVBAoeIAQeBB4EHgAgACIEGgQgBBgEHwQiBB4ALQQfBCAEGhAiMUEwPwYD
VQODHjgEIGQ1BEEEOgQ+BDIESwQ5ACAEIwQmACAEGhQeBB4AIAAIBBoEIAQYBB8EIGQe

```
AC0EHwQgBB4AIgIKSkrtAAADAANE5DAMBggqhQMHAQECAGUAoIICHDAYBqkqhkiG9w0B
CQMxCwYJKoZIHvcNAQcBMBwGCSqGSIB3DQEJBTEPFw0xODA4MjkxNDE1MjRaMC8GCSqG
SIB3DQEJBDEiBCB9An2pcj1/MOfuHX8cJgNNzDp7wG+DWLhkqIeeSY04fjCCAa8GCyqG
SIB3DQEEJEAIVMYIBnjCCAZowggGMMIIBkjAKBggqhQMHAQECAGQg2kW4PyhwpD0j1ZI9
AHkdyAY1j6oDX6EfrMufeO/L/fYwggFgMIIBUKSCAUwwggFIMRgwFgYFKoUDZAESDTEw
Mzc3MDAwODU0NDQxGjAYBggqhQMDgQMBARIMMDA3NzE3MTA3OTkxMTkwNwYDVQQJHjAE
QwQ7AC4AIAQhBEMESQRRBDIEQQQ6BDgEQQAgBDIEMAQ7ACwAIAQ0AC4AIAAxADgxITAf
BgNVBAgeGAA3ADcAIAQzAC4AIAQcBD4EQQQ6BDIEMDEVMBMGA1UEBx4MBBwEPgRBBDoE
MgQwMSAwHgYJKoZIHvcNAQkBFhFpbmZvQGNyeXB0b3Byby5ydTELMakGA1UEBhMCU1Ux
KTANBgNVBAoeIAQeBB4EHgAgACIEGgQgBBgEHwQiBB4ALQQfBCAEHGAiMUEwPwYDVQQD
HjgEIgQ1BEEEQgQ+BDIESwQ5ACAEIwQmACAEHGAiBB4AIAAiBBoEIAQYBB8EIGQeAC0E
HwQgBB4AIgIKSkrtAAADAANE5DAMBggqhQMHAQEBAQUABECGAwmGd26uEblqctxuxmy6
K8+WAX148Bksn0ENrxm8Zbb8pAT2yU8IDARWmPIUAMGPEnn85anoRasZJqpyYI4y" ,
```

```
"request_id " : "2ba0919b-6dbc-4729-b886-df19dd60566d"
```

(request_id: must be always unique)

```
}
```

Response:

```
{
  "document_id": "cc7890a8-39ac-4ba9-ae9a-b20f406a781c"
}
```

Don't use the data from this sample.

Get a link for viewing a document

```
GET
https://api.stage.mdlp.crpt.ru/api/v1/documents/download/{document_id}
Content-Type: application/json;charset=UTF-8
Authorization: token fd780e64-b753-45c0-a375-31218210eadf
```

Response

```
{
  "link": "https://api.stage.mdlp.crpt.ru/webdav/upload/cc7890a8-39ac-4ba9-ae9a-b20f406a781c/cc7890a8-39ac-4ba9-ae9a-b20f406a781c"
}
```

Download a document

```
GET https://api.stage.mdlp.crpt.ru/webdav/upload/cc7890a8-39ac-4ba9-ae9a-b20f406a781c/cc7890a8-39ac-4ba9-ae9a-b20f406a781c
Authorization: token fd780e64-b753-45c0-a375-31218210eadf
```

Get a link for a receipt

```
GET https://api.stage.mdlp.crpt.ru/api/v1/documents/040fb1c3-69e4-43f7-b656-259a742eb944/ticket
Authorization: token d780e64-b753-45c0-a375-31218210eadf
```

Response

```
{
```



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

Download a receipt

```
GET https://api.stage.mdlp.crpt.ru/webdav/upload/040fb1c3-69e4-43f7-b656-259a742eb944/ticket_040fb1c3-69e4-43f7-b656-259a742eb944
Authorization: token d780e64-b753-45c0-a375-31218210eadf
```