01. Работа с E-IMZO

E-IMZO

Для корректной интеграции необходимо установить и запустить приложение E-IMZO (https://e-imzo.uz/)
При установленном E-IMZO доступна документация по адресу
https://127.0.0.1:64443/apidoc.html

1. Получение списка доступных ключей на локальном диске

Для получения списка доступных ключей на локальном диске необходимо:

1. Установить соединение к веб-сокету wss://127.0.0.1:64443/service/cryptapi

Headers:

```
Host:127.0.0.1:64443
Origin:https://{your web site}
```

2. В установленном соединении отправить сообщение:

```
{
    "plugin":"pfx",
    "name":"list_all_certificates"
}
```

Получаем ответ:

```
{
    "certificates": [
        {
            "disk": "C:\\",
            "path": "",
            "name": "DS242141224",
            "alias": "cn=fio,name=имя,surname=фамилия,l=район,st=город,c=uz,o=
        }
    ],
```

```
    "success": true
}
```

2. Получение keyId

Для получения ID ключа необходимо:

1. Установить соединение к веб-сокету wss://127.0.0.1:64443/service/cryptapi
   Headers:

```
Host:127.0.0.1:64443
Origin:https://{your web site}
```

2. Получить список ключей (Первый пункт на странице)
3. Отправить сообщение, в arguments передавать массив из значений ключа, полученные в списке ключей

```
{
    "plugin": "pfx",
    "name": "load_key",
    "arguments": [
        "C:\\", // значение disk
        "", // значение path
        "DS242141224", // значение name
        "cn=fio,name=имя,surname=фамилия,l=район,st=город,c=uz,o=название комп
    ]
}
```

Получаем ответ:

```
{
    "keyId": "4523456aec67ds568234f9500d151222",
    "type": "PFX_KEY_STORE",
    "success": true
}
```

3. Создание подписи

Для создания подписи необходимо:

1. Установить соединение к веб-сокету wss://127.0.0.1:64443/service/cryptapi
   Headers:

```
Host:127.0.0.1:64443
Origin:https://{your web site}
```

2. Получить keyId (Второй пункт на странице)
3. Отправить сообщение, в arguments передавать массив значений:
   a. Данные для подписи в base64
   b. keyId // полученный во втором шаге
   c. "no"

```
{
    "plugin": "pkcs7",
    "name": "create_pkcs7",
    "arguments": [
        "MjA3MTE5OTYz", // JSON документа или инн в base64
        "4523456aec67ds568234f9500d151222", // keyId
        "no"
    ]
}
```

Получаем ответ:

```
{
    "pkcs7_64": "MIAGCSqGSIb3DQEHAqCAMI...",
    "signer_serial_number": "7283d8ca",
    "signature_hex": "5b1433b23cddfd877d4a6ef4d7715d4f98c39d90c16d12340f23dbef
    "success": true
}
```

4. Прикрепление timestamp к подписи

Чтобы использовать API необходимо получить партнерский токен
Тестовая URL https://stage.goodsign.biz/
Прод партнерский URL https://api-partners.didox.uz/

Для получения партнеского токена нужно обратиться к одному из специалистов:
https://t.me/anomaluna - Шахноза
https://t.me/lightmoon5227 - Ёркиной

1. Создать подпись (Третий пункт на странице)
2. В полученном ответе взять значения pkcs7_64 и signature_hex

3. Отправить значения в /v1/dsvs/timestamp
   Body:

```
{
  "pkcs7": pkcs7_64 // Значение поля pkcs7_64
  "signatureHex": signature_hex // Значение поля signature_hex
}
```

Получаем ответ:

```
{
    "timeStampTokenB64": "MIAGCSqGSIb3DQEHAqEQG5ojtL3W8Ir4Qc17fInWcTykUVTxTYNH
    "success": true,
    "isAttachedPkcs7": true
}
```

Шаги:

1. Логин под компанией по ЭЦП

1. Получить список ключей
2. Получить keyId
3. Создать подпись
   Первым аргументов передать инн в base64
4. Прикрепить timestamp к подписи
5. Получить значение timeStampTokenB64 с респонса 4го шага
6. Отправить значение timeStampTokenB64 на эндпоинт POST /v1/auth/ИНН_КОМПАНИИ/token/ru
   Body:

```
{
    "signature": timeStampTokenB64 // значение с 5го шага
}
```

Пример получение данных:

Response : *200*

```
{
    "token": "87138df4-9426-49d7-a409-3ed986c49bb5",
    "related_companies": null,

```

```
    "related_branches": null
}
```

2. Подписание исходящего документа

1. Получить список ключей
2. Получить keyId
3. Получить значение data->json с респонса GET /v1/documents/айди_документа?owner=1
4. Преобразовать data->json с 3го шага в base64
5. Создать подпись
   Первым аргументов передать base64 с 4го шага
6. Прикрепить timestamp к подписи
7. Получить значение timeStampTokenB64 с респонса 6го шага
8. Отправить значение timeStampTokenB64 на эндпоинт POST /v1/documents/айди_документа/sign
   Body:

```
{
    "signature": timeStampTokenB64 // значение с 7го шага
}
```

Пример получение данных:

Response : *200*

Copy

```
{
    "data": true
}
```