

**Описание работы Сервиса BankID
для аутентификации гражданина
и получение информации о нем**

Публичная оферта

Согласно [оферты](#), принципа работы аутентификации клиента через BankID, которому все доверяют.

Задача данного портала, предоставить информацию о клиенте или подтвердить его подлинность (клиента).

Самое главное, чтоб эта информация была достоверной и не скомпрометированной.

Логистически работа BankID построена на договорных началах.

С одной стороны будут заключены соглашения с банками и юридическими организациями о предоставлении клиентской информации.

С другой стороны будут заключены соглашения с внешними сайтами Агентами о передаче им интересующей клиентской информации.

Авторизация внешних Агентов реализована на базе протокола OAuth2.0.

Аутентификация клиента проходит средствами предоставленными интернет-банком представляющий интересы Банка.

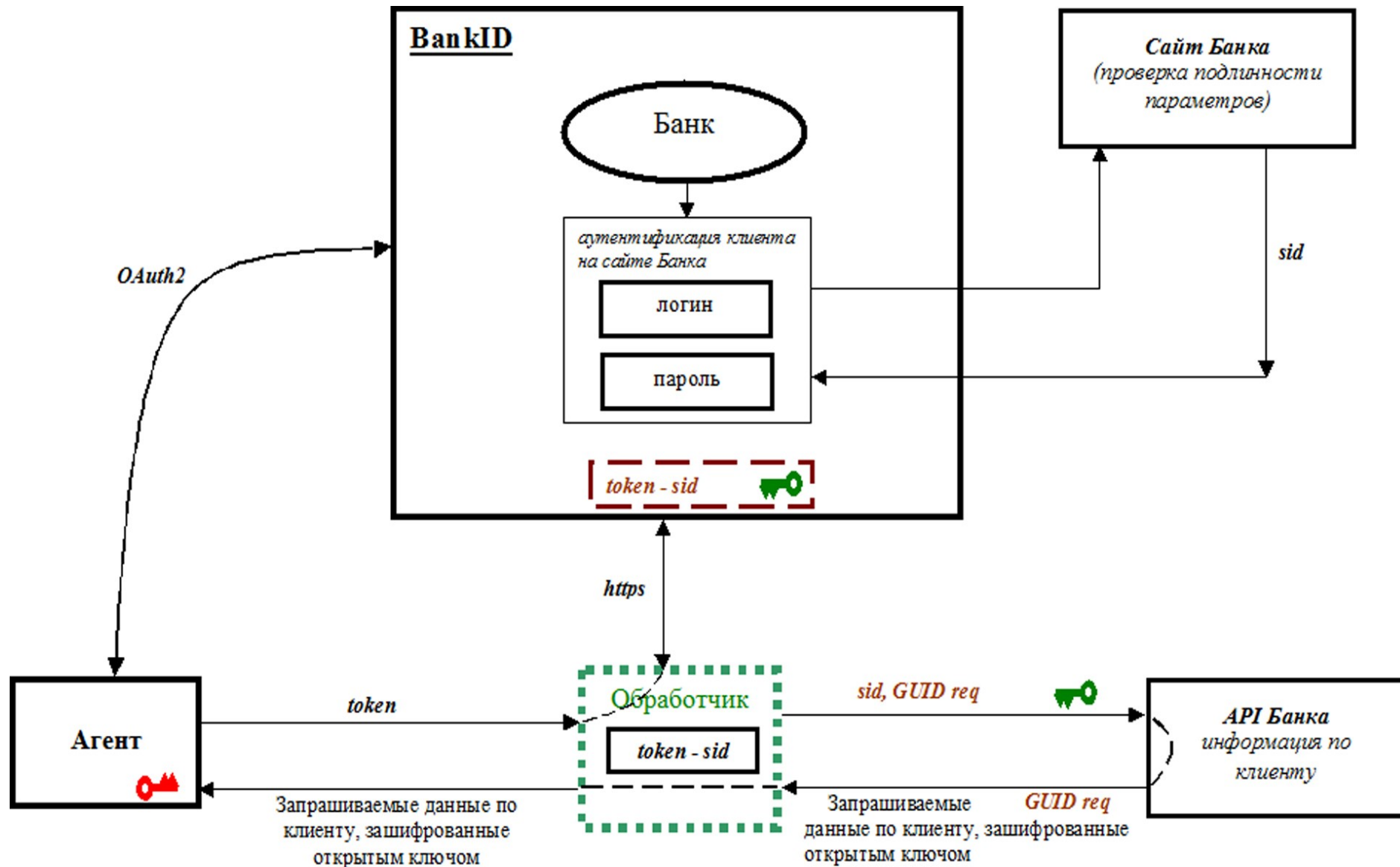
В основе передачи данных лежит SSL протокол. Пересылаемые данные в запросе от Банка к Агенту, шифруются открытым Агентским ключом. (RSA/2048), а так же накладывается подпись SHA1 для целостности протокола.

Схема, которая отражает общие подходы удобна тем, что для Агентов и для Банков существует только один поставщик с одной стороны и один потребитель с другой.

Вся техническая нагрузка и бизнес-логическое управление лежит в плоскости BankID.

Такой подход максимально продуктивен и удобен для всех.

Общая модель взаимодействия Агент - BankID



Краткое описание авторизации внешних Агентов по OAuth2.0

1. Согласно спецификации протокола OAuth2.0, BankID выделяет Агенту для одной услуги *client_id* (уникальный идентификатор вашей услуги на сайте Агента, вида f355f894-00a4-47a8-b5d0-d7431f8ade97) и *client_secret* (секрет, согласно которого вы будите получать *token*, OGRkMDZjZjg5NWU0Yzk0ZjgxOGY5MWZjMWQ5MThjYQ==).

2. Агент со своего сайта перенаправляет интернет браузер клиента по ссылке методом GET:

https://{IP:port}/DataAccessService/das/authorize?response_type=code&client_id=client_id&redirect_uri=http://vash/url/dlia/redirect

где *redirect_uri* - это URL_CallBack Агента, на который BankID перенаправит интернет браузер клиента после аутентификации в Банке.

Дополнение : Расширенные функциональные возможности первого запроса.

Через *redirect_uri* будет передан параметр *code*, запрос вида:

http://vash/url/dlia/redirect?code=jOS6lz

PS: *redirect_uri* можно прошить в BankID на уровне параметров *client_id* и *client_secret* для поднятия безопасности и тогда у Агента отпадет необходимость передавать его в запросах.

3. Агент, с полученным кодом, делаете GET запрос на получение *access_token*. В запросе передается код авторизации *code*, *client_id*, *client_secret* и свой *redirect_uri*

https://{IP:port}/DataAccessService/oauth/token?grant_type=authorization_code&client_id=client_id&client_secret=SHA1{client_id+client_secret+code}&code=jOS6lz&redirect_uri=http://vash/url/dlia/redirect

В ответ Агент получит *access_token* и *refresh_token* в таком формате

{"access_token":"095be9eb-01e7-4045-b60b-9d71581fb4d9","token_type":"bearer","refresh_token":"a3798f16-8039-49c9-9b61-82936977489d","expires_in":179,"scope":"read trust write"}

или отказ

{"error":"invalid_grant","error_description":"Invalid authorization code: jOS6lz"}

4. Определившись с получением *access_token*, Агент делает запрос методом POST для получения данных по клиенту в

BankID, передав параметрами *client_id* и заполнить следующие поля в Header:

Content-Type = "application/json"

Authorization = "Bearer access_token, Id client_id"

Accept = "application/json"

https://{IP:port}/ResourceService/checked/data

PS: Продлить время жизни access_token можно через refresh_token

https://{IP:port}/DataAccessService/oauth/token?grant_type=refresh_token&client_id=client_id&client_secret=SHA1{client_id+client_secret+refresh_token}&refresh_token=refresh_token

Описание запроса получения данных по клиенту согласно *access_token*

В основе запроса лежит принцип заявки.

Агенту необходимо задекларировать с какими полями он будет работать в ходе оказания услуги клиенту. Таким образом в сторону BankID необходимо сделать запрос вида:

Request полной декларации полей JSON формата методом POST:

https://{IP:port}/ResourceService/checked/data

Заполнить следующие поля в Header:

Content-Type = "application/json"

*Authorization = "Bearer **access_token**, Id **client_id**" - (последовательность не важна)*

Accept = "application/json"

Внимание ! Перечисленные поля в запросе, будут отданы в ответе. Если какое-то из полей будет отсутствовать в точках выдачи информации (Банки), заказанное поле не вернется.

передать в теле запроса декларацию полей:

```

{"type":"physical",
"cardToCheck":
"nw5zznM73Jq2qBE6z8cu9unxV2iP5IxTLieygtW/q4pyd/tu2wKyG4Q/CF9fGuLWDVMwsG1I6XTIM8oGjduN6Uns6U+/AEkh34sWSn
S5L3aKHT0jiR3IIXwE6NAwG/FZwKzFaw3WEfCwfidxIlkFLZA+
+g11E1gGfRNPo4D7d39Q426opLrVYprhFEqKeJ4tKeqV5F9G739pR7pHC1nKdLHr9G6Lm0uPo0BLyuMikZWsuJJK/MSx8LlnO5/NPS
+lgVX77noXMMXwXGi61VCmFQEkcSlb4j7qYYswiL+ar8puOBHUG/F3XXCBOztMH+gsF8HH/lvpiRwr42DVx/F26Q==",
"fields":
["firstName","middleName","lastName","phone","inn","clId","clIdText","birthDay","email","sex","resident","dateModification"],
"scans":[
{"type":"passport","fields":["link","dateCreate","extension","dateModification"]},
/*{"type":"zpassport","fields":["link","dateCreate","extension","dateModification"]},*/
/*{"type":"inn","fields":["link","dateCreate","extension","dateModification"]},*/
/*{"type":"personalPhoto","fields":["link","dateCreate","extension","dateModification"]},*/
"addresses":[
{"type":"factual","fields":["country","state","area","city","street","houseNo","flatNo","dateModification"]},
/*{"type":"birth","fields":["country","state","area","city","street","houseNo","flatNo","dateModification"]},*/
"documents":[

```

```

{"type":"passport","fields":["series","number","issue","dateIssue","dateExpiration","issueCountryIso2","dateModification"]},
/*{"type":"zpassport","fields":["series","number","issue","dateIssue","dateExpiration","issueCountryIso2","dateModification"]}*/
}

```

Область данных общей информации	
firstName	имя
middleName	отчество
lastName	фамилия
phone	актуальный телефон
inn	идентификационный номер налогоплательщика
clientId	идентификатор клиента (необязательное поле и может отсутствовать)
clientIdText	статических текст подтверждения выдачи информации <Передана інформація є достовірною і підтверджена BankID dd.MM.yyyy HH:mm>
birthDay	дата рождения
sex	пол (М - мужской, F - женский)
email	email
resident	определение резидентности (стандарт ISO(2)) UA — резидент, /UA — нерезидент
dateModification	последняя дата модификации данных в объекте (dd.MM.yyyy HH:mm:ss.S)

Область данных адреса регистрации/рождения																	
addresses	<table border="1"> <tr> <td>type</td> <td>тип адреса factual — адрес регистрации birth – адрес рождения</td> </tr> <tr> <td>country</td> <td>страна</td> </tr> <tr> <td>state</td> <td>область</td> </tr> <tr> <td>area</td> <td>район</td> </tr> <tr> <td>city</td> <td>город</td> </tr> <tr> <td>street</td> <td>улица</td> </tr> <tr> <td>houseNo</td> <td>номер дома</td> </tr> <tr> <td>flatNo</td> <td>номер квартиры</td> </tr> </table>	type	тип адреса factual — адрес регистрации birth – адрес рождения	country	страна	state	область	area	район	city	город	street	улица	houseNo	номер дома	flatNo	номер квартиры
	type	тип адреса factual — адрес регистрации birth – адрес рождения															
	country	страна															
	state	область															
	area	район															
	city	город															
	street	улица															
	houseNo	номер дома															
	flatNo	номер квартиры															
Область данных документа регистрации																	
documents	<table border="1"> <tr> <td>type</td> <td>тип документа passport – гражданский паспорт zpassport – загранпаспорт ident – удостоверение личности (высылается любой документ регистрации)</td> </tr> <tr> <td>series</td> <td>серия</td> </tr> <tr> <td>number</td> <td>номер</td> </tr> <tr> <td>issue</td> <td>кем выдан</td> </tr> <tr> <td>dateIssue</td> <td>когда выдан</td> </tr> </table>	type	тип документа passport – гражданский паспорт zpassport – загранпаспорт ident – удостоверение личности (высылается любой документ регистрации)	series	серия	number	номер	issue	кем выдан	dateIssue	когда выдан						
	type	тип документа passport – гражданский паспорт zpassport – загранпаспорт ident – удостоверение личности (высылается любой документ регистрации)															
	series	серия															
	number	номер															
	issue	кем выдан															
dateIssue	когда выдан																

	dateExpiration	срок действия (dd.ММ.уууу)
	issueCountryIso2	страна выдачи
scans	Сканы	
	type	<p style="text-align: center;">типы сканов</p> <p>passport — скан гражданского паспорта</p> <p>zpassport — скан загранпаспорта</p> <p>inn – скан идентификационного налогового номера</p> <p>personalPhoto – фотография личности (анфас)</p>
	link	URL выкачки скана
	dateCreate	дата сканирования
	extension	тип расширения файла
	signature	подпись SHA1
cardToCheck	номер карты клиента (ПриватБанка, А-Банк) в зашифрованном виде с помощью закрытого (приватного) ключа Агента	

Ответ будет предоставлен в формате декларации запроса, на пример:

Внимание ! Для обратной совместимости с предыдущими версиями работы АПИ, необходимо обращать внимание на атрибут `signature`. Его наличие в ответе данных будет говорить о том, что данные по полям зашифрованы. Расшифровать значения поля необходимо закрытым ключом пары, но перед расшифровкой, на каждое из зашифрованных значений необходимо наложить `decodeBase64`. Открытый ключ хранится на стороне BankID и является гарантом целостности данных и ограждает от несанкционированного просмотра данных третьими лицами.

Внимание ! Возможный вид бизнесовой ошибки, которая считается положительной (платной)

```
{"state":"err","code":"CLD010","desc":"Помилка, карта не належить авторизованому клієнту"}
```

Response

```
{"state":"ok","customer":
  {
    "type":"physical",
    "inn":"1112618222",/*если есть открытый ключ, значение может выглядеть так: "inn":"fgvb5476/v8634c347f64tf4f/3452g2985f"*/
    "birthDay":"20.01.1973",
    "clId":"46378c05eb48127552e60cffd086e6e34287ff04",
    "firstName":"ЄВГЕН",
    "lastName":"САЛЮ",
    "middleName":"МИКОЛАЙОВИЧ",
    "phone":"+380681231212",
    "signature":"bad3419d8f4aee6945636e8b3fe7e492798e4564",
    "addresses":[
      {"type":"factual",
        "country":"UA",
        "state":"ДНЕПРОПЕТРОВСКАЯ",
```

```
"city": "ДНЕПРОПЕТРОВСК",
"street": "МИРА ",
"houseNo": "11",
"flatNo": "75", "dateModification": "19.06.2015 00:47:36"},
{"type": "birth",
"country": "UA",
"street": "Нет улицы", "dateModification": "29.05.2014 17:52:25"}],
"documents": [
{"type": "passport",
"series": "ШО",
"number": "123456",
"issue": "ДНЕПРОПЕТРОВСКИМ РО УМВД",
"dateIssue": "11.06.1997",
"issueCountryIso2": "UA", "dateModification": "29.05.2014 17:52:25"}],
"scans": [
{"type": "passport",
"link": "http://10.0.0.15:29932/DataAccessService/checked/scan/pasport",
"dateCreate": "09.04.2015",
"extension": "zip",
"dateModification": "29.05.2014 17:52:25"},
{"type": "zpassport",
"link": "http://10.0.0.15:29932/DataAccessService/checked/scan/zpasport",
"dateCreate": "09.04.2015"}]}
}
```

Работа со сканами

Если Агент заказал объект скан копий документов, то в ответ на запрос будут предоставлены ссылки для загрузки файлов сканов см. ниже.

```
. . . . .  
"scans":[  
  {"type":"passport",  
   "link":"http://10.0.0.15:29932/DataAccessService/checked/scan/pasport",  
   "dateCreate":"09.04.2015"},  
  {"type":"zpassport",  
   "link":"http://10.0.0.15:29932/DataAccessService/checked/scan/zpasport",  
   "dateCreate":"09.04.2015"}]}
```

.

Для того, что бы загрузить файл с диска BankID, необходимо сделать запрос согласно присланного линка методом GET:

Заполнить следующие поля в Header:

*Authorization = "Bearer **access_token**, **Id client_id**" - (последовательность не важна)*

Accept = "application/json"

В ответе можно проследить с каким именем будет прислан объект (файл) *Content-Disposition:attachment; filename=passport.zip*

Возможные виды ошибок

Ошибки в BankID делятся на две категории: **логические** и **технические**.

В любом случае, рекомендуем ориентироваться на [Список кодов состояния HTTP](#).

Если статус http запроса равен 200, то необходимо проверить тело в запросе (логическая ошибка) и оценить ситуацию согласно протокола вида:

```
{"state":"err","code":"CLO001","desc":"Ошибка получения данных клиента"}
```

или для XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```
<message state="err">
```

```
<code>CLO001</code>
```

```
<desc>Ошибка получения данных клиента</desc>
```

```
</message>
```

В том случае, если статус http запроса НЕ равен 200, то это один из видов технической ошибки и ее необходимо обработать в соответствии [Список кодов состояния HTTP](#).

На пример:

код 401

http://10.0.0.15:29932/DataAccessService/checked/data?access_token=3fb13e8e-1b49-4979-8cba-68fddb7f6e29

```
{"error":"invalid_token","error_description":"Invalid access_token:3fb13e8e-1b49-4979-8cba-68fddb7f6e29"}
```

<oauth>

```
<error_description>Invalid access token: 4934b13d-02e7-42c2-8054-df81911113c0</error_description>
```

```
<error>invalid_token</error>
```

</oauth>

код 401

http://10.0.0.15:29932/DataAccessService/checked/data?access_token=a47a7c69-68b8-4780-bc7a-f726d47aa2d1

```
{"error":"invalid_token","error_description":"Access token expired:a47a7c69-68b8-4780-bc7a-f726d47aa2d1"}
```

<oauth>

```
<error_description>Access token expired: 159e0450-ff32-4367-84c0-5ac1ea286315</error_description>
```

```
<error>invalid_token</error>
```

</oauth>

код 401

http://10.0.0.15:29932/DataAccessService/checked/data

```
{"error":"unauthorized","error_description":"An Authentication object was not found in the SecurityContext"}
```

```
<oauth>
```

```
<error_description>An Authentication object was not found in the SecurityContext</error_description>
```

```
<error>unauthorized</error>
```

```
</oauth>
```

код 400

http://10.0.0.15:29932/DataAccessService/oauth/token?

grant_type=authorization_code&client_id=client_id&client_secret=client_secret&code=V3xMqk10099&redirect_uri=http://10.0.0.2

14/DataAccessService/acceptCode

```
{"error":"invalid_grant","error_description":"Invalid authorization code: V3xMqk10099"}
```


Работа с ЭЦП

В BankID разработан модуль работы с ЭЦП. Т.е. верифицировать Гражданина можно по средствам личной ЭЦП и получить из сертификата данные о владельце ЭЦП.

Верификация Гражданина по ЭЦП

Согласно стр. 4 пункта *Краткое описание авторизации внешних Агентов по OAuth2.0* настоящего документа, авторизация Агента происходит по средствам протокола *OAuth2.0*. Единственное отличие, это дополнительный параметры в первом запросе: *eds=true*

*https://{IP:port}/DataAccessService/das/authorize?response_type=code&client_id=client_id&eds=true&redirect_uri=
http://vash/url/dlia/redirect*

Результатом вызова, будет Java апплет.

Получение токена и забор данных происходит стандартным образом согласно стр..4,5.

Проверка подписанного объекта на подлинность и получение информации о владельце ЭЦП

1. Авторизация Агента

Обмен данными происходит по схеме H2H, т. е. от сервиса Агента к сервису BankID.

Авторизация Агента идет согласно штатной схемы стр.. 4,5.

Отличием является первый запрос в котором добавлен параметр: *edsVerify=true*

https://{IP:port}/DataAccessService/das/authorize?response_type=code&client_id=client_id&edsVerify=true&redirect_uri=http://vash/url/dlia/redirect

2. Описание запроса получения данных из ЭЦП по клиенту согласно *access_token*

В сторону BankID необходимо сделать запрос вида:

Request полной декларации полей JSON формата методом POST:

https://{IP:port}/ResourceService/checked/signatureData

Заполнить следующие поля в Header:

Content-Type = "multipart/form-data"

Authorization = "Bearer access_token, Id client_id" - (последовательность не важна)

Accept = "application/json"

передать в теле запроса объект (файл) в бинарном или текстовом виде.

Ответ будет предоставлен в формате json объекта *customers* списка наложенных ЭЦП , на пример:

```
{  
  "state": "ok",  
  "customers": [{ /*Список, кто наложил ЭЦП*/  
    "phone": "(067) 6145793",  
    "inn": "2310907619", /*ИНН для физического лица, ОКПО для юридического лица*/  
    "email": "shaliv@mail.ru",  
    "fullName": "Шкіль Олександр Іванович",  
    "addresses": [{ /*Адрес регистрации */  
      "type": "factual",  
      "fullAddress": "69041, Запорізька обл., м. Запоріжжя, вул. Кремлівська, 79 А, кв. 112"  
    }],  
    "signatureData": { /*Информация об ЭЦП*/  
      "name": "Акредитований центр сертифікації ключів ІДД ДФС",  
      "serialNumber": "33B6CB7BF721B9CE04000004B130F00769B3B00", /*Серийный номер ЭЦП*/  
      "timestamp": "19.04.2016 20:37:19", /*Время наложения ЭЦП*/  
      "code": true, /*Проверка действительности ЭЦП в момент подписания объекта*/  
      "desc": "ПІДПИС ВІРНИЙ", /*Проверка действительности ЭЦП в момент подписания объекта*/  
      "dateFrom": "08.12.2015 00:00:00", /*Время с которого действительна ЭЦП*/  
      "dateTo": "08.12.2017 00:00:00", /*Время до которого действительна ЭЦП*/  
      "sn": "UA-39384476" /*Порядковый номер в базе АЦСК */  
    },  
  },  
  "organizations": [{ /*Информация об организации выдавшая ЭЦП */  
    "type": "edsIssuer",  
    "name": "Інформаційно-довідковий департамент ДФС",  
    "unit": "Управління (центр) сертифікації ключів ІДД ДФС",  
    "address": {  
      "type": "factual",  
      "city": "Київ"  
    }  
  }  
}
```

```

}, {
  "type": "edsOwner", /*Дополнительная информация владельца ЭЦП */
  "name": "Фізична особа - підприємець Шкіль Олександр Іванович", /*Статус владельца ЭЦП, может быть Физическим лицом */
  "unit": "Фізична особа - підприємець",
  "mfo": "2310907619", /*МФО для юридического лица*/
  "position": "фізична особа - підприємець",
  "ownerDesc": "Юридична особа",
  "address": {
    "type": "factual",
    "state": "Запорізька",
    "city": "Запоріжжя"
  }
}
}
}, {
  "phone": "(067) 6145793",
  "inn": "2310907619",
  "email": "shaliv@mail.ru",
  "fullName": "електронна печатка ФО - П Шкіль О.І.",
  "addresses": [{
    "type": "factual",
    "fullAddress": "69041, Запорізька обл., м. Запоріжжя, вул. Кремлівська, 79 А, кв. 112"
  }],
  "signatureData": {
    "name": "Акредитований центр сертифікації ключів ІДД ДФС",
    "serialNumber": "33B6CB7BF721B9CE040000055130F00849B3B00",
    "timestamp": "19.04.2016 20:36:43",
    "code": true,
    "desc": "ПІДПИС ВІРНИЙ",
    "dateFrom": "08.12.2015 00:00:00",
  }
}

```

```
"dateTo": "08.12.2017 00:00:00",
"sn": "UA-39384476"
},
"organizations": [{
  "type": "edsOwner",
  "name": "Фізична особа - підприємець Шкіль Олександр Іванович",
  "unit": "Фізична особа - підприємець",
  "mfo": "2310907619",
  "ownerDesc": "Юридична особа",
  "address": {
    "type": "factual",
    "state": "Запорізька",
    "city": "Запоріжжя"
  }
}, {
  "type": "edsIsuer",
  "name": "Інформаційно-довідковий департамент ДФС",
  "unit": "Управління (центр) сертифікації ключів ІДД ДФС",
  "address": {
    "type": "factual",
    "city": "Київ"
  }
}]
}]
}
```

Модуль мульти наложения ЭЦП

Обмен данными происходит в пять этапов:

- 1) авторизация сайта Агента по работе с методами BankID идет согласно штатной схемы см. стр. 4,5, результат - получение **access_token**;
- 2) передача объекта (объектов, средствами *multipartEntry*), результат - получение **URL_redirect** на который необходимо переориентировать браузер Гражданина;
- 3) после выполнения п.2, браузер Гражданина будет перенаправлен сервисом Агента на **URL_redirect**, механизм работы с ЭЦП, результат - загрузится **крипто плагин**;
- 4) Гражданин накладывает ЭЦП, путем выбора ключа и указанием пароля от него. Результат - подпись будет произведена на весь спектр объектов и перенаправление браузера на **URL_callback**, который ранее был получен от Агента в п.2 в шапке запроса. В **URL_callback** будет передан авторизационный ключ **code** для забора подписанного пула объектов запакованных в zip или одного объекта в большинстве случаев PDF формата;
- 5) вызвать метод BankID, передав авторизационный ключ **code**, результат - будет отдан zip (архив объектов) или PDF, или иной вид объекта (см. на Content-Type) средствами *multipartEntry*.

Методы и параметры для работы:

Для отработки по п. 2, необходимо выполнить п.1, получить **access_token**.

Сформировать запрос методом POST

<https://{PI:port}/ResourceService/checked/uploadFilesForSignature>

Заполнить следующие поля в Header:

Content-Type = "multipart/form-data"

*Authorization = "Bearer **access_token**, Id **client_id**" - (последовательность не важна)*

Accept = "application/json"

*acceptKeyUrl = **URL_callback** - (урл перенаправления браузера и передача ключа для забора подписанного PDF)*

передать в теле запроса объект (объекты) в бинарном или текстовом виде.

В результате ответа будет получен JSON вида

```
{"state":"ok","code":"000000","desc":"https://{IP:port}/IdentDigitalSignature/signPdf?sidBi=sidBi_value"}
```

В поле *desc* будет находится **URL_redirect** на который необходимо переориентировать браузер Гражданина, т.е. сделать редирект.

Во время отработки **URL_redirect**-а наступает фаза п.3

После отработки п.3 (подписание), BankID делает редирект на *https://{PI:port}/URL_callback?code=code_value* с передачей параметра авторизационного ключа *code*, тем самым заканчивая фазу п.4.

Для выполнения финального п. 5, необходимо сформировать запрос метода GET, передав в нем код авторизационного ключа *code*
https://{PI:port}/ResourceService/checked/claim/code_value/clientPdfClaim
Authorization = "Bearer access_token, Id client_id" - (последовательность не важна)

Во время отработки п. 5, просим обратить внимание на поля: **Content-Disposition** и **Content-Type**. В случае множественного подписания объектов, модуль BankID отдаст zip - архив.

Например

```
Content-Disposition=[attachment; filename=clientPdfClaim.zip]
```

```
Content-Type=[application/zip; charset=UTF-8]
```

В том случае, если было подписание одного объекта, необходимо ожидать ответ в формате PDF (в случае возможности конвертации в PDF) или в "родном" расширении.

В качестве примера

```
Content-Disposition=[attachment; filename=clientPdfClaim.pdf]
```

```
Content-Type=[application/pdf; charset=UTF-8]
```

или

Content-Disposition=[attachment; filename=clientPdfClaim.doc]

Content-Type=[application/octet-stream; charset=UTF-8]

Внимание ! Время жизни *access_token* составляет 3 мин. За это отведенное время, Гражданин может не успеть вовремя подписать объект, тем самым возбудить ошибку с кодом 401 вида {"error":"invalid_token","error_description":"Access token expired:a47a7c69-68b8-4780-bc7a-f726d47aa2d1"}. Для восстановления работы, а вернее продолжения, рекомендуем получить новый *access_token* путем отработки *refresh_token* см. стр. 5.

Авторизации внешних Агентов по OAuth2.0 с проверкой принадлежности карты

1. Авторизация Агента

Обмен данными происходит по схеме H2H, т. е. от сервиса Агента к сервису BankID, без участия гражданина.

Авторизация Агента идет согласно штатной схемы стр.. 4,5.

Отличием является первый запрос в котором добавлены параметры:

agentCheck=true

bank=privat (или abank), т. е. идет поддержка проверки карты клиента ПриватБанка и А-Банка

https://{IP:port}/DataAccessService/das/authorize?

response_type=code&client_id=client_id&bank=privat&agentCheck=true&redirect_uri= http://vash/url/dlia/redirect

Внимание ! Возможный вид бизнесовой ошибки, которая считается положительной (платной)

{"state":"err","code":"CLD011","desc":"Помилка, карта не належить до карт що видав ПриватБанк / А-Банк"}

Правки и дополнения

Место по тексту	Было	Стало
<p>Модуль наложения ЭЦП на заявку</p> <p>страница 21/ v4.04</p>	<p><i>Модуль наложения ЭЦП на заявку</i></p>	<p><i>Модуль мульти наложения ЭЦП</i></p>
<p>Описание запроса получения данных из ЭЦП по клиенту согласно <i>access_token</i></p> <p>страница 18/ v4.05</p>	<p><i>customers</i></p>	<p><i>customers(список наложенных ЭЦП)</i></p>
<p>Авторизации внешних Агентов по OAuth2.0 с проверкой принадлежности карты</p> <p>страница 25/ v4.06</p>	<p>-</p>	<p><i>Авторизации внешних Агентов по OAuth2.0 с проверкой принадлежности карты</i></p>