

My Art-Office API
version 1.2

Зміст

Опис API для інтеграції із зовнішніми системами.....	3
Загальний опис алгоритму обміну даними.....	3
Опис функції завантаження документа в систему.....	4
Опис функції створення первинного документа.....	7
Опис функції вивантаження документа з системи.....	10
Опис функції вивантаження XML первинного документа з системи.....	11
Опис функції вивантаження ZIP архіву документа з підписами.....	12
Опис функції отримання статусів документів.....	13
Опис функції отримання статусів документів з параметрами.....	15
Опис функції публікації документа.....	18
Опис функції відхилення документа.....	19
Опис функції видалення документа.....	20
Опис функції коментування документа.....	21
Опис функції отримання коментарів для документа.....	22
Опис функції трекінгу коментарів.....	24
Опис функції створення зв'язку між документами.....	25
Опис функції видалення зв'язку між документами.....	25
Опис функції отримання зв'язків для документа.....	25
Опис функції трекінгу кількості зв'язків для документа.....	26
Опис функції збереження зображень факсиміле підпису та печатки.....	26
Опис функції вивантаження зображень факсиміле підпису та печатки.....	27
Словники.....	28
Таблиця «Типи файлів».....	28
Таблиця «Види документів».....	28
Таблиця «Групи підписантів».....	28
Таблиця «Типи контрагентів».....	29
Таблиця «Типи печатки».....	29
Таблиця «Стан документа».....	29
Таблиця «Стан підписання».....	29
Таблиця «Типи доступу».....	29
Таблиця «Ідентифікатори шаблонів первинних документів».....	29
Таблиця «Типи зв'язків між документами».....	30

Опис API для інтеграції із зовнішніми системами

Для використання API необхідно мати особистий приватний ключ, його можна отримати в блоці «Загальна інформація» на сторінці «Профіль користувача» (<https://my.art-office.com.ua/home/docuser#/app/profile>)

Загальний опис алгоритму обміну даними

Передача даних на сервіс здійснюється шляхом відправки форми методом **POST** на URL (URL вказаний нижче для кожної функції окремо). Форма повинна мати два поля: **data** и **sign**.

Поле **data** містить в собі JSON рядок з даними в кодуванні UTF-8, трансформовану в байти, а потім в base64. Для кожної функції формат JSON рядку різний, він буде описаний нижче.

Поле **sign** містить в собі JSON рядок з даними для ідентифікації користувача та верифікації поля **data** в кодуванні UTF-8, трансформовану в байти, а потім в base64.

Приклад JSON для поля **sign**:

```
{
  "UserCertificate": ".....", //сертифікат підпису користувача в base64
  "SubjectType": 3, //тип суб'єкта, табл «Типи контрагентів», необхідно
  //передавати, якщо у ФОП-а в сертифікаті не вказано
  //ЄДРПОУ, інакше система визначає його як фізичну
  //особу
  "Sign": "....." //хеш отриманий з байтів поля data
  //за допомогою алгоритму SHA512 з
  //використанням приватного ключа користувача
}
```

Приклад, отримання хеша мовою програмування C#:

```
private string GetHashCode(string dataJSON, string key)
{
    byte[] data = Encoding.UTF8.GetBytes(dataJSON);
    byte[] keyBytes = Encoding.UTF8.GetBytes(key);
    byte[] result;
    using (var sha = new HMACSHA512(keyBytes))
        result = sha.ComputeHash(data);

    return BitConverter.ToString(result).Replace("-", "").ToLowerInvariant();
}
```

В момент, коли сервіс отримує запит, він формує хеш з байтів поля **data** по вище описаному алгоритму. Потім перевіряє, що хеш з поля **sign** співпадає з отриманим. Якщо хеші не співпадають, система видає помилку.

Якщо в результаті виклику функції виникла помилка, то у відповіді прийде рядок JSON у наступному форматі:

```
{"Error" : "опис помилки"}
```

Після успішного виконання функції сервіс поверне рядок JSON у форматі:

```
{ "data":"responseBase64", "sign":"responseSignBase64" }
```

де responseBase64 – це рядок JSON у кодування UTF-8, трансформований в байти, а потім в base64.

Для кожної функції результат виконання (формат JSON) буде описаний нижче.

responseSignBase64 – це хеш з байтів responseBase64, отриманий по правилам описаним вище у форматі base64. Стороння система за необхідності може перевіряти підпис даних по алгоритму, описаному вище, як це робить сервіс.

Опис функції завантаження документа в систему

URL, на який необхідно відправляти форму: <https://my.art-office.com.ua/service/createdocument>.

Формат JSON рядку, що передається в поле **data**:

```
{
  "DocumentData": //інформація про документ
  {
    "FileName": "1.docx", //оригінальна назва файлу
    "FileTypeId": 1, //з таблиці «Типи файлів»
    "TypeId": 1, //з таблиці «Види документів»
    "PolicyId": 1, //ідентифікатор політики підпису
    "Name": "Акт виконаних робіт", //назва документа
    "IsDraft":true, //ознака чернетки, якщо встановлено, то
    //документ не доступний контрагенту, поки
    //його не опублікують
    "ExtData":{ //дополнительные данные о документе
      "DocNumber": "1112", //номер документа
      "DocDateStr": "22.05.2018", //дата документа
      "AgreementNumber": "asd-11", //номер договору
      "AgreementDateStr": "09.09.2015", //дата договору
      "SumPdvStr": "20.3", //сума ПДВ
      "SumWithPdvStr": "101.22", //сума з ПДВ
      "SumWoPdvStr": "80.98", //сума без ПДВ
      "CustomData": "abc" //Будь-яка додаткова інформація про документ
    },
    "MembersData": //інформація про підписантів, масив
    [{
      "GroupId": 1, //з таблиці «Групи підписантів»
      "ContragentType": 3, //з таблиці «Типи контрагентів»
      "DrfoCode": "", //ІПН код для фіз. осіб
      "EdrpouCode": "3031111144", //ЄДРПОУ для організацій та ФОП
      "StampType": 2, //з таблиці «Типи печатки»
      "Email": "abc@in.ua" //адреса ел.пошти контрагента, на яку прийде
      //нотифікація
    },
    {
      "GroupId": 2,
      "ContragentType": 3,
      "DrfoCode": "",
      "EdrpouCode": "3031111555",
      "StampType": 1,
      "Email": "abcd@in.ua"
    }
  ]
},
```

```

    "DocShareItems":
    [{
        "ShareTypeId" : 3, //інформація про надання доступу, масив
        "Edrpou" : "43344334", //допустима лише для односторонніх документів
        "Ipn" : "", //допустим лише один елемент в масиві
        "Email": "abc@in.ua" //з таблиці «Типи доступу»
        //ЄДРПОУ для організацій/ФОП
        //ІПН код для фіз. осіб/співробітників
        //адреса ел.пошти контрагента, на яку прийде
        //нотифікація
    }],
    "DocumentBody": "....." //тіло документа в base64
}

```

При заповненні вхідних параметрів необхідно дотримуватись наступних правил:

- При значенні "ContragentType": 1 та "ContragentType": 3 необхідно вказувати лише "EdrpouCode", значення "DrfoCode" повинно бути пустим;
- При значенні "ContragentType": 2 необхідно вказувати лише " DrfoCode", значення "EdrpouCode" повинно бути пустим;
- При значенні "ContragentType": 2, "StampType" не може мати значення 1;
- Значення "GroupId" для елементів масиву "MembersData" повинно відрізнятися;
- Значення "FileTypeId" повинно відповідати типу та розширенню файлу;
- При зазначенні "ShareTypeId":3 доступ надається всій організації з вказаним кодом "Edrpou";
- При зазначенні "ShareTypeId":2 доступ надається фіз. особі з вказаним кодом "Ipn".

Формат JSON рядку результату виконання функції:

```
{"DocumentId": <Ідентифікатор документа(число)>}
```

Приклад виклику функції мовою програмування C#:

```

var doc = File.ReadAllBytes("c:\\temp\\1.docx");
var dataJson = @"
{""DocumentData"": {
    ""FileName"": ""1.docx"",
    ""FileTypeId"": 1,
    ""TypeId"": 1,
    ""Name"": "" Акт виконаних робіт"",
    ""IsDraft"":true,
    ""ExtData"":{
        ""DocNumber"": ""1112"",
        ""DocDateStr"": ""22.05.2018"",
        ""AgreementNumber"": "asd-11"",
        ""AgreementDateStr"": ""09.09.2015"",
        ""SumPdvStr": ""20.3"",
        ""SumWithPdvStr"": ""101.22"",
        ""SumWoPdvStr"": ""80.98""
    }
},
    ""MembersData"": [{
        ""GroupId"": 1,
        ""ContragentType"": 3,
        ""DrfoCode"": "",
        ""EdrpouCode"": ""3031111144"",
        ""StampType"": 2,

```

```

        ""Email"": ""abc@in.ua""
    },
    {
        ""GroupId"": 1,
        ""ContragentType"": 3,
        ""DrfoCode"": """",
        ""EdrpouCode"": ""3031111555"",
        ""StampType"": 1,
        ""Email"": ""abcd@in.ua""
    }
}],
""DocShareItems"":
[
    {
        ""ShareTypeId"" : 3,
        ""Edrpou"" : ""43344334"",
        ""Ipn"" : """",
        ""Email"": ""abcd1@in.ua""
    }
],
""DocumentBody"": "" + Convert.ToBase64String(doc) + @"""";

var dataBytes = Encoding.UTF8.GetBytes(dataJson);
var dataBase64 = Convert.ToBase64String(dataBytes);
var client = new RestClient("https://my.art-office.com.ua/service/createdocument");
var request = new RestRequest(Method.POST);

request.AddParameter("data", dataBase64);
var signData = GetHash(dataJson, "79A3E89B-ECFC-4686-8122-F973148B4493");
var cert = File.ReadAllBytes("c:\\temp\\1.cer");
var sign = "{" + string.Format(@"""UserCertificate"": ""{1}""", ""Sign"": ""{0}""", signData,
    Convert.ToBase64String(cert)) + "}";
var signBytes = Encoding.UTF8.GetBytes(sign);
var signBase64 = Convert.ToBase64String(signBytes);
request.AddParameter("sign", signBase64);
var restResponse = (RestResponse)client.Execute(request);
var response = restResponse.Content;
if (response.IndexOf("Error", StringComparison.InvariantCulture) > -1)
{
    //обробка помилки {"Error" : "опис помилки"}
}
else
{
    JavaScriptSerializer js = new JavaScriptSerializer();
    var dataDynamic = js.Deserialize<dynamic>(restResponse);
    dataBase64 = dataDynamic["data"];
    dataBytes = Convert.FromBase64String(dataBase64);
    var jsonResponseData = Encoding.UTF8.GetString(dataBytes);

    signData = GetHash(jsonResponseData, "79A3E89B-ECFC-4686-8122-F973148B4493");
    var responseSignBase64 = dataDynamic["sign"];
    var responseSign =
    Encoding.UTF8.GetString(Convert.FromBase64String(responseSignBase64));

    if (signData.ToLowerInvariant() == responseSign.ToLowerInvariant())
    {
        //обробка результату виконання jsonResponseData
        //{"DocumentId": 1000}
    }
    else
    {

```

```
        //помилка верифікації даних
    }
}
```

Опис функції створення первинного документа

URL, на який необхідно відправляти форму:

<https://my.art-office.com.ua/service/createprimarydocument>.

Первинний документ можна створити, використовуючи вбудований шаблон, тобто передавати тільки xml встановленого формату, при цьому друкований образ документа створиться системою автоматично на основі даних в xml. Також є можливість передати готовий друкований образ документа у одному з форматів, вказаних в таблиці «Типи файлів». В такому випадку система не буде автоматично створювати друкований образ і за відповідність даних в xml та друкованому образі документа відповідає автор документа.

Формат JSON рядку, для створення первинного документа з використанням вбудованого шаблону, що передається в поле **data**:

```
{
  "DocumentData":
  {
    "IsDraft":true //ознака чернетки, якщо встановлено, то
                  //документ не доступний контрагенту, поки
                  //його не опублікують
  },
  "TemplateId": 3, //з таблиці «Ідентифікатори шаблонів
                  //первинних документів»
  "CustomData": "Будь яка додаткова інформація",
  },
  "MembersData": //інформація про підписантів, масив
  [{
    "GroupId": 1, //з таблиці «Групи підписантів»
    "ContragentType": 3, //з таблиці «Типи контрагентів»
    "DrfoCode": "", //ІПН код для фіз. осіб
    "EdrpouCode": "3031111144", //ЄДРПОУ для організацій та ФОП
    "StampType": 2, //з таблиці «Типи печатки»
    "Email": "abc@in.ua" //адреса ел.пошти контрагента, на яку прийде
                        //нотифікація
  },
  {
    "GroupId": 2,
    "ContragentType": 3,
    "DrfoCode": "",
    "EdrpouCode": "3031111555",
    "StampType": 1,
    "Email": "abcd@in.ua"
  }
  ],
  "DocShareItems": //інформація про надання доступу, масив
  [{ //допустима лише для односторонніх документів
    //допустим лише один елемент в масиві
    "ShareTypeId" : 3, //з таблиці «Типи доступу»
    "Edrpou" : "43344334", //ЄДРПОУ для організацій/ФОП
  }
  ]
}
```

```

        "Ipn" : "", //ІПН код для фіз. осіб/співробітників
        "Email": "abc@in.ua" //адреса ел.пошти контрагента, на яку прийде
                                //нотифікація
    }],
    "DocumentBody": "....." //тіло xml первинного документа в base64
}

```

Формат JSON рядку, для створення первинного документа з власним друкованим образом, що передається в поле **data**:

```

{
  "DocumentData":
  {
    "FileName": "Документ.docx", //назва документа
    "FileTypeId": 1, //з таблиці «Типи файлів»
    "IsDraft":true //ознака чернетки, якщо встановлено, то
                    //документ не доступний контрагенту, поки
                    //його не опублікують
  },
  "TemplateId": 3, //з таблиці «Ідентифікатори шаблонів
                  //первинних документів»
  "CustomData": "Будь яка додаткова інформація",
  "MembersData": //інформація про підписантів, масив
  [{
    "GroupId": 1, //з таблиці «Групи підписантів»
    "ContragentType": 3, //з таблиці «Типи контрагентів»
    "DrfoCode": "", //ІПН код для фіз. осіб
    "EdrpouCode": "303111144", //ЄДРПОУ для організацій та ФОП
    "StampType": 2, //з таблиці «Типи печатки»
    "Email": "abc@in.ua" //адреса ел.пошти контрагента, на яку прийде
                          //нотифікація
  },
  {
    "GroupId": 2,
    "ContragentType": 3,
    "DrfoCode": "",
    "EdrpouCode": "3031111555",
    "StampType": 1,
    "Email": "abcd@in.ua"
  }
  ],
  "DocShareItems": //інформація про надання доступу, масив
  [{ //допустима лише для односторонніх документів
    //допустим лише один елемент в масиві
    "ShareTypeId" : 3, //з таблиці «Типи доступу»
    "Edrpou" : "43344334", //ЄДРПОУ для організацій/ФОП
    "Ipn" : "", //ІПН код для фіз. осіб/співробітників
    "Email": "abc@in.ua" //адреса ел.пошти контрагента, на яку прийде
                          //нотифікація
  }],
  "DocumentBody": ".....", //тіло xml первинного документа в base64
  "DocumentViewBody": "....." //тіло друкованого образа документа в base64
}

```

Формат JSON рядку результату виконання функції:

```

{"DocumentId": <Ідентифікатор документа(число)>}

```


Приклад виклику функції (без передачі друкованого образу) на мові програмування C#:

```

var doc = File.ReadAllBytes("c:\\temp\\PD1C820200345543550100000011112017.XML");
var dataJson = @"
{
  ""TemplateId"": 3,
  ""CustomData"": ""Будь яка додаткова інформація"",
  ""MembersData"": [{
    ""GroupId"": 1,
    ""ContragentType"": 3,
    ""DrfoCode"": "",
    ""EdrpouCode"": ""3031111144"",
    ""StampType"": 2,
    ""Email"": ""abc@in.ua""
  },
  {
    ""GroupId"": 1,
    ""ContragentType"": 3,
    ""DrfoCode"": "",
    ""EdrpouCode"": ""3031111555"",
    ""StampType"": 1,
    ""Email"": ""abcd@in.ua""
  }],
  ""DocShareItems"":
  [{
    ""ShareTypeId"" : 3,
    ""Edrpou"" : ""43344334"",
    ""Ipn"" : "",
    ""Email"": ""abcd1@in.ua""
  }],
  ""DocumentBody"": "" + Convert.ToBase64String(doc) + @""""";

var dataBytes = Encoding.UTF8.GetBytes(dataJson);
var dataBase64 = Convert.ToBase64String(dataBytes);
var client = new RestClient("https://<назва сайту>/service/createprimarydocument");
var request = new RestRequest(Method.POST);

request.AddParameter("data", dataBase64);
var signData = GetHash(dataJson, "79A3E89B-ECFC-4686-8122-F973148B4493");
var cert = File.ReadAllBytes("c:\\temp\\1.cer");
var sign = "{" + string.Format(@"""UserCertificate"": ""{1}""", ""Sign"":""{0}""", signData,
    Convert.ToBase64String(cert)) + "}";
var signBytes = Encoding.UTF8.GetBytes(sign);
var signBase64 = Convert.ToBase64String(signBytes);
request.AddParameter("sign", signBase64);
var restResponse = (RestResponse)client.Execute(request);
var response = restResponse.Content;
if (response.IndexOf("Error", StringComparison.InvariantCulture) > -1)
{
  //обробка помилки {"Error" : "опис помилки"}
}
else
{
  JavaScriptSerializer js = new JavaScriptSerializer();
  var dataDynamic = js.Deserialize<dynamic>(restResponse);
  dataBase64 = dataDynamic["data"];
}

```

```
dataBytes = Convert.FromBase64String(dataBase64);
var jsonResponseData = Encoding.UTF8.GetString(dataBytes);

signData = GetHash(jsonResponseData, "79A3E89B-ECFC-4686-8122-F973148B4493");
var responseSignBase64 = dataDynamic["sign"];
var responseSign =
Encoding.UTF8.GetString(Convert.FromBase64String(responseSignBase64));

if (signData.ToLowerInvariant() == responseSign.ToLowerInvariant())
{
    //обробка результату виконання jsonResponseData
    //{"DocumentId": 1000}
}
else
{
    //помилка верифікації даних
}
}
```

Опис функції вивантаження документа з системи

URL, на який необхідно відправляти форму: <https://my.art-office.com.ua/service/getdocument>.

Формат JSON рядку, що передається в поле **data**:

```
{"DocumentId": Ідентифікатор документа(число)}
```

Формат JSON рядку результату виконання функції:

```
{
    "DocumentFileName": "Назва файлу документа",
    "DocumentBody": "....." //тіло документа в base64
}
```

Приклад виклику функції мовою програмування C#:

```
var dataJson = @"{"DocumentId":1000}";

var dataBytes = Encoding.UTF8.GetBytes(dataJson);
var dataBase64 = Convert.ToBase64String(dataBytes);
var client = new RestClient("https://my.art-office.com.ua/service/getdocument");
var request = new RestRequest(Method.POST);

request.AddParameter("data", dataBase64);
var signData = GetHash(dataJson, "79A3E89B-ECFC-4686-8122-F973148B4493");
var cert = File.ReadAllBytes("c:\\temp\\1.cer");
var sign = "{" + string.Format(@"""UserCertificate"": ""{1}""""""Sign"":""{0}""""", signData,
    Convert.ToBase64String(cert)) + "}";
var signBytes = Encoding.UTF8.GetBytes(sign);
var signBase64 = Convert.ToBase64String(signBytes);
request.AddParameter("sign", signBase64);
var restResponse = (RestResponse)client.Execute(request);
var response = restResponse.Content;
if (response.IndexOf("Error", StringComparison.InvariantCulture) > -1)
{
    //обробка помилки {"Error" : "опис помилки"}
```

```
}
else
{
    JavaScriptSerializer js = new JavaScriptSerializer();
    var dataDynamic = js.Deserialize<dynamic>(restResponse);
    dataBase64 = dataDynamic["data"];
    dataBytes = Convert.FromBase64String(dataBase64);
    var jsonResponseData = Encoding.UTF8.GetString(dataBytes);

    signData = GetHash(jsonResponseData, "79A3E89B-ECFC-4686-8122-F973148B4493");
    var responseSignBase64 = dataDynamic["sign"];
    var responseSign =
    Encoding.UTF8.GetString(Convert.FromBase64String(responseSignBase64));

    if (signData.ToLowerInvariant() == responseSign.ToLowerInvariant())
    {
        //обробка результату виконання jsonResponseData
        var dataDynamic1 = js1.Deserialize<dynamic>(jsonResponseData);
        File.WriteAllBytes("c:\\temp\\input\\" + dataDynamic1["DocumentFileName"],
            Convert.FromBase64String(dataDynamic1["DocumentBody"]));
    }
    else
    {
        //помилка верифікації даних
    }
}
}
```

Опис функції вивантаження XML первинного документа з системи

URL, на який необхідно відправляти форму:

<https://my.art-office.com.ua/service/getprimarydocumentxml>.

Формат JSON рядку, що передається в поле **data**:

```
{"DocumentId": Ідентифікатор документа(число)}
```

Формат JSON рядку результату виконання функції:

```
{
    "DocumentFileName": "Назва файлу документа",
    "DocumentBody": "....." //XML первинного документа в base64
}
```

Приклад виклику функції мовою програмування C#:

```
var dataJson = @"{"DocumentId":1000}";

var dataBytes = Encoding.UTF8.GetBytes(dataJson);
var dataBase64 = Convert.ToBase64String(dataBytes);
var client = new RestClient("https://my.art-office.com.ua/service/getprimarydocumentxml");
var request = new RestRequest(Method.POST);

request.AddParameter("data", dataBase64);
var signData = GetHash(dataJson, "79A3E89B-ECFC-4686-8122-F973148B4493");
```

```
var cert = File.ReadAllBytes("c:\\temp\\1.cer");
var sign = "{" + string.Format(@"""UserCertificate"": ""{1}""""Sign"":""{0}""""", signData,
    Convert.ToBase64String(cert)) + "}";
var signBytes = Encoding.UTF8.GetBytes(sign);
var signBase64 = Convert.ToBase64String(signBytes);
request.AddParameter("sign", signBase64);
var restResponse = (RestResponse)client.Execute(request);
var response = restResponse.Content;
if (response.IndexOf("Error", StringComparison.InvariantCulture) > -1)
{
    //обробка помилки {"Error" : "опис помилки"}
}
else
{
    JavaScriptSerializer js = new JavaScriptSerializer();
    var dataDynamic = js.Deserialize<dynamic>(restResponse);
    dataBase64 = dataDynamic["data"];
    dataBytes = Convert.FromBase64String(dataBase64);
    var jsonResponseData = Encoding.UTF8.GetString(dataBytes);

    signData = GetHash(jsonResponseData, "79A3E89B-ECFC-4686-8122-F973148B4493");
    var responseSignBase64 = dataDynamic["sign"];
    var responseSign =
        Encoding.UTF8.GetString(Convert.FromBase64String(responseSignBase64));

    if (signData.ToLowerInvariant() == responseSign.ToLowerInvariant())
    {
        //обробка результату виконання jsonResponseData
        var dataDynamic1 = js1.Deserialize<dynamic>(jsonResponseData);
        File.WriteAllBytes("c:\\temp\\input\\" + dataDynamic1["DocumentFileName"],
            Convert.FromBase64String(dataDynamic1["DocumentBody"]));
    }
    else
    {
        //помилка верифікації даних
    }
}
}
```

Опис функції вивантаження ZIP архіву документа з підписами

URL, на який необхідно відправляти форму для вивантаження ZIP архіву: <https://my.art-office.com.ua/service/getdocumentzip>.

Формат JSON рядку, що передається в поле **data**:

```
{"DocumentId": Ідентифікатор документа(число)}
```

Формат JSON рядку результату виконання функції:

```
{
    "DocumentFileName": "Назва файлу документа",
    "DocumentBody": "....." //тіло zip архіву base64
}
```

Приклад виклику функції на мові програмування C#:

```
var dataJson = @"{"DocumentId":1000}";

var dataBytes = Encoding.UTF8.GetBytes(dataJson);
var dataBase64 = Convert.ToBase64String(dataBytes);
var client = new RestClient("https://my.art-office.com.ua/service/getdocumentzip");
var request = new RestRequest(Method.POST);

request.AddParameter("data", dataBase64);
var signData = GetHash(dataJson, "79A3E89B-ECFC-4686-8122-F973148B4493");
var cert = File.ReadAllBytes("c:\\temp\\1.cer");
var sign = "{" + string.Format(@"{"UserCertificate": "{1}", "Sign": "{0}"", signData,
                               Convert.ToBase64String(cert)) + "}";
var signBytes = Encoding.UTF8.GetBytes(sign);
var signBase64 = Convert.ToBase64String(signBytes);
request.AddParameter("sign", signBase64);
var restResponse = (RestResponse)client.Execute(request);
var response = restResponse.Content;
if (response.IndexOf("Error", StringComparison.InvariantCulture) > -1)
{
    //обробка помилки {"Error" : "опис помилки"}
}
else
{
    JavaScriptSerializer js = new JavaScriptSerializer();
    var dataDynamic = js.Deserialize<dynamic>(restResponse);
    dataBase64 = dataDynamic["data"];
    dataBytes = Convert.FromBase64String(dataBase64);
    var jsonResponseData = Encoding.UTF8.GetString(dataBytes);

    signData = GetHash(jsonResponseData, "79A3E89B-ECFC-4686-8122-F973148B4493");
    var responseSignBase64 = dataDynamic["sign"];
    var responseSign =
        Encoding.UTF8.GetString(Convert.FromBase64String(responseSignBase64));

    if (signData.ToLowerInvariant() == responseSign.ToLowerInvariant())
    {
        //обробка результату виконання jsonResponseData
        var dataDynamic1 = js1.Deserialize<dynamic>(jsonResponseData);
        File.WriteAllBytes("c:\\temp\\" + dataDynamic1["DocumentFileName"],
            Convert.FromBase64String(dataDynamic1["DocumentBody"]));
    }
    else
    {
        //помилка верифікації даних
    }
}
}
```

Опис функції отримання статусів документів

URL, на який необхідно відправляти форму: <https://my.art-office.com.ua/service/getdocumentsinfo>.

Формат JSON рядку, що передається в поле **data**:

```
{"Documents": [ //масив
```

```

    {"DocumentId": <Ідентифікатор документа 1(число)>},
    .....
    {"DocumentId": <Ідентифікатор документа N (число)>}
  ],
  "IsPrivsInclude": false, //Ознака, чи повертати дані про можливість виконувати дії над
  документом }

```

Формат JSON рядку результату виконання функції:

```

{
  "Documents": [ //масив
    {
      "DocumentId": ідентифікатор документа (число),
      "Name": "назва документа",
      "AuthorName": "ПІБ автора",
      "TypeId": Ідентифікатор виду документа (число),
      "TypeName": "Назва виду документа",
      "FileTypeId": Ідентифікатор типу файлу (число),
      "FileTypeName": "Назва типу файлу",
      "InsertDate": "Дата завантаження документа в систему(строка)",
      "StateId": Ідентифікатор стану документа (число),
      "StateName": "Назва стану документа",
      "ChangeSignStateDate": "Дата зміни стану підписання документа",
      "SignStateId": Ідентифікатор стану підписання (число),
      "SignStateName": "Назва стану підписання",
      "ChangeStatusDate": "Дата зміни статусу документа",
      "IsAched": false, //Ознака архівного документа
      "IsDeleted": false, //Ознака видаленого документа
      "DocNumber": "номер документа",
      "DocDate": "Дата документа(строка)",
      "AgreementNumber": "Номер договору",
      "AgreementDate": "Дата договору(строка)",
      "SumWithPdv": Сума з ПДВ (число),
      "SumWoPdv": Сума без ПДВ (число),
      "SumPdv": Сума ПДВ (число),
      "ContragentCode": "Код контрагента",
      "AuthorCode": "Код автора",
      "IsPrimary": true, //ознака первинного документа
      "CustomData": "додаткова інформація",
      "Reason": "Причина відхилення, якщо документ у відхиленому стані",
      "ErrorMessage": "повідомлення про помилку",
      "Signatures": [ //масив
        {
          "DRFOCode": "Код за ДРФО",
          "EDRPOUCode": "Код за ЄДРПОУ",
          "SignType": "Тип слоту (1 – слот підпису, 2 – слот печатки)",
          "IsSigned": false, //Ознака слот вже підписано
        },
        .....
      ],
      "Privileges": //доступні дії з документом
      {
        "CanSign": true, //Ознака можливості підписання
        "CanSignStamp": false, //Ознака можливості підписання печаткою
        "CanDelete": true, //Ознака можливості видалення
        "CanSaveZipWithSigns": false, //Ознака можливості завантаження ZIP з
        підписами
        "CanDownload": true, //Ознака можливості завантаження
        "CanDecline": true //Ознака можливості відхилення
      }
    }
  ]
}

```

```

    },
    .....
]
}

```

Приклад виклику функції мовою програмування C#:

```

var dataJson = @"{[{"DocumentId":998}, {"DocumentId":999}, {"DocumentId":1000}]}";

var dataBytes = Encoding.UTF8.GetBytes(dataJson);
var dataBase64 = Convert.ToBase64String(dataBytes);
var client = new RestClient("https://my.art-office.com.ua/service/getdocumentsinfo");
var request = new RestRequest(Method.POST);

request.AddParameter("data", dataBase64);
var signData = GetHash(dataJson, "79A3E89B-ECFC-4686-8122-F973148B4493");
var cert = File.ReadAllBytes("c:\\temp\\1.cer");
var sign = "{" + string.Format(@"""UserCertificate"": ""{1}""", ""Sign"": ""{0}""", signData,
    Convert.ToBase64String(cert)) + "}";
var signBytes = Encoding.UTF8.GetBytes(sign);
var signBase64 = Convert.ToBase64String(signBytes);
request.AddParameter("sign", signBase64);
var restResponse = (RestResponse)client.Execute(request);
var response = restResponse.Content;
if (response.IndexOf("Error", StringComparison.InvariantCulture) > -1)
{
    //обробка помилки {"Error" : "опис помилки"}
}
else
{
    JavaScriptSerializer js = new JavaScriptSerializer();
    var dataDynamic = js.Deserialize<dynamic>(restResponse);
    dataBase64 = dataDynamic["data"];
    dataBytes = Convert.FromBase64String(dataBase64);
    var jsonResponseData = Encoding.UTF8.GetString(dataBytes);

    signData = GetHash(jsonResponseData, "79A3E89B-ECFC-4686-8122-F973148B4493");
    var responseSignBase64 = dataDynamic["sign"];
    var responseSign =
    Encoding.UTF8.GetString(Convert.FromBase64String(responseSignBase64));

    if (signData.ToLowerInvariant() == responseSign.ToLowerInvariant())
    {
        //обробка результату виконання jsonResponseData
    }
    else
    {
        //помилка верифікації даних
    }
}
}

```

Опис функції отримання статусів документів з параметрами

URL, на який необхідно відправляти форму:

<https://my.art-office.com.ua/service/getdocumentsinfobyparams>.

Формат JSON рядку, що передається в поле **data**:

```
{
  "Filter": {
    "StartDate": "30.08.2017", //початкова дата у форматі dd.MM.yyyy
    "EndDate": "04.09.2017", //кінцева дата у форматі dd.MM.yyyy
    "ChangeSignStateDate": "10.09.2017 09:23:51", //дата, з якої змінився стан
    підписання документа у форматі dd.MM.yyyy HH:mm:ss
    "IgnoreDeleted": true, //ознака ігнорувати видалені документи
    "IsPrimary": true, //ознака шукати первинні документи
    "DocDirectionType": 0, //Тип направлення документа (0 - всі, 1 - вихідні,
    2 - вхідні)
    "IsNeedCustomData": false, //Ознака, чи повертати додаткову інформацію про
    документ
    "IsPrivsInclude": false, //Ознака, чи повертати дані про можливість виконувати
    дії над документом
    "ChangeStatusDate": "10.09.2017 09:23:51", //дата, з якої змінився стан
    документа у форматі dd.MM.yyyy HH:mm:ss
    "DocTypes": [{ //масив видів документів
      "Id": 1 //ідентифікатор виду
    }],
    .....
    {
      "Id": N
    }],
    "DocStates": [{ //масив станів документів
      "Id": 1 //ідентифікатор стану
    }],
    .....
    {
      "Id": N
    }],
    "DocSignStates": [{ //масив станів підпису документів
      "Id": 3 //ідентифікатор стану підпису
    }],
    .....
    {
      "Id": N
    }],
    "DocDatePeriodStart": "30.08.2017", //Поочаткова дата періоду, для поля
    "Дата документа" у форматі dd.MM.yyyy
    "DocDatePeriodEnd": "04.09.2017", //Кінцева дата періоду, для поля "Дата
    документа"
    "ContragentCode": "12345678" //Код ЄДРПОУ контрагента
  }
}
```

Формат JSON рядку результату виконання функції:

```
{
  "ServerDate": "Серверна дата синхронізації (строка)",
  "Documents": [ //масив
  {
    "DocumentId": ідентифікатор документа (число),
    "Name": "назва документа",
    "AuthorName": "ПІБ автора",
    "TypeId": Ідентифікатор виду документа (число),
    "TypeName": "Назва виду документа",
    "FileTypeId": Ідентифікатор типу файлу (число),
    "FileTypeName": "Назва типу файлу",
    "InsertDate": "Дата завантаження документа в систему(строка)",
  }
]
```



```

"StateId": Ідентифікатор стану документа (число),
"StateName": "Назва стану документа",
"ChangeSignStateDate": "Дата зміни стана підписання документа",
"ChangeStatusDate": "Дата зміни статусу документа",
"SignStateId": Ідентифікатор стану підписання (число),
"SignStateName": "Назва стану підписання",
"IsAchived": false, //Ознака архівного документа
"IsDeleted": false, //Ознака видаленого документа
"DocNumber": "номер документа",
"DocDate": "Дата документа(строка)",
"AgreementNumber": "Номер договору",
"AgreementDate": "Дата договору(строка)",
"SumWithPdv": Сума з ПДВ (число),
"SumWoPdv": Сума без ПДВ (число),
"SumPdv": Сума ПДВ (число),
"ContragentCode": "Код контрагента",
"AuthorCode": "Код автора",
"IsPrimary": true, //ознака первинного документа
"CustomData": "додаткова інформація",
"Reason": "Причина відхилення, якщо документ у відхиленому стані",
"ErrorMessage": "повідомлення про помилку",
"Signatures": [ //масив
{
    "DRFOCode": "Код за ДРФО",
    "EDRPOUCode": "Код за ЄДРПОУ",
    "SignType": "Тип слоту (1 - слот підпису, 2 - слот печатки)",
    "IsSigned": false //Ознака слот вже підписано
},
.....
],
"Privileges": //доступні дії з документом
{
    "CanSign": true, //Ознака можливості підписання
    "CanSignStamp": false, //Ознака можливості підписання печаткою
    "CanDelete": true, //Ознака можливості видалення
    "CanSaveZipWithSigns": false, //Ознака можливості завантаження ZIP з
підписами
    "CanDownload": true, //Ознака можливості завантаження
    "CanDecline": true //Ознака можливості відхилення
}
},
.....
]
}

```

Приклад виклику функції мовою програмування C#:

```

var dataJson = @"{"Filter": {"StartDate": "30.08.2017", "EndDate": "04.09.2017",
    "ChangeSignStateDate": "04.09.2017 09:23:51",
    "IgnoreDeleted": true, "ChangeStatusDate": "04.09.2017 09:23:51"},
"DocTypes": [{"Id": 1}], "DocSignStates": [{"Id": 3}]}";
var dataBytes = Encoding.UTF8.GetBytes(dataJson);
var dataBase64 = Convert.ToBase64String(dataBytes);
var client = new RestClient("https://my.art-office.com.ua/service/getdocumentsinfobyparams");
var request = new RestRequest(Method.POST);

request.AddParameter("data", dataBase64);
var signData = GetHash(dataJson, "79A3E89B-ECFC-4686-8122-F973148B4493");
var cert = File.ReadAllBytes("c:\\temp\\1.cer");

```

```
var sign = "{" + string.Format(@"""UserCertificate"": ""{1}"";""Sign"":""{0}""", signData,
                               Convert.ToBase64String(cert)) + "}";
var signBytes = Encoding.UTF8.GetBytes(sign);
var signBase64 = Convert.ToBase64String(signBytes);
request.AddParameter("sign", signBase64);
var restResponse = (RestResponse)client.Execute(request);
var response = restResponse.Content;
if (response.IndexOf("Error", StringComparison.InvariantCulture) > -1)
{
    //обробка помилки {"Error" : "опис помилки"}
}
else
{
    JavaScriptSerializer js = new JavaScriptSerializer();
    var dataDynamic = js.Deserialize<dynamic>(restResponse);
    dataBase64 = dataDynamic["data"];
    dataBytes = Convert.FromBase64String(dataBase64);
    var jsonResponseData = Encoding.UTF8.GetString(dataBytes);

    signData = GetHash(jsonResponseData, "79A3E89B-ECFC-4686-8122-F973148B4493");
    var responseSignBase64 = dataDynamic["sign"];
    var responseSign =
        Encoding.UTF8.GetString(Convert.FromBase64String(responseSignBase64));

    if (signData.ToLowerInvariant() == responseSign.ToLowerInvariant())
    {
        //обробка результату виконання jsonResponseData
    }
    else
    {
        //помилка верифікації даних
    }
}
}
```

Опис функції публікації документа

URL, на який необхідно відправляти форму: <https://my.art-office.com.ua/service/publishdocument>.

Формат JSON рядку, що передається в поле **data**:

```
{"DocumentId": Ідентифікатор документа(число)}
```

Формат JSON рядку результату виконання функції:

```
{
    "Result": "OK"
}
```

Приклад виклику функції мовою програмування C#:

```
var dataJson = @"{"DocumentId":1000}";
var dataBytes = Encoding.UTF8.GetBytes(dataJson);
var dataBase64 = Convert.ToBase64String(dataBytes);
var client = new RestClient("https://my.art-office.com.ua/service/publishdocument");
```

```

var request = new RestRequest(Method.POST);

request.AddParameter("data", dataBase64);
var signData = GetHash(dataJson, "79A3E89B-ECFC-4686-8122-F973148B4493");
var cert = File.ReadAllBytes("c:\\temp\\1.cer");
var sign = "{" + string.Format(@"""UserCertificate"": ""{1}""", ""Sign"": ""{0}""", signData,
                               Convert.ToBase64String(cert)) + "}";
var signBytes = Encoding.UTF8.GetBytes(sign);
var signBase64 = Convert.ToBase64String(signBytes);
request.AddParameter("sign", signBase64);
var restResponse = (RestResponse)client.Execute(request);
var response = restResponse.Content;
if (response.IndexOf("Error", StringComparison.InvariantCulture) > -1)
{
    //обробка помилки {"Error" : "опис помилки"}
}
else
{
    JavaScriptSerializer js = new JavaScriptSerializer();
    var dataDynamic = js.Deserialize<dynamic>(restResponse);
    dataBase64 = dataDynamic["data"];
    dataBytes = Convert.FromBase64String(dataBase64);
    var jsonResponseData = Encoding.UTF8.GetString(dataBytes);

    signData = GetHash(jsonResponseData, "79A3E89B-ECFC-4686-8122-F973148B4493");
    var responseSignBase64 = dataDynamic["sign"];
    var responseSign =
        Encoding.UTF8.GetString(Convert.FromBase64String(responseSignBase64));

    if (signData.ToLowerInvariant() == responseSign.ToLowerInvariant())
    {
        //операція виконана успішно
    }
    else
    {
        //помилка верифікації даних
    }
}
}

```

Опис функції відхилення документа

URL, на який необхідно відправляти форму: <https://my.art-office.com.ua/service/declinedocument>.

Формат JSON рядку, що передається в поле **data**:

```

{
  "DocumentId": Ідентифікатор документа(число),
  "CauseState": Причина відхилення документа (строка)
}

```

Формат JSON рядку результату виконання функції:

```

{
  "Result": "OK"
}

```

Приклад виклику функції мовою програмування C#:

```

var dataJson = @"{"DocumentId":1000, "CauseState":"Текст причини відхилення"}";

```

```
var dataBytes = Encoding.UTF8.GetBytes(dataJson);
var dataBase64 = Convert.ToBase64String(dataBytes);
var client = new RestClient("https://my.art-office.com.ua/service/declinedocument");
var request = new RestRequest(Method.POST);

request.AddParameter("data", dataBase64);
var signData = GetHash(dataJson, "79A3E89B-ECFC-4686-8122-F973148B4493");
var cert = File.ReadAllBytes("c:\\temp\\1.cer");
var sign = "{" + string.Format(@"""UserCertificate"": ""{1}""", ""Sign"": ""{0}""", signData,
                               Convert.ToBase64String(cert)) + "}";
var signBytes = Encoding.UTF8.GetBytes(sign);
var signBase64 = Convert.ToBase64String(signBytes);
request.AddParameter("sign", signBase64);
var restResponse = (RestResponse)client.Execute(request);
var response = restResponse.Content;
if (response.IndexOf("Error", StringComparison.InvariantCulture) > -1)
{
    //обробка помилки {"Error" : "опис помилки"}
}
else
{
    JavaScriptSerializer js = new JavaScriptSerializer();
    var dataDynamic = js.Deserialize<dynamic>(restResponse);
    dataBase64 = dataDynamic["data"];
    dataBytes = Convert.FromBase64String(dataBase64);
    var jsonResponseData = Encoding.UTF8.GetString(dataBytes);

    signData = GetHash(jsonResponseData, "79A3E89B-ECFC-4686-8122-F973148B4493");
    var responseSignBase64 = dataDynamic["sign"];
    var responseSign =
        Encoding.UTF8.GetString(Convert.FromBase64String(responseSignBase64));

    if (signData.ToLowerInvariant() == responseSign.ToLowerInvariant())
    {
        //операція виконана успішно
    }
    else
    {
        //помилка верифікації даних
    }
}
}
```

Опис функції видалення документа

URL, на який необхідно відправляти форму: <https://my.art-office.com.ua/service/deletedocument> .

Формат JSON рядку, що передається в поле **data**:

```
{
  "DocumentId": Ідентифікатор документа(число)
}
```

Формат JSON рядку результату виконання функції:

```
{
  "Result": "OK"
}
```

Приклад виклику функції мовою програмування C#:

```
var dataJson = @"{"DocumentId":1000}";

var dataBytes = Encoding.UTF8.GetBytes(dataJson);
var dataBase64 = Convert.ToBase64String(dataBytes);
var client = new RestClient("https://my.art-office.com.ua/service/deletedocument");
var request = new RestRequest(Method.POST);

request.AddParameter("data", dataBase64);
var signData = GetHash(dataJson, "79A3E89B-ECFC-4686-8122-F973148B4493");
var cert = File.ReadAllBytes("c:\\temp\\1.cer");
var sign = "{" + string.Format(@"""UserCertificate"": ""{1}"";""Sign"":""{0}""", signData,
                               Convert.ToBase64String(cert)) + "}";
var signBytes = Encoding.UTF8.GetBytes(sign);
var signBase64 = Convert.ToBase64String(signBytes);
request.AddParameter("sign", signBase64);
var restResponse = (RestResponse)client.Execute(request);
var response = restResponse.Content;
if (response.IndexOf("Error", StringComparison.InvariantCulture) > -1)
{
    //обробка помилки {"Error" : "опис помилки"}
}
else
{
    JavaScriptSerializer js = new JavaScriptSerializer();
    var dataDynamic = js.Deserialize<dynamic>(restResponse);
    dataBase64 = dataDynamic["data"];
    dataBytes = Convert.FromBase64String(dataBase64);
    var jsonResponseData = Encoding.UTF8.GetString(dataBytes);

    signData = GetHash(jsonResponseData, "79A3E89B-ECFC-4686-8122-F973148B4493");
    var responseSignBase64 = dataDynamic["sign"];
    var responseSign =
        Encoding.UTF8.GetString(Convert.FromBase64String(responseSignBase64));

    if (signData.ToLowerInvariant() == responseSign.ToLowerInvariant())
    {
        //операція виконана успішно
    }
    else
    {
        //помилка верифікації даних
    }
}
}
```

Опис функції коментування документа

URL, на який необхідно відправляти форму: <https://my.art-office.com.ua/service/addcomment>.

Формат JSON рядку, що передається в поле **data**:

```
{
    "DocumentId": Ідентифікатор документа(число),
    "CommentText": "Тіло коментаря"
}
```

Формат JSON рядку результату виконання функції:

```
{
    "Result": "OK"
}
```

Приклад виклику функції мовою програмування C#:

```
var dataJson = @"{"DocumentId":1000, "CommentText": "Тіло коментаря"}";

var dataBytes = Encoding.UTF8.GetBytes(dataJson);
var dataBase64 = Convert.ToBase64String(dataBytes);
var client = new RestClient("https://my.art-office.com.ua/service/addcomment");
var request = new RestRequest(Method.POST);

request.AddParameter("data", dataBase64);
var signData = GetHash(dataJson, "79A3E89B-ECFC-4686-8122-F973148B4493");
var cert = File.ReadAllBytes("c:\\temp\\1.cer");
var sign = "{" + string.Format(@"""UserCertificate"": ""{1}""""Sign"":""{0}""""", signData,
    Convert.ToBase64String(cert)) + "}";
var signBytes = Encoding.UTF8.GetBytes(sign);
var signBase64 = Convert.ToBase64String(signBytes);
request.AddParameter("sign", signBase64);
var restResponse = (RestResponse)client.Execute(request);
var response = restResponse.Content;
if (response.IndexOf("Error", StringComparison.InvariantCulture) > -1)
{
    //обробка помилки {"Error" : "опис помилки"}
}
else
{
    JavaScriptSerializer js = new JavaScriptSerializer();
    var dataDynamic = js.Deserialize<dynamic>(restResponse);
    dataBase64 = dataDynamic["data"];
    dataBytes = Convert.FromBase64String(dataBase64);
    var jsonResponseData = Encoding.UTF8.GetString(dataBytes);

    signData = GetHash(jsonResponseData, "79A3E89B-ECFC-4686-8122-F973148B4493");
    var responseSignBase64 = dataDynamic["sign"];
    var responseSign =
        Encoding.UTF8.GetString(Convert.FromBase64String(responseSignBase64));

    if (signData.ToLowerInvariant() == responseSign.ToLowerInvariant())
    {
        //операція виконана успішно
    }
    else
    {
        //помилка верифікації даних
    }
}
}
```

Опис функції отримання коментарів для документа

URL, на який необхідно відправляти форму: <https://my.art-office.com.ua/service/getdocumentcomments>

Формат JSON рядку, що передається в поле **data**:

```
{"DocumentId": <Ідентифікатор документа (число)>}
```

Формат JSON рядку результату виконання функції:

```

{
  "DocumentId": ідентифікатор документа (число),
  "Comments": [ //масив
  {
    "CommentText": "тіло коментаря",
    "InsertDate": "Дата створення (строка)",
    "AuthorName": "ПІБ автора",
    "AuthorDrfo": "ДРФО автора",
    "AuthorOrgName": "Назва організації автора",
    "AuthorOrgEdrpou": "ЄДРПОУ організації автора",
    "AuthorSubjectType": Тип суб'єкта автора,
  },
  .....
  ]}

```

Приклад виклику функції мовою програмування C#:

```

var dataJson = @"{"DocumentId":998}";

var dataBytes = Encoding.UTF8.GetBytes(dataJson);
var dataBase64 = Convert.ToBase64String(dataBytes);
var client = new RestClient("https://my.art-office.com.ua/service/getdocumentcomments");
var request = new RestRequest(Method.POST);

request.AddParameter("data", dataBase64);
var signData = GetHash(dataJson, "79A3E89B-ECFC-4686-8122-F973148B4493");
var cert = File.ReadAllBytes("c:\\temp\\1.cer");
var sign = "{" + string.Format(@"""UserCertificate"": ""{1}""", ""Sign"":""{0}""", signData,
                               Convert.ToBase64String(cert)) + "}";
var signBytes = Encoding.UTF8.GetBytes(sign);
var signBase64 = Convert.ToBase64String(signBytes);
request.AddParameter("sign", signBase64);
var restResponse = (RestResponse)client.Execute(request);
var response = restResponse.Content;
if (response.IndexOf("Error", StringComparison.InvariantCulture) > -1)
{
  //обробка помилки {"Error" : "опис помилки"}
}
else
{
  JavaScriptSerializer js = new JavaScriptSerializer();
  var dataDynamic = js.Deserialize<dynamic>(restResponse);
  dataBase64 = dataDynamic["data"];
  dataBytes = Convert.FromBase64String(dataBase64);
  var jsonResponseData = Encoding.UTF8.GetString(dataBytes);

  signData = GetHash(jsonResponseData, "79A3E89B-ECFC-4686-8122-F973148B4493");
  var responseSignBase64 = dataDynamic["sign"];
  var responseSign =
  Encoding.UTF8.GetString(Convert.FromBase64String(responseSignBase64));

  if (signData.ToLowerInvariant() == responseSign.ToLowerInvariant())
  {
    //обробка результату виконання jsonResponseData
  }
  else
  {

```

```

        //помилка верифікації даних
    }
}

```

Опис функції трекінгу коментарів

URL, на який необхідно відправляти форму: <https://my.art-office.com.ua/service/trackcomments>

Формат JSON рядку, що передається в поле **data**:

```

{
    "Filter": {
        "ChangeStatusDate": "10.09.2017 09:23:51" //дата, останнього трекінгу у
        форматі dd.MM.yyyy HH:mm:ss
    }
}

```

Формат JSON рядку результату виконання функції:

```

{
    "Comments": [ //масив
        {
            "DocumentId": ідентифікатор документа (число),
            "CommentText": "тіло коментаря",
            "InsertDate": "Дата створення (строка)"
        },
        .....
    ],
    "ChangeStatusDate": "серверна дата синхронізації"
}

```

Приклад виклику функції мовою програмування C#:

```

var dataJson = @"{"Filter": {"ChangeStatusDate": "10.09.2017 09:23:51"}}";
var dataBytes = Encoding.UTF8.GetBytes(dataJson);
var dataBase64 = Convert.ToBase64String(dataBytes);
var client = new RestClient("https://my.art-office.com.ua/service/trackcomments");
var request = new RestRequest(Method.POST);

request.AddParameter("data", dataBase64);
var signData = GetHash(dataJson, "79A3E89B-ECFC-4686-8122-F973148B4493");
var cert = File.ReadAllBytes("c:\\temp\\1.cer");
var sign = "{" + string.Format(@"""UserCertificate"": ""{1}""""Sign"":""{0}""""", signData,
    Convert.ToBase64String(cert)) + "}";
var signBytes = Encoding.UTF8.GetBytes(sign);
var signBase64 = Convert.ToBase64String(signBytes);
request.AddParameter("sign", signBase64);
var restResponse = (RestResponse)client.Execute(request);
var response = restResponse.Content;
if (response.IndexOf("Error", StringComparison.InvariantCulture) > -1)
{
    //обробка помилки {"Error" : "опис помилки"}
}
else
{
    JavaScriptSerializer js = new JavaScriptSerializer();
    var dataDynamic = js.Deserialize<dynamic>(restResponse);
    dataBase64 = dataDynamic["data"];
}

```



```
dataBytes = Convert.FromBase64String(dataBase64);
var jsonResponseData = Encoding.UTF8.GetString(dataBytes);

signData = GetHash(jsonResponseData, "79A3E89B-ECFC-4686-8122-F973148B4493");
var responseSignBase64 = dataDynamic["sign"];
var responseSign =
Encoding.UTF8.GetString(Convert.FromBase64String(responseSignBase64));

if (signData.ToLowerInvariant() == responseSign.ToLowerInvariant())
{
    //обробка результату виконання jsonResponseData
}
else
{
    //помилка верифікації даних
}
}
```

Опис функції створення зв'язку між документами

URL, на який необхідно відправляти форму: <https://my.art-office.com.ua/service/addlink>.

Формат JSON рядку, що передається в поле **data**:

```
{
    "DocumentId": 11, //ідентифікатор першого документа (число)
    "LinkedDocId": 12, //ідентифікатор другого документа (число)
    "LinkTypeId": 1 //тип зв'язку з табл. «Типи зв'язків між документами» (число)
}
```

Формат JSON рядку результату виконання функції:

```
{
    "Result": "OK"
}
```

Опис функції видалення зв'язку між документами

URL, на який необхідно відправляти форму: <https://my.art-office.com.ua/service/removelink>.

Формат JSON рядку, що передається в поле **data**:

```
{
    "Id": 151, //ідентифікатор зв'язку між документами (число)
    "DocumentId": 11 //ідентифікатор документа (число)
}
```

Формат JSON рядку результату виконання функції:

```
{
    "Result": "OK"
}
```

Опис функції отримання зв'язків для документа

URL, на який необхідно відправляти форму: <https://my.art-office.com.ua/service/getdocumentlinks>.

Формат JSON рядку, що передається в поле **data**:

```
{
  "DocumentId": 11 //ідентифікатор документа (число)
}
```

Формат JSON рядку результату виконання функції:

```
{
  "Links": [ //масив зв'язків
    {
      "Id": 151, //ідентифікатор зв'язку між документами (число)
      "DocumentId": 11, //ідентифікатор документа (число)
      "LinkedDocId": 12, //ідентифікатор другого документа (число)
      "LinkId": 1, //тип зв'язку з т. «Типи зв'язків між документами» (число)
      "LinkIdForDoc": 1, //тип зв'язку стосовно даного документа з т. «Типи зв'язків між документами» (число)
      "TypeName": "Є додатком до", //опис типу зв'язку
      "DisplayName": "Документ 1", //назва зв'язаного документа
      "AuthorName": "Іванов Іван Іванович", //ПІБ особи, що створила зв'язок
      "CanDelete": true //Ознака можливості видалення
      "ShareDeleted": false //Ознака що зв'язаний документ більше недоступний
    },
    .....
  ]
}
```

Опис функції трекінгу кількості зв'язків для документа

URL, на який необхідно відправляти форму: <https://my.art-office.com.ua/service/trackdoclinkscount>

Формат JSON рядку, що передається в поле **data**:

```
{
  "Filter": {
    "ChangeStatusDate": "10.09.2017 09:23:51" //дата, останнього трекінгу у форматі dd.MM.yyyy HH:mm:ss
  }
}
```

Формат JSON рядку результату виконання функції:

```
{
  "DocLinksCountList": [ //масив
    {
      "DocumentId": 11, //ідентифікатор документа (число)
      "LinksCount": 2 //кількість коментарів (число)
    },
    .....
  ],
  "ChangeStatusDate": "серверна дата синхронізації"
}
```

Опис функції збереження зображень факсиміле підпису та печатки

URL, на який необхідно відправляти форму:

<https://my.art-office.com.ua/service/changesignimage>.

Формат JSON рядку, що передається в поле **data**:

```
{
  "ImageBase64String": ".....", //зображення у форматі base64
  "IsStamp": false //ознака печатки
}
```

Формат JSON рядку результату виконання функції:

```
{
  "Result": "OK"
}
```

Опис функції вивантаження зображень факсиміле підпису та печатки

URL, на який необхідно відправляти форму:

<https://my.art-office.com.ua/service/getsignimages>.

Формат JSON рядку, що передається в поле **data**:

```
{
  "Salt": "dfjhgker erjgklene" //будь-яка соль
}
```

Формат JSON рядку результату виконання функції:

```
{
  "SignImages": [ //масив зображень
    {
      "ImageBase64String": ".....", //зображення підпису у форматі base64
      "IsStamp": false //ознака печатки
    },
    {
      "ImageBase64String": ".....", //зображення печатки у форматі base64
      "IsStamp": true //ознака печатки
    }
  ]
}
```

Словники

Таблиця «Типи файлів»

Ідентифікатор	Опис
1	Файли з розширенням docx (MSWord)
2	Файли з розширенням xlsx (MSExcel)
3	Файли з розширенням pdf
4	Файли з розширенням xml
5	Файли з розширенням dbf
6	Файли з розширенням txt
7	Файли з розширенням png
8	Файли з розширенням jpeg
9	Файли з розширенням zip
10	Файли з розширенням doc
11	Файли з розширенням xls
12	Файли з розширенням rtf

Таблиця «Види документів»

Ідентифікатор	Опис	Доступні групи підписантів
1	Акт	Виконавець, Замовник
2	Договір	Виконавець, Замовник Або Сторона 1, Сторона 2 Або Сторона 1, Сторона 2, Сторона 3 Залежить від PolicyId
3	Додаткова угода	Виконавець, Замовник
4	Додаток до договору	Виконавець, Замовник
5	Рахунок	Виконавець
6	Заява	Автор / Заявник
7	Лист	Автор / Заявник
8	Накладна	Виконавець, Замовник
9	Довідка	Автор / Заявник
10	Листок опитування	Ініціатор, Учасник(тільки docx)
11	Акт звіряння	Сторона 1, Сторона 2
12	Довіреність	Автор / Заявник
13	eТТН	Відправник, Перевізник, Експедитор, Отримувач Або Відправник, Перевізник, Отримувач Залежить від PolicyId

Таблиця «Групи підписантів»

Ідентифікатор	Опис
1	Виконавець
2	Замовник

3	Автор / Заявник
4	Сторона 1
5	Сторона 2
6	Відправник
7	Перевізник
8	Експедитор
9	Отримувач
10	Автор
11	Ініціатор
12	Учасник
13	Сторона 3

Таблиця «Типи контрагентів»

Ідентифікатор	Опис
1	Юридична особа
2	Фізична особа
3	Фізична особа-підприємець

Таблиця «Типи печатки»

Ідентифікатор	Опис
1	Печатка обов'язкова
2	Печатка відсутня

Таблиця «Стан документа»

Ідентифікатор	Опис
1	Опублікований
2	Чернетка
4	Відхилений
5	Очікує погодження
6	Непогоджений

Таблиця «Стан підписання»

Ідентифікатор	Опис
1	Без політики
2	Не підписаний
3	Підписаний, але не повністю
4	Підписаний повністю
5	Не валідний

Таблиця «Типи доступу»

Ідентифікатор	Опис
2	Фізична особа
3	Організація/ФОП

Таблиця «Ідентифікатори шаблонів первинних документів»

Ідентифікатор	Опис
3	Акт (без ПДВ)
5	Акт (ціна без ПДВ)

6	Акт (ціна з ПДВ)
10	Видаткова накладна (без ПДВ)
11	Видаткова накладна (ціна без ПДВ)
12	Видаткова накладна (ціна з ПДВ)
13	Рахунок на оплату (без ПДВ)
14	Рахунок на оплату (ціна без ПДВ)
15	Рахунок на оплату (ціна з ПДВ)
17	Акт звірення

Таблиця «Типи зв'язків між документами»

Ідентифікатор	Опис
1	Є додатком до
2	Має додаток
3	Скасований документом
4	Скасовує документ
5	Пов'язаний з