My Art-Office API

version 1.2

Зміст

Опис API для інтеграції із зовнішніми системами 3

Загальний опис алгоритму обміну даними 3

Загальні методи 5

- Опис функції збереження зображень факсиміле підпису та печатки 5
- Опис функції вивантаження зображень факсиміле підпису та печатки 5
- Опис функції отримання лічильників к-сті документів на балансі 6
- Опис функції створення співробітника організації (користувача) 6
- Опис функції створення токена для підписання документа за допомогою віджета 7

Методи для роботи з зовнішніми документами (обмін з контрагентами) 9

- Опис функції створення документа 9
- Опис функції вивантаження документа з системи 12
- Опис функції вивантаження ZIP архіву документа з підписами 13
- Опис функції вивантаження підписаного документа у форматі p7s (CoSign) 13
- Опис функції вивантаження підписаного документа у форматі Asic-S 14
- Опис функції створення первинного документа 14
- Опис функції вивантаження XML первинного документа з системи 17
- Опис функції отримання статусів документів 19
- Опис функції отримання статусів документів з параметрами 21
- Опис функції публікації документа 24
- Опис функції відхилення документа 25
- Опис функції видалення документа 26
- Опис функції коментування документа 27
- Опис функції отримання коментарів для документа 28
- Опис функції трекінгу коментарів 29
- Опис функції створення зв’язку між документами 30
- Опис функції видалення зв’язку між документами 31
- Опис функції отримання зв’язків для документа 31

Опис функції трекінгу кількості зв’язків для документа 32
Опис функції отримання інформації про версійність документа 32
Методи для роботи із внутрішніми документами (обмін в межах організації) 33
Опис функції вивантаження словника «Види внутрішніх документів» 33
Опис функції вивантаження словника «Маршрути узгодження та підписання» 33
Опис функції створення внутрішнього документа 34
Опис функції створення внутрішнього документа за ідентифікатором маршрута узгодження та підписання 35
Опис функції видалення внутрішнього документа 37
Опис функції вивантаження внутрішнього документа з системи 38
Опис функції вивантаження ZIP архіву внутрішнього документа з підписами 39
Опис функції вивантаження внутрішнього документа у форматі p7s (CoSign) 39
Опис функції вивантаження внутрішнього документа у форматі Asic-S 40
Опис функції отримання статусів внутрішніх документів 40
Опис функції отримання статусів внутрішніх документів з параметрами 42
Опис функції завантаження нового файлу в не підписаний/не погоджений документ 44
Словники 46
Таблиця «Типи файлів» 46
Таблиця «Види документів» 46
Таблиця «Групи підписантів» 47
Таблиця «Типи контрагентів» 48
Таблиця «Типи печатки» 48
Таблиця «Стан зовнішнього документа» 48
Таблиця «Стан внутрішнього документа» 48
Таблиця «Стан підписання зовнішнього та внутрішнього документа» 48
Таблиця «Типи доступу» 49
Таблиця «Ідентифікатори шаблонів первинних документів» 49
Таблиця «Типи зв’язків між документами» 49

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.artoffice.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 GetHash(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.

Загальні методи

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

		}
	]
}
```

Опис функції отримання лічильників к-сті документів на балансі

URL, на який необхідно відправляти форму:

https://my.art-office.com.ua/service/getdocslicenseinfo .

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

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

Формат JSON рядку результату виконання функції:

```
{
	"PrepaidDocsCount": 1, //кількість оплачених документів
	"FreeDocsCount": 2, //кількість безкоштовних документів
}
```

Опис функції створення співробітника організації (користувача)

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

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

```
{
	"Email": "test@test.ua",		//Обов'язкове для заповнення
	"EmployeeIpn": "1234567890",		//Код ІПН користувача, Обов'язкове
	"EmployeeName": "Петренко Іван Тестович",
					//ПІБ користувача, Обов'язкове
	"IsAllowAccessFoKey": true,		// Дозволяти авторизацію ключами ФО в кабінет ЮО
	"IsInExtDocRole": true,			//   Ознака   наявності   ролі   Оператора   ЗЕД
"IsInIntDocRole": true,		// Ознака наявності ролі Оператора ВЕД
	"OrgRoleId": 33,			// ІД організаційної ролі, зі списку підприємства
	"PermissionProfileId": 1,		// ІД профілю доступу для оператора ЗЕД
	"Phone": "+380991234564"		// Телефон співробітника, не обв'язкове
}
```

При заповненні вхідних параметрів необхідно дотримуватись наступних правил:

- Якщо IsInExtDocRole = false, тоді параметр PermissionProfileId не потрібно передавати;
- Поле OrgRoleId не обов'язкове для заповнення;
- Поле IsAllowAccessFoKey регулює, чи можна буде користувачу логінитись в систему ключами фіз особи в кабінет співробітника компанії (Юр особа або ФОП)

Формат JSON рядку результату виконання функції:

```
{ "Result":"OK", "UserId": <Ідентифікатор користувача(число)>}
```

***Примітка**: При виконанні запита для даного метода, обов'язковою умовою є використання сертифіката підприємства, в рамках якого створюється новий співробітник.*

Наприклад:

Якщо ФОП (1234567890) належить організації (11223344) то виконати такий запит, з метою додавання співробітника до організації ключем ФО\ФОП - неможливо.

Запит повинен виконуватись за допомогою ключа співробітника організації (11223344)

Опис функції створення токена для підписання документа за допомогою віджета

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

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

```
{
	"DocumentId": "72553",		//ідентифікатор документа в системі
	"SignerId": "SignerP1"		//ідентифікатор слота підписа в документі
}
```

Формат JSON рядку результату виконання функції:

```
{
	"DocumentId": <Ідентифікатор документа(число)>,
	"SignerId": <Ідентифікатор слотапідписа(строка)>,
	"WidgetSignerKey": <Токен для підписання (гуід)>,
}
```

Приклад виклику функції мовою програмування C#:

```
var doc = File.ReadAllBytes("c:\\temp\\1.docx");
var dataJson = @"
{
		""DocumentId"": 72553,
		""SignerId"": ""contr_org_sign""
}";

var dataBytes = Encoding.UTF8.GetBytes(dataJson); var dataBase64 =
Convert.ToBase64String(dataBytes);
var client = new RestClient(
"https://my.art-office.com.ua/ service/widgetgeneratesigntoken ");
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}"",""UseOrgApiKey"":""{2}""", signData, Convert.ToBase64String(cert) ,
true) + "}";
```

```
// UseOrgApiKey – необов’язковий параметр який визначає який ключ АПІ буде використано
// Ключ АПІ користувача або ключ АПІ організації (зазвичай такі виклики виконуються за
// допомогою сертифіката печатки)
// Якщо UseOrgApiKey =true, запит необхідно виконувати з використанням сертифікату печатки
// та ApiKey вашої організації
// Якщо UseOrgApiKey=false, запит необхідно виконувати з використанням сертифікату підпису
// та ApiKey користувача

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, "SignerId": "contr_org_sign", "WidgetSignerKey": "CB6FAB0B-E97E-4CA2-
B846-0CC746DF6A8A" }
} else
{ //помилка верифікації даних } }
```

Додатково:

- URL, на який необхідно відправляти запит, для отримання форми підписання документа: https://my.art-office.com.ua/Widgets/SignDocument?widgetToken=<Токен для підписання (гуід)>

Методи для роботи з зовнішніми документами (обмін з контрагентами)

Опис функції створення документа

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
		"AuthorNotInSignersList":true,      //ознака того, що автор документа відсутній
                                            в списку підписантів (учасників)
                                            Якщо встановлено, то списання документів
                                            виконується під час створення документа.
                                            Списання документів з підписантів – не
                                            відбувається

		"IsDraft":true,                     //ознака чернетки, якщо встановлено, то
                                            документ не доступний контрагенту, поки
                                            його не опублікують

		"ExtData":{                            //додаткові дані про документ
			"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":                  //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 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}"",""UseOrgApiKey"":""{2}"",""CertificateOwnerDrfo"":""{3}""",
signData, Convert.ToBase64String(cert) , true, "1212121205") + "}";
// UseOrgApiKey – необов'язковий параметр, який визначає який ключ АПІ буде використано
// Ключ АПІ користувача або ключ АПІ організації (зазвичай такі виклики виконуються за
// допомогою сертифіката печатки)
// Якщо UseOrgApiKey =true, запит необхідно виконувати з використанням сертифікату печатки
// та ApiKey вашої організації
// Якщо UseOrgApiKey=false, запит необхідно виконувати з використанням сертифікату підпису
// та ApiKey користувача
// CertificateOwnerDrfo - необов'язковий параметр, який визначає за ДРФО (ІПН) користувача –
// автора, в тому разі коли використовується сертифікат печатки
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 результата 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 ZIP архіву документа з підписами

URL, на який необхідно відправляти форму для вивантаження ZIP архіву: https://my.artoffice.com.ua/service/getdocumentzip.

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

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

Формат JSON рядку результата виконання функції:

```
{
	"DocumentFileName":"Назва файла документа",
	"DocumentBody": "……" //тіло zip архіву base64 }
```

Приклад виклику функції на мові програмування C# у розділі «Опис функції вивантаження документа з системи», змінюється лише URL, на який необхідно відправляти форму.

Опис функції вивантаження підписаного документа у форматі p7s (CoSign)

URL, на який необхідно відправляти форму для вивантаження документа https://my.artoffice.com.ua/service/getdocumentcosign .

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

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

Формат JSON рядку результата виконання функції:

```
{
	"DocumentFileName":"Назва файла документа",
	"DocumentBody": "……." //тіло документа у base64 }
```

Приклад виклику функції на мові програмування C# у розділі «Опис функції вивантаження документа з системи», змінюється лише URL, на який необхідно відправляти форму.

Опис функції вивантаження підписаного документа у форматі Asic-S

URL, на який необхідно відправляти форму для вивантаження документа https://my.artoffice.com.ua/service/getdocumentasics .

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

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

JSON рядку результата виконання функції:

```
{
	"DocumentFileName":"Назва файла документа",
	"DocumentBody": "……." //тіло документа у base64 }
```

Приклад виклику функції на мові програмування C# у розділі «Опис функції вивантаження документа з системи», змінюється лише URL, на який необхідно відправляти форму.

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": "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
       "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\\PD1C8202003455435501000000011112017.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 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 результата 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 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

"IsNeedCustomData": false  //Ознака, чи повертати додаткову інформацію про документ }
```

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",
              "FileName": "Назва файлу",
              "FileExtension": "Розширення 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 статуса документа",
              "DirectionId": Напрямок документа: 1 - Вихідний; 2 - Вхідний,
              "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 котрагента",
              "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
                    "SignerId": "Ідентифікатор підписанта" (Наприклад: org_sign)
             },
             .........
             ],
             "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
		"IsNeedAllOrgs": true, //Ознака, чи повертати документи і по ФОП і по ФО одночасно
  "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		//ідентфікатор виду
		},
		……
		{
			"Id": N
		}],
		"DocStates": [{		//масив станів документів
			"Id": 1		//ідентфікатор стану
		},
		……
		{
			"Id": N
		}],
		"DocSignStates": [{	//масив станів підпису документів
			"Id": 3		//ідентфікатор 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",
              "FileName": "Назва файлу",
              "FileExtension": "Розширення 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 статуса документа",
              "DirectionId": Напрямок документа: 1 - Вихідний; 2 - Вхідний,
              "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 котрагента",
              "AuthorCode": "Код автора",
              "IsPrimary": true, //ознака первинного документа
              "IdOrgType": 1, //тип тип суб’єкта, з таблиці «Типи контрагентів»
              "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
                     "SignerId": "Ідентифікатор підписанта" (Наприклад: org_sign)
              },
              .........
              ],
```

```
            "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 результата 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 результата 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 результата 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 результата 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'яку з табл. «Типи зв'язків між документами» (число) }
```

Формат JSON рядку результата 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 результата 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 результата 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)
		    "LinkTypeId": 1, //тип зв’яку з т. «Типи зв’язків між документами» (число)
		"LinkTypeIdForDoc": 1, //тип зв’яку стосовно даного документа з т. «Типи
		                       зв’язків між документами» (число)
		"TypeName": "Є додатком до", 		//оипс типу зв’яку
		"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 быльше 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 отримання інформації про версійність документа

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

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

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

Формат JSON рядку результата виконання функції:

```
{
	"DocumentVersionsInfo": [  // інформація про версії документа, масив
	{
	    "DocumentID": айді документа в системі
	    "DocDate": дата створення документа
	    "Name": Назва документа
	    "VersionNumber": номер версії документа
	    "Reason": Причина відхилення, якщо документ знаходиться у стані "Відхилено"
	 },
		……
	]
}
```

Методи для роботи із внутрішніми документами (обмін в межах організації)

Опис функції вивантаження словника «Види внутрішніх документів»

URL, на який необхідно відправляти форму:

https://my.art-office.com.ua/service/getinternaldocumenttypes.

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

```
{
	"Salt": "А баба галамага..."	//будь-яка соль }
```

Формат JSON рядку результату виконання функції:

```
{
	[{
		"Id": 1,
		"Name": "Наказ"
	},
	{
		"Id": 2,
		"Name": "Складський документ"
	},
	{
		"Id": 3,
		"Name": "Кадровий документ"
	}] }
```

Опис функції вивантаження словника «Маршрути узгодження та підписання»

URL, на який необхідно відправляти форму:

https://my.art-office.com.ua/service/GetInternalDocumentRoutes.

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

```
{
	"Salt": "А баба галамага..."	//будь-яка соль
}
```

Формат JSON рядку результату виконання функції:

```
{
	[{
		"Id": 1,
		"Name": "Два підписа"
	},
	{
		"Id": 2,
		"Name": "Три підписа"
	},
	{
```

```
            "Id": 3,
            "Name": "Тільки узгодження"
      }] }
```

Опис функції створення внутрішнього документа

URL, на який необхідно відправляти форму:

https:// my.art-office.com.ua/service/createinternaldocument.

При створенні внутрішнього документа необхідно окрім тіла документа також передавати файл маршрута узгодження та підписання документа (файл маршрута можна вивантажити за панелі адміністрування адміністратора організації). Також необхідно передати тип документа (наприклад «Складський документ» або «Наказ») всі наявні типи можна подивитись скориставшись функцією вивантаження словника «Типи внутрішніх документів».
Не обовязковим параметром є CustomData, в якому можна передати довільну додаткову інформацію про документ

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

```
{
      "DocumentData":                          //інформація про документ
      {
            "FileName": "1.docx",              //оригінальна назва файлу
            "TypeId": 1,                       //з таблиці «Види внутрішніх документів»
            "Name": "Акт виконаних робіт",     //назва документа
            "ExtData":
               {
                "CustomData": "abc"  //Будь-яка додаткова інформація про документ
               }
       },

      "DocumentBody": "……",           //тіло документа в base64
      "DocumentRouteBody": "……"       //тіло файла маршрута в base64
}
```

Формат JSON рядку результату виконання функції:

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

Приклад виклику функції мовою програмування C#:

```
var doc = File.ReadAllBytes("c:\\temp\\PD1C8202003455435010000011112017.XML"); var
dataJson = @"
    {
       ""DocumentData"": [{
            ""FileName"": ""1.docx"",
            ""Name"": ""Акт виконаних робіт"",
            ""TypeId"": 1
     },
       ""DocumentBody"": """ + Convert.ToBase64String(doc) + @"""

       ""DocumentRouteBody"": """ + Convert.ToBase64String(route) + @"""
```

```
}";

var dataBytes = Encoding.UTF8.GetBytes(dataJson); var
dataBase64 = Convert.ToBase64String(dataBytes);
var client = new RestClient("https://<назва сайту>/service/createinternaldocument"); 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 внутрішнього документа за ідентифікатором маршрута узгодження та підписання

URL, на який необхідно відправляти форму:

https://my.art-office.com.ua/service/CreateInternalDocumentByRouteId.

При створенні внутрішнього документа необхідно окрім тіла документа також передавати ідентифікатор маршруту узгодження та підписання, який можна подивитись в словнику «Маршрути узгодження та підписання». Також необхідно передати тип документа (наприклад «Складський документ» або «Наказ») всі наявні типи можна подивитись скориставшись функцією вивантаження словника «Типи внутрішніх документів».

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

```
{
	"DocumentData":                          //інформація про документ
	{
		"FileName": "1.docx",            //оригінальна назва файлу
		"TypeId": 1,                     //з таблиці «Види внутрішніх документів»
		"RouteId": 1,                            //з словника «Маршрути узгодження та
підписання»
		"Name": "Акт виконаних робіт"    //назва документа
	}

	"DocumentBody": "……"           //тіло документа в base64
}
```

Формат JSON рядку результату виконання функції:

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

Приклад виклику функції мовою програмування C#:

```
var doc = File.ReadAllBytes("c:\\temp\\PD1C8202003455435501000000111112017.XML"); var
dataJson = @"
    {
       ""DocumentData"": [{
             ""FileName"": ""1.docx"",
             ""Name"": ""Акт виконаних робіт"",
             ""TypeId"": 1,
             ""RouteId"": 1
     },
       ""DocumentBody"": """ + Convert.ToBase64String(doc) + @"""

}";

var dataBytes = Encoding.UTF8.GetBytes(dataJson); var dataBase64 =
Convert.ToBase64String(dataBytes); var client = new RestClient("https://<назва
сайту>/service/CreateInternalDocumentByRouteId "); 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 видалення внутрішнього документа

URL, на який необхідно відправляти форму:

https://my.art-office.com.ua/service/deleteinternaldocument .

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

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

Формат JSON рядку результата 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/deleteinternaldocument"); 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 вивантаження внутрішнього документа з системи

URL, на який необхідно відправляти форму: https://my.art-office.com.ua/service/getinternaldocument . Вивантаження документу може бути виконане користувачем, що є автором документа або перебуває на маршруті обробки документа або має права Адміністратора організації.

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

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

Формат JSON рядку результата виконання функції:

```
{
	"DocumentFileName":"Назва файла",
	"DocumentBody": "……." //тіло файла 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/getinternaldocument"); 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 внутрішнього документа з підписами

URL, на який необхідно відправляти форму для вивантаження ZIP архіву: https://my.artoffice.com.ua/service/getinternaldocumentzip.

Вивантаження документу може бути виконане користувачем, що є автором документа або перебуває на маршруті обробки документа або має права Адміністратора організації.

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

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

Формат JSON рядку результата виконання функції:

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

Приклад виклику функції на мові програмування C# у розділі «Опис функції вивантаження внутрішнього документа з системи», змінюється лише URL, на який необхідно відправляти форму.

Опис функції вивантаження внутрішнього документа у форматі p7s (CoSign)

URL, на який необхідно відправляти форму для вивантаження документа: https://my.artoffice.com.ua/service/getinternaldocumentcosign.

Вивантаження документу може бути виконане користувачем, що є автором документа або перебуває на маршруті обробки документа або має права Адміністратора організації.

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

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

Формат JSON рядку результата виконання функції:

```
{
	"DocumentFileName":"Назва файла документа",
	"DocumentBody": "……." //тіло документа у base64 }
```

Приклад виклику функції на мові програмування C# у розділі «Опис функції вивантаження внутрішнього документа з системи», змінюється лише URL, на який необхідно відправляти форму.

Опис функції вивантаження внутрішнього документа у форматі Asic-S

URL, на який необхідно відправляти форму для вивантаження документа: https://my.artoffice.com.ua/service/getinternaldocumentasics.

Вивантаження документу може бути виконане користувачем, що є автором документа або перебуває на маршруті обробки документа або має права Адміністратора організації.

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

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

Формат JSON рядку результата виконання функції:

```
{
	"DocumentFileName":"Назва файла документа",
	"DocumentBody": "……." //тіло документа у base64 }
```

Приклад виклику функції на мові програмування C# у розділі «Опис функції вивантаження внутрішнього документа з системи», змінюється лише URL, на який необхідно відправляти форму.

Опис функції отримання статусів внутрішніх документів

URL, на який необхідно відправляти форму:

https://my.art-office.com.ua/service/getinternaldocumentsinfo .

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

```
{"Documents": [ 	//масив
	{"DocumentId": <Ідентифікатор документа 1(число)>},
	…….
	{"DocumentId": <Ідентифікатор документа N (число)>} ],
```

```
 "IsNeedCustomData": false  //Ознака, чи повертати додаткову інформацію про документ
}
```

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 статуса документа",
                "IsDeleted": false, //Ознака видаленого документа
                "Reason": "Причина відхилення, якщо документ у відхиленому стані",
                "ErrorMessage": "повідомлення про помилку",
                "CustomData": "додаткова інформація"
        },
        .........
        ]
}
```

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/getinternaldocumentsinfo"); 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 внутрішніх документів з параметрами

URL, на який необхідно відправляти форму:

https://my.art-office.com.ua/service/getinternaldocumentsinfobyparams.

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
		"IgnoreDeleted": true,      //ознака ігнорувати видалені документи
		"ChangeStatusDate":  "10.09.2017  09:23:51",  //дата,  з  якої  змінився  стан
документа у форматі dd.MM.yyyy HH:mm:ss
		"IsNeedCustomData":  false   //Ознака,  чи  повертати  додаткову  інформацію  про
документ
	}
}
```

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 стану підписання документа",
		"SignStateId": Ідентифікатор стану підписання (число),
```

```
            "SignStateName": "Назва стану підписання",
            "ChangeStatusDate": "Дата зміни статуса документа",
            "IsDeleted": false, //Ознака видаленого документа
            "Reason": "Причина відхилення, якщо документ у відхиленому стані",
            "ErrorMessage": "повідомлення про помилку",
            "CustomData": "додаткова інформація"
      },
      .........
      ]
}
```

Приклад виклику функції мовою програмування C#:

```
var dataJson = @"{""Filter"": {""StartDate"": ""30.08.2017"", ""EndDate"": ""04.09.2017"",
""IgnoreDeleted"": true, ""ChangeStatusDate"": ""04.09.2017 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/getinternaldocumentsinfobyparams"); 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 завантаження нового файлу в не підписаний/не погоджений документ

URL, на який необхідно відправляти форму:

https:// my.art-office.com.ua/service/UpdateInternalDocument.

Використання функції зміни параметрів документа

Опис:

Дана функція дозволяє змінювати параметри існуючого документа в системі, включаючи вид документа, назву документа та його зміст. Для коректного виконання функції необхідно передати наступні параметри:

- Ідентифікатор існуючого документа.
- Нову назву файлу.
- Ідентифікатор виду документа, на який необхідно виконати заміну.
- Нову назву документа в системі.
- Тіло документа у форматі Base64.

Цей метод дозволяє одночасно вносити зміни до всіх зазначених параметрів або до окремих параметрів за потреби.

Доступ:

Функція зміни параметрів доступна для користувачів, які перебувають на маршруті обробки даного документа, а також для автора документа.

Важливо:

Користувачам, можливість змінювати тіло документа, назву та вид документа, доступна до моменту нанесення першого підпису або виконання першого погодження будь-яким користувачем на маршруті.

Автору документа, можливість редагувати вид документа, назву документа та тіло документа після нанесення першого підпису або виконання першого погодження, також не доступна.

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

```
{
       "DocumentId": 12345    //ІД вже існуючого в системі документа
       "DocumentData":     //інформація про документ
              {
              "FileName": "1.docx",   //нова назва файлу
              "TypeId": 1,     //новий вид документа з таблиці «Види внутрішніх документів»
              "Name": "Акт виконаних робіт" //нова назва документа
              },
       "DocumentBody": "……",  //нове тіло документа в base64
}
```

Формат JSON рядку результату виконання функції:

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

Словники

Таблиця «Типи файлів»

Ідентифікатор	Опис
1	Файли з розширенням docx (MSWord)
2	Файли з розширенням xslx (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	eTTH	Відправник, Перевізник, Експедитор, Отримувач **Або** Відправник, Перевізник, Отримувач Залежить від PolicyId
14	Відомість	Автор / Заявник

15	Розрахунок корегування	Виконавець, Замовник
16	Протокол	Сторона 1, Сторона 2, Сторона 3 **Або** Сторона 1, Сторона 2, Сторона 3, Сторона 4 … Залежить від PolicyId
18	Договір про нерозголошення(NDA) (Двосторонній)	Виконавець, Замовник **Або** Сторона 1, Сторона 2
19	Акт коригування	Виконавець, Замовник **Або** Сторона 1, Сторона 2
20	Акт наданих послуг	Виконавець, Замовник **Або** Сторона 1, Сторона 2
21	Акт повернення	Виконавець, Замовник **Або** Сторона 1, Сторона 2
22	Акт приймання-передачі	Виконавець, Замовник **Або** Сторона 1, Сторона 2
23	Замовлення	Автор / Заявник
24	Наказ (Односторонній)	Автор / Заявник
25	Наказ (Двосторонній)	Виконавець, Замовник **Або** Сторона 1, Сторона 2
26	Повернення	Виконавець, Замовник **Або** Сторона 1, Сторона 2
27	Протокол розбіжностей	Виконавець, Замовник **Або** Сторона 1, Сторона 2
28	Специфікація	Автор / Заявник
29	EDI (Односторонній)	Автор / Заявник
30	EDI (Двосторонній)	Виконавець, Замовник **Або** Сторона 1, Сторона 2

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
14	Сторона 4
15	Сторона 5
16	Сторона 6
17	Сторона 7
18	Сторона 8
19	Сторона 9

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 зовнішнього документа»

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

Таблиця «Стан внутрішнього документа»

Ідентифікатор	Опис
1	Створений
4	Відхилений

Таблиця «Стан підписання зовнішнього та внутрішнього документа»

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 звіряння

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