

Опис варіантів використання API Deals

- [1. Введення](#)
- [2. Загальний опис](#)
- [3. Варіанти використання API](#)
 - [3.1 Отримання доступу та автентифікація](#)
 - [3.2 Використання API для роботи з угодами та документами](#)
 - [3.2.1 Отримання ідентифікатора організації \(getOrganization\)](#)
 - [3.2.2 Отримання переліку ідентифікаторів угод \(getDealsList\)](#)
 - [3.2.3 Створення нової угоди \(addNewDeal\)](#)
 - [3.2.4 Запрошення контрагента до угоди \(invitePartner\)](#)
 - [3.2.5 Додавання документа\(ів\) до угоди \(addDocument\)](#)
 - [3.2.6 Завантажити вміст документа \(setDocumentContent\)](#)
 - [3.2.7 Створити запит на видалення документа \(deleteDocumentRequest\)](#)
 - [3.2.8 Додавання існуючих підписів до документа \(addDocumentSignature\) \(DEPRECATED\)](#)
 - [3.2.9 Додавання зовнішніх підписів до документа \(addDocumentSignatureExternal\)](#)
 - [3.2.10 Отримання документа з підписами \(getDocument\)](#)
 - [3.2.11 Отримання поточного стану документа в Системі \(getDocumentInfo\)](#)
 - [3.2.12 Отримання списку ідентифікаторів документів за угодою \(getDealDocumentList\)](#)
 - [3.2.13 Отримати перелік документів за параметрами \(getDocumentList\)](#)
 - [3.2.14 Прийняття документа \(acceptDocument\)](#)
 - [3.2.15 Зворотній виклик за подіями в Системі](#)
 - [3.2.16 Відхилення документа rejectDocument](#)
 - [3.2.17 Отримання списку запрошень до угод getInvitationList](#)
 - [3.2.18 Прийняття/Відхилення запрошень до угоди processInvitation](#)
 - [3.2.19 Старт підписання користувачем за mobileID \(startMobileIDSign\)](#)
 - [3.2.20 Обробка запиту на видалення документа \(processDeleteDocumentRequest\)](#)
 - [3.2.21 Додавання контакту \(addContact\)](#)
 - [3.2.22 Отримати список додаткових атрибутів \(getAdvProperties\)](#)
 - [3.2.23 Отримати список додаткових атрибутів \(getAdvPropertyEnums\)](#)
 - [3.2.24 Отримати список додаткових атрибутів \(getDocProperties\)](#)
 - [3.2.25 Додати додатковий атрибут \(setAdvProperty\)](#)
 - [3.2.26 Додати дані переліку \(setAdvPropertyEnum\)](#)
 - [3.2.27 Створити запит на закриття угоди \(createCloseDealRequest\)](#)
 - [3.2.28 Підтвердити запит на закриття угоди \(confirmCloseDealRequest\)](#)
 - [3.2.29 Відмінити запит на закриття угоди, створений вашою організацією \(cancelCloseDealRequest\)](#)
 - [3.2.30 Відхилити запит на закриття угоди \(rejectCloseDealRequest\)](#)
 - [3.2.31 Перевірити чи існують запити на закриття угоди \(checkCloseDealRequest\)](#)
 - [3.2.32 Створити посилання на документ для зовнішніх контрагентів \(поділитись документом\) \(shareDocuments\)](#)
 - [3.2.33 Оновлення параметрів існуючої угоди \(updateDeal\)](#)
 - [3.2.34 Перелік типів документів організації \(getDocKindList\)](#)
 - [3.2.35 Додати новий тип документу організації \(addDocKind\)](#)
 - [3.2.36 Оновлення типу документу організації \(updateDocKind\)](#)
 - [3.2.36 Видалення типу документу організації \(deleteDocKind\)](#)

1. Введення

Цей документ описує взаємодію API и Системи за допомогою POST и GET запитів через HTTPS-протокол. Параметри API чутливі до реєстру. Для коректної роботи Системи мають бути точно відтворені всі команди, приклади яких описані. API інтерфейс дозволяє здійснювати всі необхідні операції для роботи системою Deals.

2. Загальний опис

Для роботи з API найкраще використовувати модуль "deals-api". Модуль призначений для використання в середовищі **"NodeJS"** та містить зразок використання API.

```
npm install @unitybase/deals-api --registry=https://registry.unitybase.info/
```

Готовий приклад

```
git clone https://github.com/mushu-maxy/deals-api-demo.git
```

Параметри в запитих передаються в тілі запиту в форматі JSON. Для запитів де треба параметри передавати в URL вказано окремо. Дати повинні передаватись як строка в форматі YYYY-MM-DDTHH:mm:ss.sssZ наприклад "2018-01-01T09:32:09.665Z"

Всі запити пердаються за протоколом HTTPS. Адреса API <https://deals.dealssign.com/dealsapi>

Якщо виникає помилка сервіс повертає статус 400 та опис в тілі відповіді

3. Варіанти використання API

3.1 Отримання доступу та автентифікація

Доступ до API можливий для будь-якого користувача зареєстрованого в Системі за допомогою його Логіну та Пароля.

Механізм авторизації описаний в <https://unitybase.info/api/server-v5/tutorial-security.html>. В системі використовується авторизація UB https://unitybase.info/api/server-v5/tutorial-security_ub_schema.html.

Часті помилки при реалізації авторизації:

- метод авторизації <https://deals.dealssign.com/auth?AUTHTYPE=UB&...>
- appName не використовується тому завжди '/'
- При додаванні електронної адреси в параметри адреси треба екранувати символи

<https://deals.dealssign.com/auth?AUTHTYPE=UB&userName=aa%40gmail.com>

`encodeURIComponent('aa@gmail.com') "aa%40gmail.com"`

3.2 Використання API для роботи з угодами та документами

3.2.1 Отримання ідентифікатора організації (getOrganization)

Використовується для отримання ідентифікатора організації (ФОП) в Системі.

Параметри:

- **code** - String ()
- **confirmed** - Boolean (-)

Зразок запиту:

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method":"getOrganization","params":{"code":"00000000","confirmed":true}}
```

Зразок відповіді:

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

{ID: 2000000584907}

<- 400
Content-Type: application/json; charset=UTF-8
{error: 10, description: 'Invalid parameters'}
```

3.2.2 Отримання переліку ідентифікаторів угод (getDealsList)

Використовується для отримання переліку угод організації відповідно до параметрів запити

Параметри:

- orgID - Number (. *getOrganization*) (не обов'язковий) Якщо не вказано всі доступні угоди в тому числі без контрагента
- dateFrom - Date () (не обов'язковий)
- dateTo - Date () (не обов'язковий)
- appendArchive - Boolean () (не обов'язковий)
- integration - Угода для інтеграції

Зразок запити:

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method":"getDealsList","params":{"orgID":123456654321,"dateFrom":"2018-01-01T09:32:09.665Z",
"dateTo":"2018-05-05T00:00:00.000Z","appendArchive":true,"integration":true}}
```

Зразок відповіді:

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

[ { ID: 1000000584907,
  name: ' 1',
  partnerCode: '12345678',
  partnerName: ' ',
  mi_createDate: '2018-01-01T12:00:06Z' },
  { ID: 1000000601203,
```

```
name: ' 2',
mi_createDate: '2018-01-02T14:27:26Z' ]]

<- 400
Content-Type: application/json; charset=UTF-8
{error: 10, description: 'Invalid parameters'}
```

3.2.3 Створення нової угоди (addNewDeal)

Використовується для створення нової угоди в Системі. Якщо передаються шаблони документів, треба щоб всі виконавці мали доступ до угоди.

Параметри:

- **name** - String(120) ()
- **incomeTemplate** - String Назва шабону маршруту для вхідних документів
- **incomeTemplateAuto** - Boolean Шаблон маршруту для вхідних документів в автоматичному режимі
- **outcomeTemplate** - String Назва шабону маршруту для вихідних документів
- **outcomeTemplateAuto** - Boolean Шаблон маршруту для вихідних документів в автоматичному режимі
- **incomeTemplateMap** - {"document type name": "Template name"} Шаблони маршрутів документів для окремих типів вхідних документів
- **outcomeTemplateMap** - {"document type name": "Template name"} Шаблони маршрутів документів для окремих типів вихідних документів
- **description** - String(512) ()
- **onlyAuthor** - Boolean (')
- **payForPartner** - Boolean () (')
- **accessType** - String(20) ['GENERAL', 'LIMITED'] (')
- **isDocConfirm** - Boolean (')
- **moderator** - String (')
- **accessUser** - (String|String[]) () (')
- **accessGroup** - (String|String[]) Назва груп(пп) які мають досуп до угоди (не обов'язковий параметр)
- **accessGroupFull** - (String|String[]) Назва груп(пп) (включаючи підлеглі) які мають досуп до угоди (не обов'язковий параметр)
- **partnerCount** [Number] . dealType ('ONE_SIDE', 'TWO_SIDE') . MANY_SIDE 3 - 10
- **dealType** - String Тип угоди 'ONE_SIDE', 'TWO_SIDE', 'MANY_SIDE'. По замовчанню 'TWO_SIDE'- двустороння угода. 'MANY_SIDE' - трьохстороння угода чи більше, якщо передано partnerCount. 'ONE_SIDE' - угода без контрагентів (внутрішні документи)
- **integration** - Boolean (')

:

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method":"addNewDeal","params":{"name":" 1", "accessGroup": ["grp1", "grp2"], "
incomeTemplateMap": {"act": "template1", "contract": "template2" }}
```

Зразок відповіді:

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

{ID: 2000000584907}

<- 400
Content-Type: application/json; charset=UTF-8
{error: 10, description: 'Invalid parameters'}
```

3.2.4 Запрошення контрагента до угоди (invitePartner)

Використовується для запрошення контрагента до угоди.

Параметри:

- **dealID** - Number
- orgID - Number (не обов'язковий параметр)
- orgCode - String Код організації (не обов'язковий параметр)
- confirmed - Boolean Перевіряти що організація визначена по orgCode пройшла перевірку ЕДПРОУ за КЕП
- eMail - String (')
- publicGroup - String(120) (')
- **invite** - String(.)

Якщо запрошення в угоду для організації, що ще не зареєстрована в системі (не має orgID) то необхідно заповнити тільки параметри dealID, eMail, invite. Кількість запрошень в угоду не обмежена але прийняти запрошення зараз може тільки одна організація. Якщо угода вже має партнера, то створення нового запрошення буде відхилено з помилкою.

Для запрошення зареєстрованої в системі організації, необхідно заповнити тільки параметри dealID та orgID або orgCode та confirmed.

Зразок запиту:

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method":"invitePartner","params":{"orgID": 1000000010001, "dealID": 12300000789, "invite": "123" }}
```

Зразок відповіді:

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

{sucess: true}

<- 400
Content-Type: application/json; charset=UTF-8
{error: 10, description: 'Invalid parameters'}
```

3.2.5 Додавання документа(ів) до угоди (addDocument)

Використовується для додавання документа (документів) до угоди

Параметри:

- **dealID** - Number ()
- **document** - Object|Object[] (, ')
 - **name** - String(120) ()
 - **requireConfirm** - Boolean ()
 - **notifyWhenComplete** - Boolean (. -)
 - **docType** - String Код
 - **docKind** - String Код виду документа. Код повинен бути в переліку типів документів організації
 - **documentDate** - String Дата документа у форматі YYYY-MM-DDTHH:mm:ss.sssZ (ISO 8601)
 - **storageEndDate** - String Кінцева дата зберігання документа у форматі YYYY-MM-DDTHH:mm:ss.sssZ (ISO 8601) (**не обов'язковий параметр**)
 - **documentSumm** - Number Сума документа
 - **signers** - sting[] Перелік назв користувачей, які будуть назначені підписувачами. Користувач повинен бути з тієї самої організації.
 - **approvers** - sting[] Перелік назв користувачей, які будуть назначені согласуючими. Користувач повинен бути з тієї самої організації. (**не обов'язковий параметр**)
 - **signatures** - {signatureB64: string.<base64>}[] Массив об'єктів з атрибутом signatureB64 де вказано підпис в форматі base64. Підписи будуть додані як зовнішні (Так саме як в методі addDocumentSignatureExternal)
 - **payForPartner** - boolean true - Автор платить за підписи (не обов'язковий параметр)
 - **firstSignerPhone** - string Номер телефона для автоматичного запуску підписання першим підписантом (Необхідно щоб був хоч один підписант)
 - **docProperties** String - Об'єкти з даними додаткових атрибутів
 - **docProperties.advProperty** String - Код додаткового атрибуту
 - **docProperties.enumGroup** String - Група переліку обов'язкова для атрибуту з attrType = Enum
 - **docProperties.valDate** Date - Значення для атрибуту з attrType = Date
 - **docProperties.valString** String - Значення для атрибуту з attrType = String
 - **docProperties.valEnum** Array<String> - Значення для атрибуту з attrType = Enum
 - **docProperties.valNum** Number - Значення для атрибуту з attrType = Number
 - **docProperties.valBool** Boolean - Значення для атрибуту з attrType = Boolean

Можливі значення для **docType**

- INFORMATION - Інформаційний
- ONE_SIDE_SIGNING - Підпис автора
- RECIPIENT_SIGNING - Підпис контрагента
- TWO_SIDE_SIGNING - Першим підписує автор, потім контрагент
- TWO_SIDE_SIGNING_REVERS - Першим підписує контрагент потім автор
- AUTHOR_SIGNING - Внутрішнє підписання (для підписання внутрішніх документів в односторонніх угодах)

Можливі значення для коду **docKind**

- invoice
- act
- order
- application
- bill
- contract

Перелік може бути змінено адміністратором системи або через методи api addDocKind, updateDocKind, deleteDocKind, getDocKindList!

Повертає ідентифікатори доданих документів

Зразок запиту:

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method":"addDocument","params":{"dealID": 1000000010001, "document": [{"name": "",
"requireConfirm": false, "docType": "ONE_SIDE_SIGNING", "signers": ["user1@gmail.com"],
docProperties: '{"advProperty":"sum","valNum":2}', "signatures": [{"signatureB64": "
MIIMrgYJKoZIhvcNA...HI6IQ==" } ] ]}}
```

Зразок відповіді:

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

[100000345001]

<- 400
Content-Type: application/json; charset=UTF-8
{"error": 10, description: 'Invalid parameters'}
```

3.2.6 Завантажити вміст документа (setDocumentContent)

Використовується для завантаження вмісту документа

Параметри (Передаються в URL):

- **ID** - Ідентифікатор документа (Його повертає метод addDocument)

Зразок запиту:

```
-> POST /dealsapi?method=setDocumentContent&ID=10000456001 HTTP/1.1
Content-Type: application/octet-stream; charset=UTF-8

--binary data
```

Зразок відповіді:

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

{"success": true}

<- 400
Content-Type: application/json; charset=UTF-8
{"error": 10, description: 'Invalid parameters'}
```

3.2.7 Створити запит на видалення документа (deleteDocumentRequest)

Параметри:

- **documentID** - Number - Ідентифікатор документа

- `reason` - String - (не обов'язкове Причина запиту буде додано як повідомлення в обговорення угоди)

Якщо документ на стороні автора то він відразу віддається. Якщо друга сторона зробила запит на видалення то теж відразу видається. В інших випадках створюється запит на видалення.

Видалення дозволено якщо:

- доокумент доступний користувачу
- користувач або адмін організації, або підписант, або користувач з розширеними правами

Повертає JSON `{requestID: Number, status: String}`

Де `requestID` - ід запиту на видалення.

`status` - Статус операції. Можливі значення ('DELETED', 'REQUEST_CREATED')

Зразок запиту

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method": "deleteDocumentRequest", "params": {"documentID": 1000000010001}}
```

Зразок відповіді

```
<- 200 OK
Content-Type: application/json; charset=UTF-8
{requestID: 10000099, status: 'DELETED'}

<- 400
Content-Type: application/json; charset=UTF-8
{error: 10, description: 'Invalid parameters'}
```

3.2.8 Додавання існуючих підписів до документу (`addDocumentSignature`) (DEPRECATED)

Підтримку цієї функції відключено. Замість неї слід використовувати `addDocumentSignatureExternal`

Використовується для додавання підписів зроблених в інших ситемах. При завантаженні перевіряється відповідність документу та відповідність ЄДРПОУ з сертифікату підписувача та ЄДРПОУ організації. Не передавати разом з `approvers`

Параметри (Передаються в URL):

- **ID** - Ідентифікатор документа
- **user** - Логін користувача чий підпис додається в систему

Сам підпис передається в тілі запиту як бінарні дані

Зразок запиту

```
-> POST /dealsapi?method=addDocumentSignature&ID=10000456001&user="user%40domain" HTTP/1.1
Content-Type: application/octet-stream; charset=UTF-8

--binary data
```

Зразок відповіді

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

{"success": true}

<- 400
```

```
Content-Type: application/json; charset=UTF-8
{error: 10, description: 'Invalid parameters'}
```

3.2.9 Додавання зовішніх підписів до документу (addDocumentSignatureExternal)

Використовується для додавання підписів зроблених в інших ситемах. При завантаженні перевіряється відповідність документу. На відмінність від addDocumentSignature не перевіряє відповідність ЄДРПОУ з сертифікату підписувача та ЄДРПОУ організації (дозволяє завантажувати підписи "третіх" сторін). Підпис завантажується поза визначеним маршрутом узгодження але доступні всім користувачам, так як і інші підписи. Підпис можливо додати до документу з будь-якою "дією" та незалежно від типу документу.

Параметри (Передаються в URL):

- **ID** - Ідентифікатор документа

Сам підпис передається в тілі запиту як бінарні дані

Зразок запиту

```
-> POST /dealsapi?method=addDocumentSignatureExternal&ID=10000456001 HTTP/1.1
Content-Type: application/octet-stream; charset=UTF-8
```

```
--binary data
```

Зразок відповіді

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

{"success": true}

<- 400
Content-Type: application/json; charset=UTF-8
{error: 10, description: 'Invalid parameters'}
```

3.2.10 Отримання документа з підписами (getDocument)

Використовується для вивантаження вмісту документа з накладеними ЕЦП.

Параметри:

- **ID** - Number - Ідентифікатор документа
- **resultType** - String - Тип результату
- **visualizationType** - String - Тип візуалізації (')

Можливі значення для **resultType**

- DOCUMENT - Тільки документ
- SIGNATURES - Тільки підписи
- VISUALIZATION -
- SIGNATURES&VISUALIZATION - Підписи та візуалізація
- FULL - Документ, підписи та візуалізація

Можливі значення для **visualizationType**

- withoutTimeStamp - Візуалізація без позначки часу

Повертає zip архив

Зразок запиту:

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method":"getDocument","params":{"ID": 1000000010001, "resultType": "FULL"}}
```

Зразок відповіді:

```
<- 200 OK
Content-Type: application/octet-stream

--binary data

<- 400
Content-Type: application/json; charset=UTF-8
{"error": 10, "description": "Invalid parameters"}
```

3.2.11 Отримання поточного стану документа в Системі (getDocumentInfo)

Використовується для отримання поточного стану документа в Системі.

Параметри:

- **ID** - Number()

Можливі стани документа

- REJECTED - Відхилено
- SIGNING - Підписання
- ACCEPTING - Прийняття
- REJECTED_INNER_ACCEPTING - Відхилено модератором сторони яка створила документ
- SIGNING_PARTNER - Підписання партнером
- SIGNED - Підписано

Зразок запиту:

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method":"getDocumentInfo","params":{"ID":10000543001}}
```

Зразок відповіді:

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

{
  "STATUS": "SIGNING",
```

```
"statusInner": "SIGNING",
"name": "testDoc.pdf",
"docType": "ONE_SIDE_SIGNING",
"docKind": null,
"requireConfirm": 0,
"summ": null,
"signers": [
  {
    "appointedUser": "user@gmail.com",
    "complete": false,
    "signatureCreated": null,
    "user": null
  }
]
}

<- 400
Content-Type: application/json; charset=UTF-8
{error: 10, description: 'Invalid parameters'}
```

3.2.12 Отримання списку ідентифікаторів документів за угодою (getDealDocumentList)

Параметри:

- dealID - Number[[Number] Ідентифікатор угоди або декілька (масив ідентифікаторів) угод.

Зразок запиту

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method": "getDealDocumentList", "params": {"dealID": 100000000001}}
```

Зразок відповіді

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

[ { ID: 1000000584907,
  name: ' 1',
  doc_type: 'ONE_SIDE_SIGNING',
  dt: '2018-06-25T07:55:28Z',
  summ: 1.2
},
  { ID: 1000000601203,
  name: ' 2',
  doc_type: 'ONE_SIDE_SIGNING',
  dt: '2018-06-25T12:00:00Z',
  summ: null
}]

<- 400
Content-Type: application/json; charset=UTF-8
{error: 10, description: 'Invalid parameters'}
```

Поля у відповіді

- ID {Number}
- name {String}
- extension {String}

- doc_type {String} Дія за документом
- docKindCode {String}
- docKindName {String}
- dt {Date} Дата документу (Не обов'язкове для заповнення)
- summ {Number} Сума документу (Не обов'язкове для заповнення)
- deal {Number} Ідентифікатор угоди
- creatorRegNum {String} ЕДРПО сторони що створила документ
- state {String} Стан документу
- require_confirm {Boolean} Необхідно підтвердження прийняття від партнера
- payForPartner {Boolean} Сторона що створила документ оплачує підписи партнера за цим документом
- dateCreate {Date}
- dateModify {Date}

3.2.13 Отримати перелік документів за параметрами (getDocumentList)

Параметри:

- dealID - Number | [Number] Ідентифікатор угоди або декілька (масив) ідентифікаторів угод.
- state - String | [String] Стан документу або перелік станів
- dateCreateFrom Date - Дата створення починаючи з
- dateModifyFrom Date - Дата останніх змін починаючи з
- createdBy String можливі значення (partner, self)
- docType String можливі значення (TWO_SIDE_SIGNING, ONE_SIDE_SIGNING, RECIPIENT_SIGNING, INFORMATION)
- documentComplete Boolean notifyWhenComplete

Зразок запиту

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method": "getDocumentList", "params": {"state": ['SIGNED', 'INFORMATION', 'ACCEPTED']}}
```

Зразок відповіді

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

[ { ID: 1000000584907,
  name: ' 1',
  doc_type: 'ONE_SIDE_SIGNING',
  dt: '2018-06-25T07:55:28Z',
  summ: 1.2
},
{ ID: 1000000601203,
  name: ' 2',
  doc_type: 'ONE_SIDE_SIGNING',
  dt: '2018-06-25T12:00:00Z',
  summ: null
}]

<- 400
Content-Type: application/json; charset=UTF-8
{"error": 10, "description": 'Invalid parameters'}
```

Поля у відповіді

- ID {Number}
- name {String}

- extension {String}
- doc_type {String} Дія за документом
- docKindCode {String}
- docKindName {String}
- dt {Date} Дата документу (Не обов'язкове для заповнення)
- summ {Number} Сума документу (Не обов'язкове для заповнення)
- deal {Number} Ідентифікатор угоди
- creatorRegNum {String} ЕДРПО сторони що створила документ
- state {String} Стан документу
- require_confirm {Boolean} Необхідно підтвердження прийняття від партнера
- payForPartner {Boolean} Сторона що створила документ оплачує підписи партнера за цим документом
- dateCreate {Date}
- dateModify {Date}

3.2.14 Прийняття документу (acceptDocument)

Метод дозволяє прийняти документ, який прийшов від партнера та визначити маршрут. Користувач під правами якого здійснюється операція повинен мати доступ до угоди та мати роль підписант або адміністратор організації.

Користувач, що приймає документ повинен бути модератором угоди.

Якщо документ знаходиться не в стані прийняття ACCEPTING, буде повернута помилка.

Якщо документ не має маршрута або в маршруті немає жодного підписанта, буде повернута помилка.

Якщо користувачі в маршруті не мають доступу до угоди, буде повернута помилка.

Якщо користувачі передані як підписанти не мають відповідну роль, буде повернута помилка.

Параметри:

- **ID** - Number ()
- **signers** - sting[] Перелік назв користувачей, які будуть назначені підписувачами. Користувач повинен бути з тієї самої організації.
- **approvers** - sting[] Перелік назв користувачей, які будуть назначені согласуючими. Користувач повинен бути з тієї самої організації. **(не обов'язковий параметр)**

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method": "acceptDocument", "params": {"ID": 10000543001, "signers": ["user1@gmail.com"]}}
```

Зразок відповіді

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

{ ID: 10000543001}

<- 400
Content-Type: application/json; charset=UTF-8
{error: 10, description: 'Invalid parameters'}
```

3.2.15 Зворотній виклик за подіями в Системі

Зворотній виклик буде здійснюватись, якщо в профілі організації вказано адресу (web).

Зразок зворотного виклику

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"data":{"event":"documentStatusChanged","params":{"documentID":10000543001, "STATUS": "SIGNED"}}, signature '11212121212'}
```

Разом з даними передається підпис.

Підпис розраховується за алгоритмом:

```
data.signature = SHA256(JSON.stringify(data) + ':' + organization.privateKey).toString()
```

privateKey - String Код, який вказано в профілі організації.

3.2.16 Відхилення документу rejectDocument

Метод дозволяє відхилити документ, якщо він в станах (SIGNING_PARTNER, SIGNING) підписання

Параметри:

- **ID - Number()**

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method":"rejectDocument","params":{"ID":10000543001}}
```

Зразок відповіді

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

{ ID: 10000543001}

<- 400
```

```
Content-Type: application/json; charset=UTF-8
{error: 10, description: 'Invalid parameters'}
```

3.2.17 Отримання списку запрошень до угод `getInvitationList`

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method": "getInvitationList", "params": {}}
```

Зразок відповіді

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

[{"ID": 10000543001, requestID: 123000321, dateCreate: "2020-09-14T10:29:47Z", partnerCode:
"00000001", partnerName: " 5", name: ' 1'}]

<- 400
Content-Type: application/json; charset=UTF-8
{error: 10, description: 'Invalid parameters'}
```

В ответе

ID- Ид сделки

requestID - ()

dateCreate -

partnerCode - (' ')

3.2.18 Прийняття/Відхилення запрошень до угоди `processInvitation`

Параметри:

- **invitationID** - Number (Ідентифікатор заявки. Можна отримати з відповіді на `getInvitationList` атрибут `requestID`)
- **invitationStatus** - String (*APPROVED* or *REJECTED*)
- **integration** - Boolean (Прийняти угоду з признаком зовнішньої інтеграції по замовчанню `false`)

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method": "processInvitation", "params": {"invitationID": 1230000888, "invitationStatus": 'APPROVED'
}}
```

Зразок відповіді

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

{success: true}
```

```
<- 400
Content-Type: application/json; charset=UTF-8
{error: 10, description: 'Invalid parameters'}
```

3.2.19 Старт підписання користувачем за mobileID (startMobileIDSign)

Параметри:

- **documentID** - Number ()
- **firstSignerPhone** - String Номер телефону для автоматичного запуску підписання першим підписантом (Необхідно щоб була активна задача на підписанті)

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method":"startMobileIDSign","params":{"documentID": 1230000888, "firstSignerPhone":
"+380675555555" }}
```

Зразок відповіді

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

{success: true}

<- 400
Content-Type: application/json; charset=UTF-8
{error: 10, description: 'Invalid parameters'}
```

3.2.20 Обробка запиту на видалення документу (processDeleteDocumentRequest)

Параметри:

- **documentID** - Number - Ідентифікатор документу
- **state** - String Можливі стани REJECTED - відхилити, APPROVED - підтвердити, CANCELED - відмінити власний запит
- **reason** - String Для стару REJECTED

Якщо для документу немає запиту на видалення буде повернуто помилку

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method":"processDeleteDocumentRequest","params":{"documentID": 1230000888, "state": "APPROVED" }}
```

Зразок відповіді

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

{success: true}

<- 400
Content-Type: application/json; charset=UTF-8
{error: 10, description: 'Invalid parameters'}
```

3.2.21 Додавання контакту (addContact)

Параметри:

- **name** - String - Назва крнтакту
- **regNum** - String ЕДРПОУ/РНОКПП контакту
- **eMail** - String Електронна пошта контакту
- **phone** - String Телефон контакту
- **tags** - String[] Теги контакту

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method":"addContact","params":{"name": "name", "regNum": "00000000", "eMail": "test@test.com", "phone": "+(380)123456789", "tags": ["tag1"] }}
```

Зразок відповіді

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

{success:true}

<- 400
Content-Type: application/json; charset=UTF-8
{error: 10, description: 'Invalid parameters'}
```

3.2.22 Отримати список додаткових атрибутів (getAdvProperties)

Параметри:

- **orgDocKindCode** - String - Код типу документа

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method":"getAdvProperties","params":{"orgDocKindCode": "exampleCode" } }
```

Зразок відповіді

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

{success:true,data:[{"ID":3000000253158,"name":"sad","code":"sad","attrType":"Enum","allowNull":0,"sortOrder":1,"defValEnum":null,"enumGroup":"gg"}]}

<- 400
Content-Type: application/json; charset=UTF-8
{error: 10, description: 'orgDocKindCode must be specified'}
```

3.2.23 Отримати список додаткових атрибутів (getAdvPropertyEnums)

Параметри:

- **enumGroup** - String - Назва групи переліків

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method": "getAdvPropertyEnums", "params": {"enumGroup": "exampleGroup" } }
```

Зразок відповіді

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

{success: true, data: [{"ID": 3000000250603, "enumValue": "dd", "enumGroup": "exampleGroup"}]}

<- 400
Content-Type: application/json; charset=UTF-8
{error: 10, description: 'enumGroup must be string'}
```

3.2.24 Отримати список додаткових атрибутів (getDocProperties)

Параметри:

- **ID** - Number - ID документу

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method": "getDocProperties", "params": {"ID": 3000000750564 } }
```

Зразок відповіді

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

{success: true, data: [{"ID": 3000000254784, "name": "Сумма", "code": "sum", "attrType": "Number", "valNum": 33}]}

<- 400
Content-Type: application/json; charset=UTF-8
{error: 10, description: 'ID must be specified'}
```

3.2.25 Додати додатковий атрибут (setAdvProperty)

Параметри:

- **orgDocKindCode** - String - Код типу документа
- **data** - String - JSON з даними атрибуту
- **data.code** - String Код додаткового атрибуту
- **data.name** - String Назва додаткового атрибуту

- **data.attrType** String - Тип атрибуту String | Enum | Number | Date | Boolean
- **data.allowNull** Boolean - Обов'язковість типу
- **data.sortOrder** Number - Порядок сортування типу
- **data.enumGroup** String - Група переліку обов'язкова для attrType = Enum
- **data.defValDate** Date - Значення за замовчуванням для attrType = Date
- **data.defValString** String - Значення за замовчуванням для attrType = String
- **data.defValEnum** Array<String> - Значення за замовчуванням для attrType = Enum
- **data.defValNum** Number - Значення за замовчуванням для attrType = Number
- **data.defValBool** Boolean - Значення за замовчуванням для attrType = Boolean

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method":"setAdvProperty","params":{"orgDocKindCode": "exampleCode", "data": '{"name":"String prop","code":"String_prop","sortOrder":1,"attrType":"String","defValString":"String_prop"}' }
```

Зразок відповіді

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

{success:true}

<- 400
Content-Type: application/json; charset=UTF-8
{error: 20, description: 'property with this code already exist.'}
```

3.2.26 Додати дані переліку (setAdvPropertyEnum)

Параметри:

- **data** - String - JSON з даними атрибуту
- **data.enumValue** - String - Значення переліку
- **data.enumGroup** String - Група переліку

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method":"setAdvPropertyEnum","params":{"data": '{"enumGroup":"apiEnumTest","enumValue":"testApiVall"}' }
```

Зразок відповіді

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

{success:true}

<- 400
Content-Type: application/json; charset=UTF-8
{error: 10, description: 'data must be passed as json string'}
```

3.2.27 Створити запит на закриття угоди (createCloseDealRequest)

Параметри:

- **dealID** - Number - ID of the deal

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method":"createCloseDealRequest","params":{"dealID: 3000078956712"} }
```

Зразок відповіді

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

{success:true}

<- 400
Content-Type: application/json; charset=UTF-8
{error: 10, description: 'Invalid request. Empty parameter dealID'}
```

3.2.28 Підтвердити запит на закриття угоди (confirmCloseDealRequest)

Параметри:

- **dealID** - Number - ID of the deal

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method":"confirmCloseDealRequest","params":{"dealID: 3000078956712"} }
```

Зразок відповіді

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

{success:true}

<- 400
Content-Type: application/json; charset=UTF-8
{error: 10, description: 'Invalid request. Empty parameter dealID'}
```

3.2.29 Відмінити запит на закриття угоди, створений вашою організацією (cancelCloseDealRequest)

Параметри:

- **dealID** - Number - ID of the deal

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method":"cancelCloseDealRequest","params":{"dealID: 3000078956712"} }
```

Зразок відповіді

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

{success: true}

<- 400
Content-Type: application/json; charset=UTF-8
{error: 10, description: 'Invalid request. Empty parameter dealID'}
```

3.2.30 Відхилити запит на закриття угоди (rejectCloseDealRequest)

Параметри:

- **dealID** - Number - ID of the deal
- **reason** - String

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method":"rejectCloseDealRequest","params":{"dealID: 3000078956712, reason: 'The reason'}} }
```

Зразок відповіді

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

{success: true}

<- 400
Content-Type: application/json; charset=UTF-8
{error: 10, description: 'Invalid request. Empty parameter dealID'}
```

3.2.31 Перевірити чи існують запити на закриття угоди (checkCloseDealRequest)

Повертає перелік ідентифікаторів угод де є запит на закриття угоди

Параметри:

- **dealID** - Number - Якщо не вказано буде передано список всіх угод, де є запити на закриття угоди

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method":"checkCloseDealRequest","params":{}} }
```

Зразок відповіді

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

{success: true, deals: [3000078956712, 3000078956713]}

<- 400
Content-Type: application/json; charset=UTF-8
{error: 10, description: 'Invalid request. Invalid parameter dealID'}
```

3.2.32 Створити посилання на документ для зовнішніх контрагентів (поділитись документом) (shareDocuments)

Створює посилання для зовнішнього контрагента на документи та відправляє запрошення на електронну пошту
Параметри:

- **documents** - Number[] - Массив ідентифікаторів документів в системі, які потрібно додати в посилання.
- **eMail** - String Електронна адреса контрагента
- **accessType** -String (Не обов'язковий) Може приймати значення LINK та EMAIL_OTP (по замовчанню). LINK - Документи доступні по наявності посилання, EMAIL_OTP - Для доступу необхідно підтвердити свою електронну адресу
- **dateTo** - Date (Не обов'язковий) Дата до якої посилання дійсне. По замовчанню 14 днів з поточної дати.
- **edrpou** - String (Не обов'язковий) ЕДРПОУ або РНОКПП контрагента. Якщо потрібен підпис контрагента, то буде перевірено відповідне значення з сертифікату контрагенту. Якщо значення не співпадає, підписання не буде зафіксовано. Якщо буде знайдено контрагента з таким ЕДРПОУ або РНОКПП в системі, буде повернуто помилку.
- **requireSigning** - Boolean (Не обов'язковий) Пропонувати контрагенту підписати документ. Зроблений підпис буде доступний в системі як зовнішній. По замовчанню підпис не потрібен.

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method":"shareDocuments","params":{"documents": [300000568002,300000568003], "eMail":
"conragent.address@gmail.com", "requireSigning": true, "edrpou": "11111111", "accessType":
"EMAIL_OTP" } }
```

Зразок відповіді

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

{"success":true,"shareID":3000078956712}

<- 400
Content-Type: application/json; charset=UTF-8
{"error":10,"description":'Invalid or empty request parameter eMail'}
```

3.2.33 Оновлення параметрів існуючої угоди (updateDeal)

Використовується для оновлення параметрів існуючої угоди в Системі. Якщо передаються шаблони документів, треба щоб всі виконавці мали доступ до угоди.

Параметри:

- **ID** - Number **Обов'язковий параметр**
- **name** - String(120) ()
- **incomeTemplate** -String Назва шабону маршруту для вхідних документів
- **incomeTemplateAuto** - Boolean Шаблон маршруту для вхідних документів в автоматичному режимі
- **outcomeTemplate** - String Назва шабону маршруту для вихідних документів
- **outcomeTemplateAuto** - Boolean Шаблон маршруту для вихідних документів в автоматичному режимі
- **incomeTemplateMap** - {"document type name": "Template name"} Шаблони маршрутів документів для окремих типів вхідних документів
- **outcomeTemplateMap** - {"document type name": "Template name"} Шаблони маршрутів документів для окремих типів вихідних документів
- **description** - String(512) ()
- **onlyAuthor** - Boolean (')
- **payForPartner** - Boolean (') (')

- `accessType` - `String(20)` `['GENERAL', 'LIMITED']` (')
- `isDocConfirm` - `Boolean` (')
- `moderator` - `String` (')
- `accessUser` - `(String|String[])` () (')
- `accessGroup` - `(String|String[])` Назва груп(пн) яким треба додати досуп до угоди (не обов'язковий параметр)
- `accessGroupFull` - `(String|String[])` Назва груп(пн) (включаючи підлеглі) яким треба досуп до угоди (не обов'язковий параметр)
- `accessUserRemove` - `(String|String[])` () (')
- `accessGroupRemove` - `(String|String[])` () (')
- `accessGroupRemoveFull` - `(String|String[])` Назва груп(пн)(включаючи підлеглі) у яких треба забрати досуп до угоди (не обов'язковий параметр)
- `useOrgDocTypes` `(String)` . (')
- `integration` - `Boolean` (')

:

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method":"updateDeal","params":{"ID": 200003436780102, "name":" 1", "accessGroup": ["grp1", "grp2"], "incomeTemplateMap": {"act": "template1", "contract": "template2" }}
```

Зразок відповіді:

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

{}

<- 400
Content-Type: application/json; charset=UTF-8
{"error": 10, "description": 'Invalid parameters'}
```

3.2.34 Перелік типів документів організації (getDocKindList)

Використовується для отримання переліку типів документів організації.

Параметри:

- **active** - **Boolean** Фільтр по ознаці активний
- **code** - **Boolean** Фільтр по коду
- **ID** - **Number** Фільтр по ID

Зразок запиту:

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method":"getDocKindList","params":{"code": "act"}}
```

Зразок відповіді:

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

[{"ID": 1234567, "code": "act", "name": "act", "description": "act", "active": 1, "locking": 0,
"filterActions": 0 }]

<- 400
Content-Type: application/json; charset=UTF-8
{"error": 10, description: 'The parameter code must be string'}
```

3.2.35 Додати новий тип документу організації (addDocKind)

Використовується для додавання нового типу документу організації. Повертає ідентифікатор типу документу в ситсемі

Параметри:

- **code** - String(50) Обов'язковий параметр. Код.
- **name** - String(100) Обов'язковий параметр.
- **description** - String(512) Опис
- **locking** - Boolean Ознака що тип документу з блокуючий
- **active** - Boolean Ознака що тип документу активний
- **filterActions** - Boolean Ознака що по типу може бути обмеження Дій

Зразок запити:

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method": "addDocKind", "params": {"code": "act", "name": "act"}}
```

Зразок відповіді:

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

{"ID": 12345678}

<- 400
Content-Type: application/json; charset=UTF-8
{"error": 10, description: 'The parameter code must be string'}
```

3.2.36 Оновлення типу документу організації (updateDocKind)

Використовується для оновлення параметрів типу документу організації.

Параметри:

- **ID** - Number Обов'язковий параметр. Ідентифікатор в системі.
- **code** - String(50) Код.
- **name** - String(100)

- **description** - String(512) Опис
- **locking** - Boolean Ознака що тип документу є блокуючий
- **active** - Boolean Ознака що тип документу активний
- **filterActions** - Boolean Ознака що по типу може бути обмеження Дій

Зразок запиту:

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method":"updateDocKind","params":{"ID": 1234567, "code": "act 1", "name": "act 1"}}
```

Зразок відповіді:

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

{ID: 12345678}

<- 400
Content-Type: application/json; charset=UTF-8
{error: 10, description: 'The parameter code must be string'}
```

3.2.36 Видалення типу документу організації (deleteDocKind)

Використовується для видалення типу документу організації. Якщо тип документу вже використано, то видалення неможливе і буде викликати помилку. Для відключення типу треба використовувати updateDocKind де передати

Unknown macro: {"ID}

Параметри:

- **ID** - Number Обов'язковий параметр. Ідентифікатор в системі.

Зразок запиту:

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8

{"method":"updateDocKind","params":{"ID": 1234567}}
```

Зразок відповіді:

```
<- 200 OK
Content-Type: application/json; charset=UTF-8

{ID: 12345678}

<- 400
Content-Type: application/json; charset=UTF-8
{error: 10, description: 'The parameter code must be string'}
```