



API Deals



1	Введення	2
2	Загальний опис	3
3	Варіанти використання API	4
3.1	Отримання доступу та автентифікація	4
3.2	Формат даних	4
3.3	Використання API для роботи з угодами та документами	4
3.3.1	Отримання ідентифікатора організації (getOrganization)	4
3.3.2	Отримання переліку ідентифікаторів угод з організацією (getDealsList).....	5
3.3.3	Отримання списку ідентифікаторів документів за угодою (getDealDocumentList)	6
3.3.4	Створення нової угоди (addNewDeal)	6
3.3.5	Створити запрошення контрагента до угоди (invitePartner).....	7
3.3.6	Додати документ(и) до угоди (addDocument)	8
3.3.7	Завантажити вміст документа (setDocumentContent).....	9
3.3.8	Додати підпис до документа (addDocumentSignature)	9
3.3.9	Отримання документа з підписами (getDocument).....	10
3.3.10	Отримання поточного стану документа в Системі (getDocumentInfo).....	11
3.3.11	Зворотній виклик за подіями в Системі	12

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 "UnityBase" або "NodeJS" та містить зразок використання API.

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

Всі запити передаються за протоколом HTTPS.

Адреса API <https://deals.dealssign.com/dealsapi>

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-V41/tutorial-security.html>. В системі використовується авторизація UB.

3.2 Формат даних

Більшість запитів використовують JSON формат. Дати повинні передаватись як рядок в форматі YYYY-MM-DDTHH:mm:ss.sssZ наприклад "2018-01-01T09:32:09.665Z"

3.3 Використання API для роботи з угодами та документами

3.3.1 Отримання ідентифікатора організації (getOrganization)

Використовується для отримання ідентифікатора організації (ФОП) в Системі.

Параметри:

- **code** - String (Код ЄДРПОУ (ІПН))
- **confirmed** - Boolean (Перевірити - чи був підтверджений код ЄДРПОУ (ІПН))

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'}
```

3.3.2 Отримання переліку ідентифікаторів угод з організацією (getDealsList)

Використовується для отримання переліку угод організації відповідно до параметрів запити

Параметри:

- **orgID** - Number (Ідентифікатор організації. Для отримання використовується метод *getOrganization*)
- **dateFrom** - Date (Дата з)
- **dateTo** - Date (Дата по)
- **appendArchive** - Boolean (Включати угоди з архіву)

Зразок запити:

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8
{"method":"getDealsList","params":{"orgID":"00000000","dateFrom":"2018-01-01T09:32:09.665Z","dateTo":"2018-05-05T00:00:00.000Z","appendArchive":true}}
```

Зразок відповіді:

```
<- 200 OK
Content-Type: application/json; charset=UTF-8
[ { ID: 1000000584907,
  name: 'Угода 1',
  mi_createDate: '2018-01-01T12:00:06Z' },
  { ID: 1000000601203,
  name: 'Угода 2',
  mi_createDate: '2018-01-02T14:27:26Z' } ]
<- 400
Content-Type: application/json; charset=UTF-8
{error: 10, description: 'Invalid parameters'}
```

3.3.3 Отримання списку ідентифікаторів документів за угодою (getDealDocumentList)

Використовується для отримання переліку документів за угодою

Параметри:

- **dealID** - Number (Ідентифікатор угоди. Для отримання використовується метод *getDealList*)

Зразок запити:

```
-> 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: 'документ 1',
  doc_type: 'ONE_SIDE_SIGNING',
  dt: '2018-06-25T07:55:28Z',
  summ: 1.2
},
  { ID: 1000000601203,
  name: 'документ 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'}
```

3.3.4 Створення нової угоди (addNewDeal)

Використовується для створення нової угоди в Системі.

Параметри:

- **name** - String(120) (Назва угоди)

- **description** - String(512) (*Опис угоди*)

Зразок запиту:

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8
{"method":"addNewDeal","params":{"name":"Угода 1"}}
```

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'}
```

3.3.5 Створити запрошення контрагента до угоди (invitePartner)

Використовується для запрошення контрагента до угоди.

Параметри:

- **dealID** - Number (*Ідентифікатор угоди*)
- **orgID** - Number (*Ідентифікатор контрагента*)
- **eMail** - String (*Адреса електронної пошти контрагента*)
- **invite** - String (*Текст запрошення. Висилається в листі електронної пошти*)

Якщо запрошення для організації, що ще не зареєстрована в системі (не має orgID) то необхідно заповнити тільки параметри dealID, eMail, invite. Для запрошення зареєстрованої в системі організації, необхідно заповнити тільки параметри dealID та orgID.

Зразок запиту:

```
-> POST /dealsapi HTTP/1.1
Content-Type: application/json; charset=UTF-8
{"method":"invitePartner","params":{"orgID": 1000000010001}}
```

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'}

```

3.3.6 Додати документ(и) до угоди (addDocument)

Використовується для додавання документа (документів) до угоди

Параметри:

- **dealID** - Number (*Ідентифікатор угоди*)
- **document** - Object|Object[] (*Якщо додається декілька документів, то це має бути масив об'єктів*)
 - **name** - String(120) (*Назва документа*)
 - **requireConfirm** - Boolean (*Необхідно підтвердження документа партнером*)
 - **notifyWhenComplete** - Boolean (*Інформувати про закінчення процесу підписання документу за допомогою зворотного виклику. Налаштування адреси - на профілі організації*)
 - **docType** - String (*тип документа*)
 - **signers** - Sting[] Перелік логінів користувачів, які будуть назначені підписантами. Користувач повинен бути з тієї самої організації. По замовчанню підписант не визначений і може бути будь яким користувачем організації

Можливі значення для docType:

- TWO_SIDE_SIGNING - Підпис двох партнерів
- ONE_SIDE_SIGNING - Підпис автора
- INFORMATION – Інформаційний

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"] ]}}}
```

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'}
```

3.3.7 Завантажити вміст документа (setDocumentContent)

Використовується для завантаження вмісту документа

Параметри (Передаються в URL):

- **ID** - Ідентифікатор документа

Зразок запити:

```
-> 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'}
```

3.3.8 Додати підпис до документа (addDocumentSignature)

Використовується для додавання підписів зроблених в інших ситемах

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

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'}
```

3.3.9 Отримання документа з підписами (getDocument)

Використовується для вивантаження вмісту документа з накладеними ЕЦП.

Параметри:

- **ID** - Number (Ідентифікатор документу)
- **resultType** - String (тип результату)

Можливі значення для **resultType**

- DOCUMENT - Тільки документ

- SIGNATURES - Тільки підписи
- VISUALIZATION – Версія для друку
- SIGNATURES&VISUALIZATION – Підписи та версія для друку
- FULL - Документ підписи, та візуалізація

Повертає zip архів

Зразок запити:

```
-> 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"}
```

3.3.10 Отримання поточного стану документа в Системі (getDocumentInfo)

Використовується для отримання поточного стану документа в Системі.

Параметри:

- **ID - Number** (*Ідентифікатор документа*)

Можливі стани документу:

- REJECTED - Відхилено
- SIGNING - Підписання
- ACCEPTING – Прийняття
- REJECTED_INNER_ACCEPTING (Відхилений 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"}
```

3.3.11 Зворотній виклик за подіями в Системі

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

Зразок зворотного виклику

```
-> POST https://callback.Address.url 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.

Коди помилок:

```
{error: 10, description: 'Invalid parameters'}
```