

# Опис інтеграції з API (V2)

## Зміст

### 1. Загальна інформація та технічний опис API

- Авторизація
- Отримання прайсу
- Створення первинної потреби (замовлення)
- Підтвердження первинної потреби
- Перевірка статусу потреби
- Отримання ММО
- Запит на повернення товару
- Перевірка статусу повернення
- Отримання списку оброблених первинних потреб
- [Опис помилок](#)

### 2. Порядок підключення до API

#### 1. Загальна інформація та технічний опис API

Основні можливості API:

- Авторизація користувача
- Отримання прайсу
- Створення та підтвердження замовлень
- Перевірка статусів
- Отримання супровідних документів (ММО)
- Оформлення повернень

HTTP	Endpoint	Операція (operationId)	Короткий опис
POST	/auth	Auth	Авторизація
POST	/demandpost	DemandPost	Запит по потребі (створення первинної потреби)
PUT	/demandput	DemandPut	Підтвердження потреби — створення замовлення
POST	/demandresponsepost	DemandStatusPost	Встановлення статусу первинної потреби
PUT	/demandresponseput	DemandResponsePut	Зміна статусу ПП та оновлення інформації по товарах
GET	/demandstatusget	DemandGet	Отримання статусу ПП та списку доступних товарів
GET	/mmoget	MMOGet	Отримання файлу ММО
POST	/orderclaimpost	OrderClaimPost	Запит на повернення/обмін товару
GET	/orderclaimstatusget	OrderClaimGetStatus	Отримання статусу запиту на повернення
GET	/priceget	Price	Отримання прайсу
GET	/processeddemandget	ProcessedDemands	Отримання списку оброблених первинних потреб

Формати: JSON / XML

Аутентифікація: JWT Bearer Token (через метод /auth)

API доступний виключно через HTTPS

## Авторизація

Метод: POST /auth

Призначення: Отримання токена доступу (SessionID) на основі логіна та пароля, який надає постачальник. Попередньо для надання доступу до тестового середовища потрібно надати список IP адрес, з яких буде відбуватись підключення.

Токен отримується виконуючи метод auth. При кожному іншому запити токен потрібно вказувати в заголовках bearerAuth Token

Дія токена 15хв, якщо викликається метод demandput, токен подовжується ще на 15хв.

Вхідні дані: MessageID, SourceID, Login, Password

Вихідні дані: SessionID, Expires

Код відповіді: 200 (успіх), 401 (невірні дані)

### Запит в форматі JSON:

Параметр	Тип даних	Обов.	Опис
MessageID	string [50]	1..1	Унікальний ідентифікатор повідомлення
SourceID	string [50]	1..1	Унікальний ідентифікатор системи джерела повідомлення (назва системи)
Login	string [25]	1..1	Логін
Password	string [15]	1..1	Пароль

### Відповідь в форматі JSON:

Параметр	Тип даних	Обов.	Опис/Джерело
SessionID	string	1..1	ID сесії Одноіменний параметр з відповіді системи
Expires	string	1..1	Час кінця сесії Одноіменний параметр з відповіді системи Формат: 2024-04-23T10:47:02

### Приклад запиту (POST /auth)

```
{
  "MessageID": "12345",
  "SourceID": "TEST_APP",
  "Login": "user@example.com",
  "Password": "securePassword123"
}
```

### Приклад відповіді (POST /auth)

```
{
  "SessionID": "abcdef1234567890",
  "Expires": "2024-04-23T10:47:02"
}
```

## Отримання прайсу

Метод: GET /priceget

Сервіс призначений для отримання прайсу

Параметри: MessageID, SourceID, ClientCodeEDRPOU, WarehouseList (необов'язково), ResponseFormat

Вихід: список товарів із цінами, ПДВ, серією та строком придатності

#### Query Param запити:

Параметр	Обов.	Опис
MessageID	1..1	Унікальний ідентифікатор повідомлення Максимальна кількість символів – 50
SourceID	1..1	Унікальний ідентифікатор системи джерела повідомлення (назва системи) Максимальна кількість символів – 50
ClientCodeEDRPOU	1..1	Код ЕДРПОУ клієнта Максимальна кількість символів – 12
WarehouseList	0..1	Список кодів склада через кому Приклад: 0001, 0002 Максимальна кількість символів – 150
ResponseFormat	0..1	Необхідний формат відповіді сервісу Допустимі значення: "json" – для отримання відповіді в JSON "xml" – для отримання відповіді в XML Якщо параметр буде відсутній – по замовчанню приймається значення "json" Максимальна кількість символів – 4

Приклад GET запити

*GET /priceget?MessageID=2c6b53f1-84b6-4e49-a04b-ba40e9b4e878&SourceID=SystemName&ClientCodeEDRPOU=12345678&WarehouseList=0001&comma;0002&comma;0003&ResponseFormat=json*

**Відповідь в форматі JSON** (якщо у вхідному запиті ResponseFormat=json):

Струк.	Параметр	Тип даних	Обов.	Опис/Джерело
<b>1</b>	<b>Items</b>	<b>array</b>	<b>0..1</b>	
<b>1.1</b>		<b>object</b>	<b>0..n</b>	
1.1.1	ItemID	string	0..1	Ідентифікатор SKU Одноіменний параметр з відповіді системи
1.1.2	Product	string	0..1	Назва продукту Одноіменний параметр з відповіді системи
1.1.3	Unit	string	0..1	Одиниця виміру Одноіменний параметр з відповіді системи
1.1.4	ProducerCode	string	0..1	Код продукту виробника Параметр <b>Producer_Code</b> з відповіді системи
1.1.5	Producer	string	0..1	Назва виробника Одноіменний параметр з відповіді системи
1.1.6	Price	number	0..1	Ціна Одноіменний параметр з відповіді системи
1.1.7	VAT	number	0..1	Ставка ПДВ Одноіменний параметр з відповіді системи
1.1.8	PriceiVAT	number	0..1	Ціна з ПДВ Параметр <b>Price_iVAT</b> з відповіді системи
1.1.9	BestBefore	string	0..1	Термін придатності Одноіменний параметр з відповіді системи
1.1.10	MorionID	string	0..1	Код Моріон Одноіменний параметр з відповіді системи
1.1.11	BarCode	string	0..1	Штрих код Одноіменний параметр з відповіді системи
1.1.12	RegNo	string	0..1	Реєстраційний номер Одноіменний параметр з відповіді системи
1.1.13	RegDate	string	0..1	Дата реєстрації Одноіменний параметр з відповіді системи
1.1.14	RegExpired	string	0..1	Термін реєстрації Одноіменний параметр з відповіді системи

1.1.15	PaymentDeferral	number	0..1	Відтермінування платежу(кількість днів) з відповіді системи
--------	-----------------	--------	------	---

### Приклад відповіді Отримання прайсу (GET /priceset)

```
{
  "Items": [
    {
      "ItemID": "ПР00031520",
      "Product": "Амінол р-н 200мл",
      "Unit": "пляшка",
      "ProducerCode": "ПР00031520",
      "Producer": "ТОВ ААА",
      "Price": 566.42,
      "VAT": 7,
      "PriceiVAT": 606.07,
      "BestBefore": "2024-04-23",
      "MorionID": "33374",
      "BarCode": "4820023312937",
      "RegNo": "UA/14166/01/01",
      "RegDate": "2024-04-23",
      "RegExpired": "2024-04-29",
      "PaymentDeferral": 90
    }
  ]
}
```

## Створення первинної потреби (замовлення)

Метод: POST /demandpost

Сервіс призначений для передачі первинної потреби (ПП) з системи клієнта.

Тіло: MessageID, SourceID, ClientOrderID, ClientCodeEDRPOU, ConsigneeCode, SupplierCodeEDRPOU, DemandDate, DeliveryDate, Items

Кожен item містить: ItemID, Quantity

### Запит в форматі JSON:

Струк.	Параметр	Тип даних	Обов.	Опис
1	MessageID	string [50]	1..1	Унікальний ідентифікатор повідомлення
2	SourceID	string [50]	1..1	Унікальний ідентифікатор системи джерела повідомлення (назва системи)
3	ClientOrderID	string [50]	1..1	Номер замовлення клієнта
4	ClientCodeEDRPOU	string [12]	1..1	Код ЕДРПОУ клієнта
5	ConsigneeCode	string [128]	1..1	Код грузоотримувача
6	SupplierCodeEDRPOU	string [12]	1..1	Код ЄДРПОУ постачальника
7	DemandDate	string [19]	1..1	Дата замовлення 2024-04-23T10:47:02
8	DeliveryDate	string [19]	0..1	Бажана дата доставки 2024-04-23T10:47:02
9	Items	array	1..1	
9.1		object	1..n	
9.1.1	ItemID	string [12]	1..1	Ідентифікатор SKU, отримується окремо (прайс)
9.1.2	Quantity	number	1..1	Кількість потреби

### Відповідь в форматі JSON:

Струк.	Параметр	Тип даних	Обов.	Опис/Джерело
1	ClientOrderID	string	0..1	Номер замовлення клієнта Одноіменний параметр з відповіді системи
2	DemandID	string	0..1	ID замовлення Одноіменний параметр з відповіді системи
3	DemandNumber	string	0..1	Номер замовлення Одноіменний параметр з відповіді системи
4	Status	string	0..1	Статус первинної потреби Одноіменний параметр з відповіді системи

### Приклад запиту (POST /demandpost):

```
{
  "MessageID": "762511b6-d056-4fe7-9e4b-ef6e37f918e2",
  "SourceID": "Smth system name",
  "ClientOrderID": "1234",
  "ClientCodeEDRPOU": "100500600",
  "ConsigneeCode": "1234",
  "SupplierCodeEDRPOU": "100500600",
  "DemandDate": "2024-04-23T10:47:02",
  "DeliveryDate": "2024-04-23T10:47:02",
  "Items": [
    {
      "ItemID": "33374",
      "Quantity": 25
    }
  ]
}
```

### Приклад відповіді Створення первинної потреби (POST /demandpost)

```
{
  "ClientOrderID": "1234",
  "DemandID": "c80ae563-b67a-4b15-bfb4-2da208d33b54",
  "DemandNumber": "МТК00001234",
  "Status": "Прийнята"
}
```

## Підтвердження первинної потреби

Метод: PUT /demandput

Сервіс призначений для підтвердження клієнтом ПП

Тіло: MessageID, SourceID, DemandID, DemandNumber, ClientOrderID, ClientCodeEDRPOU

Повертає статус підтвердження: "Підтверджена"

### Запит в форматі JSON:

Струк.	Параметр	Тип даних	Обов.	Опис
1	MessageID	string [50]	1..1	Унікальний ідентифікатор повідомлення
2	SourceID	string [50]	1..1	Унікальний ідентифікатор системи джерела повідомлення (назва системи)
3	DemandID	string [36]	1..1	ID замовлення
4	DemandNumber	string [9]	1..1	Номер замовлення

5	ClientOrderID	string [50]	1..1	Номер замовлення клієнта
6	ClientCodeEDRPOU	string [12]	1..1	Код ЕДРПОУ клієнта

### Відповідь в форматі JSON:

Струк.	Параметр	Тип даних	Обов.	Опис/Джерело
1	ClientOrderID	string	0..1	Номер замовлення клієнта Одноіменний параметр з відповіді системи
2	DemandID	string	0..1	ID замовлення Одноіменний параметр з відповіді системи
3	DemandNumber	string	0..1	Номер замовлення Одноіменний параметр з відповіді системи
4	Status	string	0..1	Статус первинної потреби Одноіменний параметр з відповіді системи

### Приклад запиту (PUT /demandput):

```
{
  "MessageID": "762511b6-d056-4fe7-9e4b-ef6e37f918e2",
  "SourceID": "Smth system name",
  "DemandID": "c80ae563-b67a-4b15-bfb4-2da208d33b54",
  "DemandNumber": "1234",
  "ClientOrderID": "1234",
  "ClientCodeEDRPOU": "100500600"
}
```

### Приклад відповіді Підтвердження потреби (PUT /demandput)

```
{
  "ClientOrderID": "1234",
  "DemandID": "c80ae563-b67a-4b15-bfb4-2da208d33b54",
  "DemandNumber": "МТК00001234",
  "Status": "Підтверджена"
}
```

## Перевірка статусу потреби

Метод: GET /demandstatusget

Сервіс призначений для отримання статусу ПП та переліку наявних товарів

Параметри: MessageID, SourceID, DemandID, ResponseFormat

Вихід: статус потреби, перелік позицій, підтверджені кількості та ціни

Статуси потреби: "Прийнята", "Оброблена", "Підтверджена", "Не підтверджена", «Створено замовлення», «Помилка»).

### Query Param запиту:

Параметр	Обов.	Опис
MessageID	1..1	Унікальний ідентифікатор повідомлення Максимальна кількість символів – 50
SourceID	1..1	Унікальний ідентифікатор системи джерела повідомлення (назва системи) Максимальна кількість символів – 50
DemandID	1..1	ID замовлення Максимальна кількість символів – 36
ResponseFormat	0..1	Необхідний формат відповіді сервісу

	Допустимі значення: "json" – для отримання відповіді в JSON "xml" – для отримання відповіді в XML Якщо параметр буде відсутній – по замовчанню приймається значення "json" Максимальна кількість символів – 4
--	---

### Приклад GET запити

GET /demandstatusget?MessageID=2c6b53f1-84b6-4e49-a04b-ba40e9b4e878&SourceID=SystemName&DemandID=1234&ResponseFormat=json

Відповідь в форматі JSON (якщо у вхідному запиті ResponseFormat=json):

Струк.	Параметр	Тип даних	Обов.	Опис/Джерело
1	DemandID	string	1..1	ID замовлення
2	DemandNumber	string	1..1	Номер замовлення
3	ClientOrderID	string	1..1	Номер замовлення клієнта
4	ClientID	string	1..1	Унікальний ID клієнта Значення <b>ClientID</b> з відповіді системи з сервісу GetSession
5	ClientCodeEDRPOU	string	1..1	Код ЕДРПОУ клієнта
6	ConsigneeCode	string	1..1	Код грузоотримувача
7	DemandDate	string	1..1	Дата замовлення 2024-04-23T10:47:02
8	DeliveryDate	string	0..1	Бажана дата доставки 2024-04-23T10:47:02
9	DemandExpirationDate	string	1..1	Дата 2024-04-23T10:47:02
10	Status	string	1..1	Статус первинної потреби
11	Items	array	0..1	Передається для випадків коли STATUS «Прийнята»
11.1		object	1..n	
11.1.1	ItemID	string	1..1	Ідентифікатор SKU, отримується окремо (прайс)
11.1.2	Quantity	number	1..1	Кількість потреби
11.1.3	ConfirmedQuantity	number	1..1	Підтверджена кількість
11.1.4	Price	number	1..1	Ціна
11.1.5	Series	string	1..1	Серія
11.1.6	SeriesExpirationDate	string	1..1	Термін придатності 2024-04-23

### Приклад відповіді Перевірка статусу потреби (GET /demandstatusget)

```
{
  "DemandID": "c80ae563-b67a-4b15-bfb4-2da208d33b54",
  "DemandNumber": "1234",
  "ClientOrderID": "1234",
  "ClientID": "ee301547-111b-4ade-8cbb-ff64e684f998",
  "ClientCodeEDRPOU": "100500600",
  "ConsigneeCode": "1234",
  "DemandDate": "2024-04-23T10:47:02",
  "DeliveryDate": "2024-04-23T10:47:02",
  "DemandExpirationDate": "2024-04-23T10:47:02",
  "Status": "Оброблена",
  "Items": [
    {
      "ItemID": "33374",
      "Quantity": 25,

```

```

    "ConfirmedQuantity": 25,
    "Price": 25,
    "Series": "25/10-000000",
    "SeriesExpirationDate": "2024-04-23"
  }
]
}

```

## Отримання ММО

Метод: GET /mmoget

Сервіс призначений для отримання файлу ММО.

Параметри: MessageID, SourceID, ClientOrderID, ClientCodeEDRPOU

Вихід: файл ММО (у base64) та його ім'я

### Query Param запити:

Параметр	Обов.	Опис
MessageID	1..1	Унікальний ідентифікатор повідомлення Максимальна кількість символів – 50
SourceID	1..1	Унікальний ідентифікатор системи джерела повідомлення (назва системи) Максимальна кількість символів – 50
ClientOrderID	1..1	Номер замовлення клієнта Максимальна кількість символів – 50
ClientCodeEDRPOU	1..1	Код ЕДРПОУ клієнта Максимальна кількість символів – 12
ResponseFormat	0..1	Необхідний формат відповіді сервісу Допустимі значення: "json" – для отримання відповіді в JSON "xml" – для отримання відповіді в XML Якщо параметр буде відсутній – по замовчанню приймається значення "json" Максимальна кількість символів – 4

### Приклад GET запити

GET /mmoget?MessageID=2c6b53f1-84b6-4e49-a04b-ba40e9b4e878&SourceID=SystemName&ClientOrderID=1234&ClientCodeEDRPOU=12345678&ResponseFormat=json

Відповідь в форматі JSON (якщо у вхідному запиті ResponseFormat=json):

Струк.	Параметр	Тип даних	Обов.	Опис/Джерело
1	Items	array	0..1	
1.1		object	0..n	
1.1.1	InvoiceNumber	string	0..1	Номер видаткової накладної Одноіменний параметр з відповіді системи
1.1.2	ClientOrderID	string	0..1	Номер замовлення клієнта Одноіменний параметр з відповіді системи
1.1.3	MMOData	string	0..1	Дані файлу ММО в base64-бінарні дані Одноіменний параметр з відповіді системи
1.1.4	MMOFileName	string	0..1	Ім'я файлу ММО Одноіменний параметр з відповіді системи

### Приклад відповіді Отримання ММО (GET /mmoget)

```

{
  "Items": [

```

```

{
  "InvoiceNumber": "1234",
  "ClientOrderID": "1234",
  "MMOData": "string",
  "MMOFileName": "string"
}
]
}

```

## Запит на повернення товару

Метод: POST /orderclaimpost

Сервіс призначений для отримання запиту на повернення товару

Тіло: MessageID, SourceID, ClientOrderClaimID, ClientOrderID, ClientCodeEDRPOU, InvoiceNumber, OrderClaimType, Items

Кожен item містить: ItemID, Quantity, причина, ExpirationDate, файл у base64

### Запит в форматі JSON:

Струк.	Параметр	Тип даних	Обов.	Опис
1	MessageID	string [50]	1..1	Унікальний ідентифікатор повідомлення
2	SourceID	string [50]	1..1	Унікальний ідентифікатор системи джерела повідомлення (назва системи)
3	ClientOrderClaimID	string [50]	0..1	Номер запиту на повернення клієнта
4	ClientOrderID	string [50]	1..1	Номер замовлення клієнта
5	ClientCodeEDRPOU	string [12]	1..1	Код ЕДРПОУ клієнта
6	InvoiceNumber	string [25]	1..1	Номер Видаткової накладної
7	OrderClaimType	string [20]	1..1	Офіційне, пересорт, брак
<b>8</b>	<b>Items</b>	<b>array</b>	<b>1..1</b>	
<b>8.1</b>		<b>object</b>	<b>1..n</b>	
8.1.1	ItemID	string [12]	1..1	Ідентифікатор SKU, отримується окремо (прайс)
8.1.2	QuantityOnInvoice	number	1..1	кількість по накладній
8.1.3	QuantityAccepted	number	1..1	кількість прийнятого товару
8.1.4	ClaimQuantity	number	1..1	кількість на повернення/претензія
8.1.5	Price	number	0..1	Ціна
8.1.6	Comments	string [200]	1..1	Короткий опис причини повернення
8.1.7	ExpirationDate	string [10]	1..1	Термін придатності 2024-04-23
8.1.8	FileData	string	1..1	Прикріплений файл в base64-бінарні дані
8.1.9	FileName	string [1024]	1..1	Ім'я файлу

### Відповідь в форматі JSON:

Струк.	Параметр	Тип даних	Обов.	Опис/Джерело
1	ClientOrderClaimID	string	0..1	Номер запиту на повернення клієнта Одноіменний параметр з відповіді системи
2	OrderClaimID	string	0..1	ID запиту на повернення Одноіменний параметр з відповіді системи
3	Status	string	0..1	Статус запиту на повернення Одноіменний параметр з відповіді системи

**Приклад запиту (POST /orderclaimpost):**

```
{  
  
  "MessageID": "762511b6-d056-4fe7-9e4b-ef6e37f918e2",  
  "SourceID": "Smth system name",  
  "ClientOrderClaimID": "1234",  
  "ClientOrderID": "1234",  
  "ClientCodeEDRPOU": "100500600",  
  "InvoiceNumber": "1234",  
  "OrderClaimType": "брак",  
  "Items": [  
    {  
      "ItemID": "33374",  
      "QuantityOnInvoice": 25,  
      "QuantityAccepted": 24,  
      "ClaimQuantity": 1,  
      "Price": 25,  
      "Comments": "Пошкоджено пакування",  
      "ExpirationDate": "2024-04-23",  
      "FileData": "string",  
      "FileName": "Видаткова_накладнаМТК0000001.mmo"  
    }  
  ]  
}
```

**Приклад відповіді Запит на повернення товару (POST /orderclaimpost)**

```
{  
  "ClientOrderClaimID": "1234",  
  "OrderClaimID": "c80ae111-b111-4b11-b111-2da208d33b54",  
  "Status": "Прийнято"  
}
```

## Перевірка статусу повернення

Метод: GET /orderclaimstatusget

Сервіс призначений для отримання статусу запиту на повернення товару

Параметри: MessageID, SourceID, OrderClaimID, ClientCodeEDRPOU

Вихід: статус повернення, рішення, позиції

**Query Param запиту:**

Параметр	Обов.	Опис
MessageID	1..1	Унікальний ідентифікатор повідомлення Максимальна кількість символів – 50
SourceID	1..1	Унікальний ідентифікатор системи джерела повідомлення (назва системи) Максимальна кількість символів – 50
OrderClaimID	1..1	Ідентифікатор запиту на повернення Максимальна кількість символів – 36
ClientCodeEDRPOU	1..1	Код ЕДРПОУ клієнта Максимальна кількість символів – 12
ResponseFormat	0..1	Необхідний формат відповіді сервісу Допустимі значення:

	"json" – для отримання відповіді в JSON "xml" – для отримання відповіді в XML Якщо параметр буде відсутній – по замовчанню приймається значення "json" Максимальна кількість символів – 4
--	--

#### Приклад GET запити

GET /orderclaimstatusget?MessageID=2c6b53f1-84b6-4e49-a04b-ba40e9b4e878&SourceID=SystemName&OrderClaimID=1234&ClientCodeEDRPOU=12345678&ResponseFormat=json

Відповідь в форматі JSON (якщо у вхідному запиті ResponseFormat=json):

Струк.	Параметр	Тип даних	Обов.	Опис/Джерело
1	ClientOrderClaimID	string	1..1	Номер запити на повернення клієнта Одноіменний параметр з відповіді системи
2	OrderClaimID	string	0..1	ID запити на повернення Одноіменний параметр з відповіді системи
3	Status	string	0..1	Статус запити на повернення Одноіменний параметр з відповіді системи
4	Items	array	1..1	
4.1		object	1..n	
4.1.1	ItemID	string	1..1	Ідентифікатор номенклатури Одноіменний параметр з відповіді системи
4.1.2	Quantity	number	1..1	Кількість на повернення Одноіменний параметр з відповіді системи
4.1.3	Decision	string	1..1	Рішення по запити Одноіменний параметр з відповіді системи

#### Приклад відповіді Перевірка статусу повернення (GET /orderclaimstatusget)

```
{
  "ClientOrderClaimID": "1234",
  "OrderClaimID": "c80ae111-b111-4b11-b111-2da208d33b54",
  "Status": "Погоджено",
  "Items": [
    {
      "ItemID": "33374",
      "Quantity": 25,
      "Decision": "Обмін товару"
    }
  ]
}
```

## Отримання списку оброблених первинних потреб

Метод: GET /processeddemandget

Сервіс призначений для отримання списку оброблених первинних потреб

Параметри: MessageID, SourceID, OrderClaimID, ClientCodeEDRPOU

#### Query Param запити:

Параметр	Обов.	Опис
MessageID	1..1	Унікальний ідентифікатор повідомлення Максимальна кількість символів – 50
SourceID	1..1	Унікальний ідентифікатор системи джерела повідомлення (назва системи) Максимальна кількість символів – 50
ResponseFormat	0..1	Необхідний формат відповіді сервісу Допустимі значення: "json" – для отримання відповіді в JSON

Параметр	Обов.	Опис
		"xml" – для отримання відповіді в XML Якщо параметр буде відсутній – по замовчанню приймається значення "json" Максимальна кількість символів – 4

Приклад GET запиту

```
GET /processeddemandget?MessageID=2c6b53f1-84b6-4e49-a04b-ba40e9b4e878&SourceID=SystemName&ResponseFormat=json
```

**Відповідь в форматі JSON** (якщо у вхідному запиті ResponseFormat=json):

Струк.	Параметр	Тип даних	Обов.	Опис/Джерело
1	Items	array	0..1	
1.1		object	0..n	
1.1.1	DemandID	string	0..1	ID замовлення
1.1.2	DemandNumber	string	0..1	Номер замовлення
1.1.3	ClientOrderID	string	0..1	Номер замовлення клієнта
1.1.4	ClientCodeEDRPOU	string	0..1	Код ЕДРПОУ клієнта

## Опис помилок

№	Опис помилки	Значення ErrorDescription	HTTP Code
1	Помилка обробки повідомлення	Помилка обробки повідомлення	500
2	Помилка валідації вхідного запиту	Помилка валідації вхідного повідомлення	400
3	Запис в БД з таким DemandID вже існує	Помилка обробки повідомлення. Оригінальна помилка при запиті до БД: ...	500
4	Не знайдена первинна потреба з таким DemandID в БД	Не знайдена первинна потреба з таким DemandID	500
5	Токен доступу відсутній або недійсний	Токен доступу відсутній або недійсний	401
6	Запит відрізняється від зазначеного в ТЗ (Swagger)	Недозволений метод запиту	405
7	Отримано запит з форматом не XML і не JSON	Недозволений тип повідомлення	415
8	Не змогли перевірити вміст запиту через якісь причини	Помилка перевірки вхідного повідомлення на заборонений вміст	400
9	Знайдено SQL ін'єкцію у вхідному запиті	Помилка, повідомлення містить заборонений вміст	418

Більш детальний опис відповідей з можливими помилками можна знайти в swagger.

## 2. Порядок підключення до API

- Ініціювання підключення:** необхідно направити запит на підключення на електронну адресу: [IT@uf.ua](mailto:IT@uf.ua)  
У листі бажано зазначити:
  - назву компанії
  - контакти технічного спеціаліста для взаємодії
- Надання IP-адрес та отримання доступів:** з метою забезпечення безпеки доступу необхідно надати перелік IP-адрес, з яких буде здійснюватися доступ до API. Після

погодження IP-адреси будуть додані до списку дозволених. Після обробки запиту замовнику надається доступ до тестового середовища та параметри авторизації.

3. **Надання переліку ID вантажоотримувачів:** для коректної інтеграції необхідно надати перелік ID вантажоотримувачів, які будуть використовуватися в API-запитах. Це дозволить активувати відповідні об'єкти в системі, перевірити коректність налаштувань та уникнути помилок при тестуванні.
4. **Тестування обраних методів API:** після отримання доступів замовник здійснює тестування обраних методів API у тестовому середовищі.

Рекомендується:

- перевірити авторизацію
- протестувати базові методи
- перевірити коректність обробки запитів та відповідей

Після успішного тестування здійснюється перехід у продуктивне середовище.

**Для налагодження взаємодії по API ви можете звернутись на пошту [it@uf.ua](mailto:it@uf.ua)**