

## **Інструкція по API Для користувачів**

Дата	Опис
13.07.2022	Додано опис параметрів PayerID (п. 7.4.3, 8.6.3, 17.3) та параметрів ConsigneeID, ConsigneeID (п. 8.6.3)
22.06.2022	Внесено зміни у формат вивантаження наявності позицій по складам (п. 1.4) та їх опис (п. 1.5), додано опис критерія Ware(п. 1.3)
07.06.2022	Внесено зміни у формат вивантаження наявності позицій по складам (п. 1.4), старий формат буде вилючено. Додано опис обов'язкового параметру WarehouseID (п. 7.4.3, 8.3.3, 8.6.3, 17.3). Розділи 7.5 «Створення замовлення в режимі «самовивіз» для уціненого товару» та 8.7 «Створення замовлення в режимі «Доставка» уціненого товару» <i>тимчасово вилучені.</i>
11.05.2022	Змінено адресу сервісу для отримання замовлення товару Microsoft ESD (п. 15.2) та додано інформацію, що замовлення повинно містити тільки ESD позиції (п. 15.1)
05.05.2022	Додано опис полів warehouse95, warehouse96(п.1.5)
24.01.2022	Внесено зміни в опис поля warehouse1, поле amount видалено (п. 1.5)
21.12.2021	Внесено зміни в опис критерія OnlyFree (п. 1.3), поля warehouse1 (п. 1.5)
02.12.2021	Оновлено опис полів warehouse1,warehouse5,warehouse44,warehouse55; видалені поля exist, stock, avail (п.1.5)
26.11.2021	Оновлено опис критерію OnlyFree (1.3)
08.10.2021	Внесено доповнення опису критерія amount (п.1.5)
21.07.2021	Додано опис критерія isMarked (п.1.5)
23.06.2021	Додано опис критеріїв isNewWare, isAlreadyExistWare (п. 1.5)
12.02.2021	Оновлено формат виведення результатів, а саме додано параметр serialControl (пп. 1.4, 1.5). Оновлено перелік вхідних даних для отримання курсу валют (п. 2.3). Додано параметр PaperworkCurrencyRate(п.п. 7.4.5, 7.5.5, 8.6.5, 8.7.5, 15.5, 17.5), оновлено формат виведення результатів (пп. 7.4.4, 8.6.4, 15.4, 17.4)
04.02.2021	Додано параметр IsDDP (п.п. 7.4.5, 7.5.5, 8.6.5, 8.7.5, 9.5, 17.5), оновлено формат виведення результатів (пп. 7.4.4, 8.6.4, 9.4, 17.4)
20.01.2021	Оновлено формат виведення результатів, а саме додано параметри bar, upc (пп. 1.4, 1.5)
23.12.2020	Внесено уточнення по опису параметра tnvd (пп. 1.5, 12.5, 13.5)
07.12.2020	Оновлено перелік вхідних даних для створення замовлення (пп. 8.6.3, 8.7.3).
30.11.2020	Оновлено перелік вхідних даних для створення замовлення (пп. 8.6.3, 8.7.3, 17.3).
05.11.2020	Додано параметр DocExtNum(п.п. 7.4.3, 7.5.3, 8.6.3, 8.7.3, 12.3, 15.3, 17.3), оновлено формат виведення результатів (пп. 7.4.4, 8.6.4, 15.4, 16.4, 17.4)
30.10.2020	Додано параметр Version (п.п. 7.4.3, 7.5.3, 8.6.3, 8.7.3, 15.3, 16.3, 17.3), оновлено формат виведення результатів (пп. 7.4.4, 7.5.4, 8.6.4, 8.7.4, 15.4,

	16.4, 17.4)
26.08.2020	Додано метод отримання договору (п. 7.3), оновлено перелік вхідних даних для створення замовлення (пп.7.4.3, 7.5.3, 8.6.3, 8.7.3, 15.3, 17.3). Пункти 7.3 та 7.4 версії документа 1.4 перенумеровані в 7.4 і 7.5 відповідно.
28.07.2020	Додано новий спосіб замовлення товару по схемі «Дропшипінг» (розділ 17)
01.07.2020	Доданий параметр DiscountType (п. 4.3) Змінено тип параметра Specs Змінено приклади виклику для замовлень в режимі «Самовивіз» (п. 7.3.6) і «Доставка» (п. 8.6.6) Додана можливість замовлення уціненого товару в режимі «Самовивіз» (п. 7.4) і «Доставка» (п. 8.7)
02.06.2020	Внесено зміни в методику створення замовлень в режимі «Доставка» (розділ 8)
29.05.2020	Внесено уточнення по опису параметра amount (п. 1.5)
15.05.2020	Створення документа

## Зміст

Загальні відомості .....	5
Web-методи.....	6
1. Каталог товарів .....	6
2. Курси валют .....	14
3. Активні акції.....	16
4. Пошкоджений товар .....	18
5. Список кодів вендорів .....	21
6. Код категорії.....	23
7. Створення замовлень в режимі «Самовивіз» .....	25
7.1 Отримання списку вантажоодержувачей з ПН .....	25
7.2 Отримання списку платників/вантажоотримувачів юр. осіб .....	26
7.3 Отримання списку договорів .....	27
7.4 Створення замовлення в режимі «самовивіз» .....	29
8. Створення замовлень в режимі «Доставка» .....	33
8.1 Отримання списку адрес партнера з прив'язаним контактом.....	34
8.2 Отримання списку вантажоодержувачей партнера з ПН.....	35
8.3 Отримання списку рейсів .....	36
8.4 Отримання списку платників/вантажоотримувачів юр. осіб .....	37
8.5 Отримання списку способів оплати доставки .....	38
8.6 Створення замовлення в режимі «Доставка».....	39
9. Отримання замовлень .....	44
10. Резерви .....	49
11. Відвантажувальні накладні.....	53
12. Бухгалтерські рахунки-фактури .....	57
13. Бухгалтерські видаткові накладні.....	62
14. Акт зміни ціни.....	67
15. Замовлення товару Microsoft ESD .....	71
16. Отримання ключа товару Microsoft ESD .....	74
17. Створення замовлення для доставки товару з зазначенням ТТН (дропшипінг) .....	77
Додаткова інформація .....	83

## Загальні відомості

Цей документ використовується для опису методів (API-методи), за допомогою яких здійснюються запити і повертаються відповіді для кожної операції. Всі методи описуються з вхідними параметрами і повертають шаблон результату.

API - потужний інструмент, який дозволяє інтегруватися в бізнес-процес закупівель і працювати із замовленнями, товарами, а також безпосередньо контролювати свій баланс. API інтерфейс використовується для того, щоб інтегрувати можливості сервісу ERC в облікову систему клієнта.

Використовуючи API, користувач може:

- отримувати актуальні дані про покупки, інформацію про замовлення, прогнозах надходження товарів, історію і відстежувати свої замовлення на всіх етапах бізнес-процесів;
- бути в курсі реального стану складу (рівня цін (спец-прайс)), отримувати повні дані про продукти (характеристики товару, опис та ін.);
- організувати ефективну взаємодію з головним офісом.

### Примітки:

- Для доступу до сервісів API потрібно звернутися до Ведучого дилер-менеджерів ERC.
- Для входу в систему використовуються ті ж електронна пошта і пароль, що користувач використовує для входу на портал B2B.
- Користувачі повинні надати Ведучому дилер-менеджеру в ERC статичну IP-адресу, з якої будуть виконуватися запити.

## Web-методи

### 1. Каталог товарів

#### 1.1 Опис

InfoType = 6. Цей метод дозволяє отримувати дані про кількість товару, в розрізі складів і резерви по товару, наявності новинок і повторної доступності, а також відстежувати рух товару (в дорозі, розвантажуються).

#### 1.2 Адреса сервіса

<https://connect.erc.ua/connectservice/api/specprice/DoExport>

#### 1.3 Вхідні параметри

Найменування	Тип	Опис
Email	String	Email користувача
Pass	String	Пароль користувача
InfoType	Int	Тип експорту: 6
IsJson	Boolean	Не обов'язковий параметр (опціонально - при відсутності результат буде вивантажено в XML). Можливі значення: 0, 1, true, false
VendorId	Int	Код вендора (опціонально). Отримання списку кодів вендорів див. розділ 5
CategoryId	Int	Код категорії (опціонально). Отримання списку кодів категорій див. розділ 6
IsNew	Boolean	Вивантажувати тільки нові позиції (опціонально). Можливі значення: 0,1, true, false
OnlyFree	Int	Вивантажувати тільки доступні для замовлення позиції (опціонально), не виводить позиції не доступні до відвантаження з 1-го, 5-го, 95-го та 96-госкладів. Можливі значення: 0,1, true, false
IsAction	Boolean	Вивантажувати тільки позиції, які беруть участь в акціях (опціонально). Можливі значення: 0,1, true, false
Ware	String	Вивантажує результат для вказаної позиції (опціонально).
Lang	String	Мова, на якій буде вивантажено результат (за замовчуванням - українська). Можливі значення: ru, ua, ru-ru, uk-ua (без урахування

регістру)

## Приклад JSON параметрів

```
{
  "Email": "user@email.com",
  "Pass": "*****",
  "Infotype": 6,
  "IsNew": 1,
  "VendorId": "018",
  "lang": "ua-uk"
}
```

## 1.4 Приклад даних, що повертаються

Результат доступний в XML і в JSON форматі.

Для XML

```
<ercsprice datetime="Ср май 06 03:25:44 ЕЕТ 2020">
  <vendor name="2E Computer peripherals">
    <goods>
      <category>Комп`ютерна периферія </category>
      <subcategory>Клавіатури бездротові</subcategory>
      <gname>Клавіатура 2E KS210 Slim WL Black</gname>
      <code>2E-KS210WB</code>
      <sprice>XXX,XX</sprice>
      <ddp>1</ddp>
      <vat>20</vat>
      <pic>http://www.erc.ua/i/goods/2E-KS210WB.jpg</pic>
      <ean />
    </goods>
  </vendor name>
</ercsprice>
<bar />
<upc>856963007293</upc>
  <declar>UA.TR.052.D.502-19</declar>
  <tnvd>8471 60 60 00</tnvd>
  <cert />
  <warr>12</warr>
  <warrunit>2</warrunit>
  <width>450</width>
  <height>160</height>
  <depth>30</depth>
  <weight>0,6</weight>
  <volume>0,00216</volume>
  <RRP_UAH>XXX,00</RRP_UAH>
  <monitor>True</monitor>
</isMarked>1</isMarked>
  <isaction>0</isaction>
</serialControl>False</serialControl>
  <country>Китай</country>
  <whs>
    <wh id="1" q="0" />
    <wh id="5" q="0" />
    <wh id="44" q="0" />
    <wh id="55" q="0" />
    <wh id="95" q="0" />
    <wh id="96" q="&gt;500" />
  </whs>
</>
```

```

<reserve>1</reserve>
<quantityinbox>20</quantityinbox>
<isNewWare>False</isNewWare>
<isAllreadyExistWare>False</isAllreadyExistWare>
<isNotmakingWare>False</isNotmakingWare>
<isTransitU>False</isTransitU>
<isTransitW>False</isTransitW>
<isTransity>False</isTransity>
<actions />
</goods>

```

## ДляJSON

```

[
  {
    "vendorId": "0520001",
    "vendor": "2E Computer peripherals",
    "categoryId": "0008",
    "category": "Комп`ютерна периферія ",
    "subCategoryId": "00080001",
    "subcategory": "Клавіатури бездротові",
    "code": "2E-KS210WB",
    "gname": "Клавіатура 2E KS210 Slim WL Black",
    "sprice": XXX.XX,
    "ddp": 1,
    "vat": 20.0,
    "pic": "http://www.erc.ua/i/goods/2E-KS210WB.jpg",
    "ean": "",
    "bar": "",
    "upc": "856963007293",
    "declar": "UA.TR.052.D.502-19",
    "tnvd": "8471 60 60 00",
    "cert": null,
    "warr": 12,
    "warrunit": "2",
    "width": 450,
    "height": 160,
    "depth": 30,
    "weight": 0.6,
    "volume": 0.00216,
    "RRP_UAH": XXX.X,
    "monitor": true,
    "isMarked": 1,
    "isaction": 0,
    "serialControl": false,
    "country": "Китай",
    "whs": [
      {
        "id": 1,
        "q": "0"
      },
      {
        "id": 5,
        "q": "0"
      }
    ]
  }
]

```



```

    },
    {
      "id": 44,
      "q": "0"
    },
    {
      "id": 55,
      "q": "0"
    },
    {
      "id": 95,
      "q": "0"
    },
    {
      "id": 96,
      "q": ">500"
    }
  ],
  "reserve": 1,
  "quantityinbox": 20,
  "isNewWare": false,
  "isAllreadyExistWare": false,
  "isNotmakingWare": false,
  "isTransitU": false,
  "isTransitW": false,
  "isTransitY": false,
  "actions": "",
},
...
]

```

### 1.5 Поля відповіді

Найменування	Опис
ercsprice datetime	Дата вивантаження
vendor name	Вендор
category	Категорія
subcategory	Підкатегорія
gname	Найменування
code	SKU
sprice	Ціна
ddp	Ознака товару ДДП - не ДДП
vat	ПДВ
pic	Посилання на зображення позиції
ean	EAN
bar	BAR
upc	UPC

declar	Декларація
tnvd	Код ТНВЕД, для послуги виводиться значення "УСЛ", для віртуального товару -значення "ВТ", для не імпортного товару (внутрішні закупки) - значення "ВЗ"
cert	Сертифікат
warr	Гарантія (гарантія число 1, 12, 24, ... )
warrunit	Тип гарантії (1-день, 2-місяць, 3-рік)
width	Габарити (ширина, мм)
height	Габарити (висота, мм)
depth	Габарити (глибина, мм)
weight	Вага (кг)
volume	Об'єм (куб. м)
RRP_UAH	Ціна «On-Line RRP UAH»
monitor	Монітор
isMarked	Ознака маркування продукції для ПЗ (1 - марковане, необхідний голографічний захисний елемент)
isaction	Акція
serialControl	Необхідність сканування SN при відвантаженні (true – потребує вихідного сканування, false – не потребує)
country	Країна
whs	Кількість товару по відповідним складам
<pre>&lt;whs&gt;&lt;wh id="1" q="0" /&gt;</pre> <p>(для XML формату)</p> <pre>"whs": [{"id": 1, "q": "0" }]</pre> <p>(для JSON формату)</p>	<p>Кількість товару на 1-му складі:</p> <p>до 50 включно, відображається точна кількість товару</p> <p>&gt;50 – від 51 до 100</p> <p>&gt;100 – від 101 до 500</p> <p>&gt;500 – від 501 до 1000</p> <p>&gt;1000 – від 1001 до 5000</p> <p>&gt;5000 – від 5001 до 10000</p> <p>&gt;10000 – від 10001</p> <p>Значення в полі дорівнює різниці кількості товару (на 1 складі – Резерв - Товар в збірці).</p> <p><b>Приклад виводу:</b> <code>wh id="1" q="43"</code>, тобто 43 шт. на 1-му складі</p>
<pre>&lt;whs&gt;&lt;wh id="44" q="2" /&gt;</pre> <p>(для XML формату)</p>	Кількість товару на 44-му складі(градація аналогічна градації кількості товару на 1-му складі)

<p>формату)</p> <pre>"whs": [{"id": 44, "q": "2" }]</pre> <p>(для JSON формату)</p>	Значення в полі дорівнює різниці кількості товару (на 44 складі – Резерв - Товар в збірці)
<pre>&lt;whs&gt;&lt;wh id="55" q="0" /&gt;</pre> <p>(для XML формату)</p> <pre>"whs": [{"id": 55, "q": "0" }]</pre> <p>(для JSON формату)</p>	<p>Кількість товару на 55-му складі(градація аналогічна градації кількості товару на 1-му складі)</p> <p>Значення в полі дорівнює різниці кількості товару (на 55 складі – Резерв - Товар в збірці)</p>
<pre>&lt;whs&gt;&lt;wh id="5" q="0" /&gt;</pre> <p>(для XML формату)</p> <pre>"whs": [{"id": 5, "q": "0" }]</pre> <p>(для JSON формату)</p>	<p>Кількість товару на 5-му (софт) складі(градація аналогічна градації кількості товару на 1-му складі)</p> <p>Значення в полі дорівнює різниці кількості товару (на 5 складі – Резерв - Товар в збірці)</p>
<pre>&lt;whs&gt;&lt;wh id="95" q="0" /&gt;</pre> <p>(для XML формату)</p> <pre>"whs": [{"id": 95, "q": "0" }]</pre> <p>(для JSON формату)</p>	<p>Кількість товару надодатковому95-му складі(градація аналогічна градації кількості товару на 1-мускладі)</p> <p>Значення в полі дорівнює різниці кількості товару (на 95 складі – Резерв - Товар в збірці)</p>
<pre>&lt;whs&gt;&lt;wh id="96" q="0" /&gt;</pre> <p>(для XML формату)</p> <pre>"whs": [{"id": 96, "q": "0" }]</pre> <p>(для JSON формату)</p>	<p>Кількість товару надодатковому96-му складі(градація аналогічна градації кількості товару на 1-мускладі)</p> <p>Значення в полі дорівнює різниці кількості товару (на 96складі – Резерв - Товар в збірці)</p>
reserve	Товар є в резерві (1-так, 0 -ні)
quantityinbox	Кількість в упаковці
isNewWare	Новинка. Дана ознака встановлюється, якщо був прихід товару протягом останніх 21 дня
isAlreadyExistWare	Знову доступний. Дана ознака встановлюєтьсядля товару в прайсі та остання дата продажу більше ніж 60 днів

isNotmakingWare	Товар більше не виробляється
isTransitU	Розвантажуються (true, false)
isTransitW	Завантажений в машину та їде (true, false)
isTransitY	Прибув в Київ, заїхав на митницю(true, false)
action	Вивантажуються всі коди активних акцій для роздрібною торгівлі і дилерів

## 1.6 Приклади виклику

Для PHP (Online запуск php скриптів:<https://ideone.com/>)

```
<?php

$email = 'user@email.com';
$password = '****';
$infoType = 6;

$data = array("email" => $email, "pass" => $password, "infoType" => $infoType);
$data_string = json_encode($data);

$ch = curl_init('https://connect.erc.ua/connectservice/api/specprice/DoExport');
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "POST");
    curl_setopt($ch, CURLOPT_POSTFIELDS, $data_string);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
    'Content-Type: application/json',
    'Content-Length: ' . strlen($data_string)
));
curl_setopt($ch, CURLOPT_TIMEOUT, 50);
curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 50);

//execute post
$result = curl_exec($ch);

//close connection
curl_close($ch);

echo $result;

?>
```

Для C#

```
public class SpecPriceClient
{
    public string Email { get; set; }
    public string Pass { get; set; }
    public int InfoType { get; set; }
}

var user = new SpecPriceClient() { Email = "user@email.com", Pass = "****",
```

```
InfoType = 6 };
using (var httpClient = new HttpClient())
{
    var requestUri = "https://connect.erc.ua/connectservice/api/specprice/DoExport";
    var response = httpClient.PostAsJsonAsync(requestUri, user).Result;
    var xml = response.Content.ReadAsStringAsync().Result;

    return response;
}
```

### Для AJAX

```
var json = JSON.stringify({ "email": 'user@email.com', "pass": '****', "infotype": 6 });

$.ajax({
    url: 'https://connect.erc.ua/connectservice/api/specprice/DoExport',
    type: 'post',
    contentType: 'application/json', // type of data
    data: json,
    dataType: 'xml',
    success: function (data) {
        var $xml = $(data);
        $($xml).each(function () {
            var text = $(this).find("vendor[name]").text();
            alert(text);
        });
    },

    error: function(errorObj) {
        alert(errorObj.responseText);
    }
}
```

## 2. Курси валют

### 2.1 Опис

InfoType = 7. Цей метод дозволяє отримувати курси валют: курс в Києві, курс виписування, курс зарахування.

*Примітка:* Якщо значення курсу дорівнює 1000000 - це означає, що курс на даний момент ще не визначений.

### 2.2 Адреса сервіса

<https://connect.erc.ua/connectservice/api/specprice/DoExport>

### 2.3 Вхідні параметри

Найменування	Тип	Опис
Email	String	Email користувача
Pass	String	Пароль користувача
InfoType	Int	Тип експорту: 7
IsJson	Boolean	Не обов'язковий параметр (опціонально - при відсутності результат буде вивантажено в XML). Можливі значення: 0, 1, true, false
CurrencyRateDate	Date	Дата, на яку буде повернуто курс, формат дати: "04.04.2021" або "04/04/2021"
Lang	String	Мова, на якій буде вивантажено результат (за замовчуванням - українська). Можливі значення: ru, ua, ru-ru, uk-ua (без урахування регістру)

### 2.4 Приклад даних, що повертаються

Результат доступний в XML і в JSON форматі.

Для XML

```
<ercsprice datetime="Пт Бер 10 04:36:07 EET 2017">
<rates>
  <paperwork>27,2</paperwork>
  <cash>27,25</cash>
  <setoff>0</setoff>
</rates>
</ercsprice>
```

Для JSON

```
{
  "paperwork": 26.85,
```

```
"cash": 26.69,  
"setoff": 26.69  
}
```

## 2.5 Поля відповіді

Найменування	Опис
ercsprice datetime	Дата вивантаження
paperwork	Курс виписування
cash	Курс в Києві
setoff	Курс зарахування

## 2.6 Приклади виклику

Дивись розділ 1.6.

### 3. Активні акції

#### 3.1 Опис

InfoType = 8. Цей метод дозволяє отримувати інформацію про всі активні (діючі) акції.

#### 3.2 Адреса сервіса

<https://connect.erc.ua/connectservice/api/specprice/DoExport>

#### 3.3 Вхідні параметри

Найменування	Тип	Опис
Email	String	Email користувача
Pass	String	Пароль користувача
InfoType	Int	Тип експорту: 8
IsJson	Boolean	Не обов'язковий параметр (опціонально - при відсутності результат буде вивантажено в XML). Можливі значення: 0, 1, true, false
Lang	String	Мова, на якій буде вивантажено результат (за замовчуванням - українська). Можливі значення: ru, ua, ru-ru, uk-ua (без урахування регістру)

#### 3.4 Приклад даних, що повертаються

Результат доступний в XML і в JSON форматі.

Для XML

```
<?xml version="1.0" encoding="UTF-8"?>
<actions>
<items>
<item>
<actionId>21337</actionId>
<descr>&lt;p&gt;3&nbsp;15 червня&nbsp;до 15&nbsp;липня 2018р. до кожного
БФП &lt;strong&gt;Brother DCP-1602R&lt;/strong&gt; чи принтери &lt;/descr>
</item>
<item>
<actionId>21876</actionId>
<descr>&lt;p&gt;До кожного пристрою категорії &lt;strong&gt;HP S-Printing&lt;/strong&gt; ми
даруємо &lt;strong&gt;&lt;font color="#ff0000"&gt;смаколики!&lt;/font color="&lt;/descr>
</item>
</items>
</actions>
```



## ДляJSON

```
[
  {
    "actionId": 29686,
    "dateEnd": "2020-05-15T00:00:00",
    "descr": "При купівлі акційних моделей кондиціонерів ELECTROLUX* (відвантаження не нижче колонки F) - в подарунок тепловентилятор!\nВ акції приймають участь партери, які дотримуються РІЦ"
  },
  ...
]
```

### 3.5 Поля відповіді

Найменування	Опис
actionId	Номер акції
dateEnd	Дата закінчення акції
descr	Опис акції

### 3.6 Приклади виклику

Дивись розділ 1.6.

## 4. Пошкоджений товар

### 4.1 Опис

InfoType = 9. Цей метод дозволяє отримувати інформацію про пошкоджений товар з їх описом і посиланнями на фото ушкоджень (товар з 44 і 55 складів).

*Примітка:* На 44 складі зберігається товар з ушкодженнями (подряпини, невеликі вм'ятини і ін.), на 55 складі - товар з пошкодженою упаковкою.

### 4.2 Адреса сервіса

<https://connect.erc.ua/connectservice/api/specprice/DoExport>

### 4.3 Вхідні параметри

Найменування	Тип	Опис
Email	String	Email користувача
Pass	String	Пароль користувача
InfoType	Int	Тип експорту: 9
IsJson	Boolean	Не обов'язковий параметр (опціонально - при відсутності результат буде вивантажено в XML). Можливі значення: 0, 1, true, false
DiscountType	Int	Тип пошкодженого товару (опціонально). Можливі значення : 44 (ОГТ); 55 (ТПУ). <b>Якщо не передавати, то буде 44 (ОГТ).</b>
VendorId	Int	Код вендора (опціонально). Отримання списку кодів вендорів див. розділ 5
CategoryId	Int	Код категорії (опціонально). Отримання списку кодів категорій див. розділ 6
IsNew	Boolean	Вивантажувати тільки нові позиції (опціонально). Можливі значення: 0,1, true, false
OnlyFree	Int	Вивантажувати тільки доступні для ззамовлення позиції (опціонально)
IsAction	Boolean	Вивантажувати тільки позиції, що беруть участь в акціях акціях (опціонально). Можливі значення: 0,1,true,false
Lang	String	Мова, на якій буде вивантажено результат (за замовчуванням - українська). Можливі значення: ru, ua, ru-ru, uk-ua (без урахування регістру)

#### 4.4 Приклад даних, що повертаються

Результат доступний в XML і в JSON форматі.

Для XML

```
<goods>
<vendor>ВЕКО</vendor>
<category>Бытовая техника крупная</category>
<subcategory>Посудомоечные машины</subcategory>
<gname>Отдельно стоящая посудомоечная машина Веко DFN26220X - 60 см./12 компл./6
прогр/дисплей/A++/нерж ст</gname>
<code>DFN26220X</code>
<sprice>XXXXX</sprice>
<ddp>1</ddp>
<comment>Сильно повреждена упаковка (Вмятина на передней панели)</comment>
<serial>7619983942165084640901</serial>
<images>
<url>https://connect.erc.ua/files/sn/f5c5b2e1-5c1b-4a02-88b3-4b04107449f5.JPG</url>
<url>https://connect.erc.ua/files/sn/713a41aa-d1f5-43cd-bb68-b867aecdff95.JPG</url>
</images>
</goods>
```

ДляJSON

```
[
{
  "vendor": "Ariston",
  "category": "Климатическая техника",
  "subcategory": "Водонагреватели",
  "gname": "Водонагреватель Ariston ANDRIS LUX ECO 10U PL EU, 10 л, под мойкой, электр.
упр-ние, Италия",
  "code": "3100693",
  "sprice": "XXXX",
  "ddp": "1",
  "comment": "есть не большие царапины.",
  "serial": "310069319100001",
  "images": [
    {
      "url": "https://connect.erc.ua/files/sn/48ba61f7-73e2-4198-ae79-cd0a5781d472.JPG"
    },
    {
      "url": "https://connect.erc.ua/files/sn/f66b270c-c73e-463f-86c4-ccc32ba6aed6.JPG"
    },
    {
      "url": "https://connect.erc.ua/files/sn/1a55ec9c-6c98-4f01-b214-681d1e680385.JPG"
    }
  ]
}
]
```

#### 4.5 Поля відповіді

Найменування	Опис
vendor name	Вендор
category	Категорія
subcategory	Підкатегорія
gname	Найменування
code	SKU
sprice	Ціна
ddp	Ознака товару ДДП - не ДДП
comment	Опис пошкоджень
serial	Серійний номер товару
images	url-посилання на зображення пошкоджень

#### 4.6 Приклади виклику

Дивись розділ 1.6.

## 5. Список кодів вендорів

### 5.1 Опис

Цей метод дозволяє отримати коди всіх вендорів, для їх подальшого використання в запитах.

### 5.2 Адреса сервіса

<https://connect.erc.ua/connectservice/api/SpecPrice/GetVendors>

### 5.3 Вхідні параметри

Найменування	Тип	Опис
Email	String	Email користувача
Pass	String	Пароль користувача
Lang	String	Мова, на якій буде вивантажено результат (за замовчуванням - українська). Можливі значення: ru, ua, ru-ru, uk-ua (без урахування регістру)

### 5.4 Приклад даних, що повертаються

Результат доступний тільки JSON форматі.

```
{
  "IsError":false,
  "ErrorCode":0,
  "ResultMessages":"OK",
  "Data":
    [
      {
        "ID":"0000015","Name":"!Розпродаж"
      },
      {
        "ID":"175","Name":"1MORE"
      },
      {
        "ID":"052","Name":"2E"
      },
      ...
    ]
}
```

## 5.5 Поля відповіді

Найменування	Опис
ErrorCode	Код помилки (0-помилки немає, відмінний від 0 – помилка)
IsError	Булевий результат (true – помилка створення замовлення, false – заказ створено)
ResultMessages	Текст помилки, або "ОК", при вдалому виконанні запиту
Data	Список вендорів
<a href="#">Атрибути вендора</a>	
ID	Код вендора
Name	Назва вендора

## 5.6 Приклади виклику

Дивись розділ 1.6.

## 6. Код категорії

### 6.1 Опис

Цей метод дозволяє отримати коди всіх категорій, для їх подальшого використання в запитах.

### 6.2 Адреса сервіса

<https://connect.erc.ua/connectservice/api/SpecPrice/GetCategories>

### 6.3 Вхідні параметри

Найменування	Тип	Опис
Email	String	Email користувача
Pass	String	Пароль користувача
CategoryId	Int	Опціональний параметр «батьківська» категорія
Lang	String	Мова, на якій буде вивантажено результат (за замовчуванням - українська). Можливі значення: ru, ua, ru-ru, uk-ua (без урахування регістру)

### 6.4 Приклад даних, що повертаються

Результат доступний тільки в JSON форматі.

```

{
  "IsError": false,
  "ErrorCode": 0,
  "ResultMessages": "OK",
  "Data": [
    {
      "ID": "0058",
      "Name": "ІТ Аудіо"
    },
    {
      "ID": "0058001",
      "Name": "Колонки для ПК"
    },
    {
      "ID": "0058002",
      "Name": "Гарнітури для ПК"
    }
  ]
}

```

### 6.5 Поля відповіді

Найменування	Опис
ErrorCode	Код помилки (0-помилки немає, відмінний від 0 – помилка)

IsError	Булевий результат (true – помилка створення замовлення, false – заказ створено)
ResultMessages	Текст помилки, або "ОК", при вдалому виконанні запиту
Data	Список категорій
<a href="#">Атрибути категорії</a>	
ID	Код категорії
Name	Назва категорії

## 6.6 Приклади виклику

Дивись розділ 1.6.



## 7. Створення замовлень в режимі «Самовивіз»

Створення замовлень в режимі «Самовивіз» дозволяє самостійно формувати замовлення (авто-замовлення) із зазначеними вантажоодержувачем і платником, а також відправляти його в збірку для самовивозу.

Для цього необхідно виконати наступні дії:

- a) отримати список вантажоодержувачей партнера з ПН;
- b) отримати список платників партнера;
- c) отримати список договорів (при необхідності);
- d) створити замовлення в режимі «самовивіз».

### 7.1 Отримання списку вантажоодержувачей з ПН

#### 7.1.1 Опис

Цей метод дозволяє отримати список вантажоодержувачів, тобто людей, які внесені в картку клієнта як одержувачі вантажу.

#### 7.1.2 Адреса сервіса

<https://connect.erc.ua/connectservice/api/Documents/GetPersons/>

#### 7.1.3 Вхідні параметри

Найменування	Тип	Опис
Email	String	Email користувача
Pass	String	Пароль користувача

#### 7.1.4 Приклад даних, що повертаються

Для JSON

```
"Data": {
  [
    {
      "FirstName": "Иван",
      "LastName": "Иванов",
      "MiddleName": "Иванович",
      "INN": 1111111111
    },
    {
      "FirstName": "Петров",
      "LastName": "Петр",
      "MiddleName": "Петрович",
      "INN": 2222222222
    }
  ]
}
```

### 7.1.5 Поля відповіді

Найменування	Опис
ErrorCode	Код помилки (0-помилки немає, відмінний від 0 – помилка)
IsError	Булевий результат (true – помилка створення замовлення, false – заказ створено)
ResultMessages	Текст помилки, або "ОК", при вдалому виконанні запиту
Data	Список вантажоотримувачей, що включає в себе ПІБ та ПІН вантажоотримувача
<b>Атрибутивантажоотримувача</b>	
FirstName	Ім'я вантажоотримувача
LastName	Прізвищевантажоотримувача
MiddleName	По-батьковівантажоотримувача
INN	ПІН вантажоотримувача

## 7.2 Отримання списку платників/вантажоотримувачів юр. осіб

### 7.2.1 Опис

Цей метод дозволяє отримати список компаній, які зазначені в картці клієнта як його платники.

### 7.2.2 Адреса сервіса

<https://connect.erc.ua/connectservice/api/Documents/GetPayers/>

### 7.2.3 Вхідні параметри

Найменування	Тип	Опис
Email	String	Email користувача
Pass	String	Пароль користувача

### 7.2.4 Приклад даних, що повертаються

Для JSON

```

{
  "Data": {
    [
      {
        "ContractorID": 1111,
        "ContractorName": "ФИРМА",
        "ContractorEDRPOU": "45121454778"
      },
      {
        "ContractorID": 2222,
        "ContractorName": "ФИРМА2",
      }
    ]
  }
}

```

```

    "ContractorEDRPOU": "86523498756"
  ]
}

```

### 7.2.5 Поля відповіді

Найменування	Опис
ErrorCode	Код помилки (0-помилки немає, відмінний від 0 – помилка)
IsError	Булевий результат (true – помилка створення замовлення, false – заказ створено)
ResultMessages	Текст помилки, або "ОК", при вдалому виконанні запиту
Data	Список платників/вантажоотримувачів юр. осіб
<b>Атрибути платника/вантажоотримувача юр. осіб</b>	
ContractorID	ID платника/вантажоотримувача юр. осіб
ContractorName	Назва платника/вантажоотримувача юр. осіб
ContractorEDRPOU	ЄДРПОУ платника/вантажоотримувача юр. осіб

## 7.3 Отримання списку договорів

### 7.3.1 Опис

Цей метод дозволяє отримати список договорів компанії. Отримані результати використовуються при створенні замовлень (метод **CreateOrder**).

### 7.3.2 Адреса сервіса

<https://connect.erc.ua/connectservice/api/documents/GetContracts>

### 7.3.3 Вхідні параметри

Найменування	Тип	Опис
Email	String	Email користувача
Pass	String	Пароль користувача
PayerId	int	Опціонально - платник, будуть обрані договору по платнику, отримання списку платників описано в п. 7.2.
ContractId	int	Будуть обрані дані за певним договором (опціонально).
isSoftware	Boolean	Будуть обрані договори по «програмному забезпеченню» (опціонально).

### 7.3.4 Приклад даних, що повертаються

## Для JSON

```

    "Data": {
      [
        «ParentContractId»: 111111,
        «ContractId»: 12547,
        «AgreemTypeName»: Право на распространение,
        «IsSoftware»: true,
        «AgreementNo»: 111111,
        «DateFrom»: 01.05.2020,
        «DateTo»: 31.12.2020,
        «ManagerName»: Иванченко Иван,
        «BankName»: КБ «Приват»,
        «MFO»: 380222,
        «BankAccount»: 260042586248,
        «ConsignorId»: 222222,
        «Consignor»: ФИРМА
      ],
      .....
    }
  
```

## 7.3.5 Поля відповіді

Найменування	Опис
ParentContractId	Номер батьківського договору (основного по додатковому) можливий варіант null якщо немає батьківського договору.
ContractId	Id контракту
AgreemTypeName	Тип договору
IsSoftware	Договір по ПЗ
AgreementNo	Номер (найменування) договору
DateFrom	Дата початку
DateTo	Дата закінчення
ManagerName	Менеджер (який уклав або створив договір)
BankName	Банк по договору
MFO	МФО банку
BankAccount	№ рахунка
ConsignorId	Idвантажовідправника
Consignor	Найменування вантажовідправника

Примітка. Якщо основний договір закінчився - є актуальний доп. договір (при цьому ID основного договору буде в полі ParentContractId), щоб

отримати інформацію про основний договір - необхідно передати його ID в параметрах.

## 7.4 Створення замовлення в режимі «самовивіз»

### 7.4.1 Опис

Цей метод дозволяє самостійно створити замовлення (авто-замовлення) і відправити його в збірку в режимі «Самовивіз».

### 7.4.2 Адреса сервіса

<https://connect.erc.ua/connectservice/api/Documents/CreateOrder/>

### 7.4.3 Вхідні параметри

Найменування	Тип	Опис
Email	String	Email користувача
Pass	String	Пароль користувача
INN	Int	ПІН вантажоотримувача клієнта (вантажоотримувач повинен бути активним на дату замовлення)
ContractId	Int	Id контракту (опціонально)
Version	Int	Версійність результату створення замовлення (опціональний, по-замовчуванню "1.0")
DocExtNum	String	Номер замовлення облікової системи дилера (опціонально)
WarehouseID	Int	Обов'язковий. Створення замовлення у розрізі складу. Припустимі значення: 1, 5, 95, 96
Specs		Масив специфікації код-позиції, кількість
Lang	String	Мова, на якій буде вивантажено результат (за замовчуванням - українська). Можливі значення: ru, ua, ru-ru, uk-ua (без урахування регістру)
<b>IsPaperWork</b>	Boolean	<b>Необхідне оформлення бухгалтерських документів (повідомлення Ведучого дилера-менеджера в ERC)*.</b> <b>Для використання цього параметру потрібно мати розширений доступ.</b> <b>Можливі значення: 1, 0</b>
EDRPOU	Int	ЄДРПОУ платника (опціонально)

PayerID	Int	Код платника (опціонально)
---------	-----	----------------------------

\* **Важливо!** При необхідності оформлення бухгалтерських документів (параметр **IsPaperWork**) обов'язково необхідно вказати хоча б один з параметрів: EDRPOU або PayerID.

#### 7.4.4 Приклад даних, що повертаються

Для JSON

- При параметрі Version "1.0"

```
[
  {
    "DocExtNum": "111111",
    "DocId": 22222222,
    "DocNum": "1/33333333",
    "OrderNum": "4444444444",
    "IsDDP": true,
    "PaperworkCurrencyRate": 28,5
    "Specs": {
      "5555555": 1
    },
    "IsError": false,
    "ErrorCode": 0,
    "ResultMessages": "Созданный \Заказнаотгрузку\"(1/33333333) В2В! | Документ
утвержден"
  }
]
```

- При параметрі Version "1.1" розділ Specs буде мати наступний вигляд:

```
"Specs":
[
  {"w": " Ware1", "q": 1, "p": 1111, "s": 1111},
  {"w": " Ware2", "q": 2, "p": 22222, "s": 44444},
]
```

В результаті додається ціна (p) та сума (s)

#### 7.4.5 Поля відповіді

Найменування	Опис
DocNum	Номер створеного замовлення (може бути пустим рядком, якщо не створено замовлення)
OrderNum	Номер замовлення В2В
IsDDP	Валюта замовлення. Можливі значення: true – гривневе замовлення, false – долларове замовлення
PaperworkCurrencyRate	Курс виписки
ErrorCode	Код помилки (0-помилки немає, відмінний від 0 – помилка)
IsError	Булевий результат (true – помилка створення)

	замовлення, false – заказ створено)
ResultMessages	Текст помилки, або текст про вдале створення замовлення
Specs	Специфікація створеного документу або документу «на погодження»

#### 7.4.6 Приклади виклику режиму «самовивіз»

Для PHP (Online запуск php скриптів:<https://ideone.com/>)

```
<?php
$email = 'user@email.com';
$pass = '****';
$inn = 1111111111;
$lang='ru';
$isPaperWork=1;
$edr Pou=111111111;

$specArray =
array(
    array(
        "w" =>"1M301-SPACEGRAY",
        "q" => 1
    ),
    array(
        "w" =>"E1003-SPACEGRAY",
        "q" => 4
    )
);
$data = array("Email" => $email, "Pass" => $pass, "INN" => $inn, "Lang" => $lang,
"IsPaperWork" => $isPaperWork, "EDRPOU" => $edr Pou, "Specs" => $specArray);
$data_string = json_encode($data);

$ch = curl_init('https://connect.erc.ua/connectservice/api/Documents/CreateOrder');
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "POST");
curl_setopt($ch, CURLOPT_POSTFIELDS, $data_string);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
    'Content-Type: application/json',
    'Content-Length: ' . strlen($data_string)
));
//execute post
$result = curl_exec($ch);
//close connection
curl_close($ch);
echo $result;
?>
```

## Для C#

```
public async Task<JObject> CallCreateOrder()
{
    var specList = new List<object>()
    {
        new { w = "1M301-SPACEGRAY", q = 1 },
        new { w = "E1003-SPACEGRAY", q = 1 },
        new { w = "EO320-GOLD", q = 4 }
    };
    var specArrayJson = JArray.FromObject(specList);

    var jsonParam = new JObject
    {
        ["Email"] = "user@email.com",
        ["Pass"] = "****",
        ["INN"] = 1111111111,
        ["Lang"] = "ru",
        ["IsPaperWork"] = 1,
        ["EDRPOU"] = 11111111
    };
    jsonParam.Add(new JProperty("Specs", specArrayJson));

    var credentials = CredentialCache.DefaultCredentials;
    var handler = new HttpClientHandler {Credentials = credentials};

    using (var client = new HttpClient(handler))
    {
        client.BaseAddress = new Uri("https://connect.erc.ua/connectservice/");
        client.DefaultRequestHeaders.Accept.Clear();
        client.DefaultRequestHeaders.Accept.Add(new
        MediaTypeWithQualityHeaderValue("application/json"));

        var response = client.PostAsJsonAsync("api/Documents/CreateOrder", jsonParam).Result;

        if (response.IsSuccessStatusCode)
        {
            var jsonString = await response.Content.ReadAsStringAsync();
            try
            {
                var retJson = JObject.Parse(jsonString);
                return retJson;
            }
            catch (Exception ex)
            {
                return null;
            }
        }
    }
}
```



```
returnnull;
}
```

Для JavaScript/AJAX

```
var specArray = [
  {'w': '1M301-SPACEGRAY', 'q': 1},
  {'w': 'E1003-SPACEGRAY', 'q': 1},
  {'w': 'EO320-GOLD', 'q': 4}
];

var postData = {
  Email: 'user@email.com',
  Pass: '****',
  Inn: 1111111111,
  Lang: 'ru',
  IsPaperWork: 1,
  EDRPOU: 11111111,
  Specs: specArray
};

$.ajax({
  url: 'https://connect.erc.ua/connectservice/api/Documents/CreateOrder/',
  data: postData,
  type: 'POST',
  contentType: 'application/json',
  dataType: 'json',

  success: function(jdata) {
    if (jdata.hasOwnProperty('d')) {
      jdata = jdata.d;
    }
    var errorCode = jdata.ErrorCode;
    var mess = jdata.ResultMessages;
  },
  loadError: function(xhr, st, err) {
    alert(xhr.responseText);
  }
});
```

## 8. Створення замовлень в режимі «Доставка»

Створення замовлень в режимі «Доставка» дозволяє самостійно формувати замовлення (авто-замовлення) із зазначеними вантажоодержувачем, платником, рейсом і методом доставки, а також відправляти його в збірку для подальшої доставки за вказаною адресою

Для цього необхідно виконати наступні дії:

- a) отримати список адрес партнера;
- b) отримати список вантажоодержувачей партнера з ІПН;
- c) отримати список активних рейсів для адреси (з датами закінчення реєстрації і відвантаження);
- d) отримати список платників партнера;
- e) отримати список способів оплати доставки (Вантажоодержувач «Двері - Склад», Вантажоодержувач «Двері-Двері» ...);
- f) створити замовлення (авто-замовлення) для режиму «Доставка».

## 8.1 Отримання списку адрес партнера з прив'язаним контактом

### 8.1.1 Опис

Цей метод дозволяє отримати список адрес партнерів.

### 8.1.2 Адреса сервіса

<https://connect.erc.ua/connectservice/api/Documents/GetAddresses/>

### 8.1.3 Вхідні параметри

Найменування	Тип	Опис
Email	String	Email користувача
Pass	String	Пароль користувача

### 8.1.4 Приклад даних, що повертаються

Для JSON

```

    "Data": {
      [
        "AddressID": 1,
        "AddressName": " м. Чернівці, вул. ***"
      ],
      [
        "AddressID": 2,
        "AddressName": "м. Київ, вул. 122, буд.2"
      ]
    ]
  }

```

### 8.1.5 Поля відповіді

Найменування	Опис
ErrorCode	Код помилки (0-помилки немає, відмінний від 0 – помилка)
IsError	Булевий результат (true – помилка створення замовлення, false – заказ створено)
ResultMessages	Текст помилки, або "ОК", при вдалому виконанні запиту

Data	Список адрес
<a href="#">Атрибути адреси</a>	
AddressID	Код адреси доставки
AddressName	Адреса складу партнера

## 8.2 Отримання списку вантажоодержувачей партнера з ПН

### 8.2.1 Опис

Цей метод дозволяє отримати список вантажоодержувачів, тобто людей, які внесені в картку клієнта як одержувачі вантажу.

### 8.2.2 Адреса сервіса

<https://connect.erc.ua/connectservice/api/Documents/GetPersons>

### 8.2.3 Вхідні параметри

Найменування	Тип	Опис
Email	String	Email користувача
Pass	String	Пароль користувача

### 8.2.4 Приклад даних, що повертаються

Для JSON

```
"Data": [
  {
    "FirstName": "Андрій",
    "LastName": "Андрієнко",
    "MiddleName": "Андрійович",
    "INN": "2369754102"
  },
  ...
```

### 8.2.5 Поля відповіді

Найменування	Опис
ErrorCode	Код помилки (0-помилки немає, відмінний від 0 – помилка)
IsError	Булевий результат (true – помилка створення замовлення, false – заказ створено)
ResultMessages	Текст помилки, або "ОК", при вдалому виконанні запиту
Data	Список вантажоотримувачів
<a href="#">Атрибутивантажоотримувача</a>	
FirstName	Им'я вантажоотримувача
LastName	Призвищевантажоотримувача
MiddleName	По-батьковівантажоотримувача

INN	ПІН вантажоотримувача
-----	-----------------------

### 8.3 Отримання списку рейсів

#### 8.3.1 Опис

Цей метод дозволяє отримати список актуальних рейсів доставки з датами закінчення реєстрації і відправки.

#### 8.3.2 Адреса сервіса

<https://connect.erc.ua/connectservice/api/Documents/GetDeliveryRotes/>

#### 8.3.3 Вхідні параметри

Найменування	Тип	Опис
Email	String	Email користувача
Pass	String	Пароль користувача
AddressID	Int	Код адреси доставки
WarehouseID	Int	Обов'язковий. Відображення рейсів у розрізі складу. Припустимі значення: 1, 5, 95, 96.

#### 8.3.4 Приклад даних, що повертаються

Для JSON

```

"Data": {
  [
    "DeliveryTimeID": 1,
    "DeliveryTimeName": "",
    "RegEndDate": "1",
    "DepartureDate": ""
  ],
  [
    "DeliveryTimeID": 1,
    "DeliveryTimeName": "",
    "RegEndDate": "1",
    "DepartureDate": ""
  ],
  [
    "DeliveryTimeID": 1,
    "DeliveryTimeName": "",
    "RegEndDate": "1",
    "DepartureDate": ""
  ]
]
}

```

#### 8.3.5 Поля відповіді

Найменування	Опис
ErrorCode	Код помилки (0-помилки немає, відмінний від 0 –

	помилка)
IsError	Булевий результат (true – помилка створення замовлення, false – заказ створено)
ResultMessages	Текст помилки, або "ОК", при вдалому виконанні запиту
Data	Список рейсів доставки
<b>Атрибути рейса доставки</b>	
DeliveryTimeID	Код рейса доставки
DeliveryTimeName	Найменування рейса
RegEndDate	Дата/час закінчення реєстрації на рейс
DepartureDate	Дата/час відправки

## 8.4 Отримання списку платників/вантажоотримувачів юр. осіб

### 8.4.1 Опис

Цей метод дозволяє отримати список компаній, які зазначені в картці клієнта як його платники.

### 8.4.2 Адреса сервіса

<https://connect.erc.ua/connectservice/api/Documents/GetPayers/>

### 8.4.3 Вхідні параметри

Найменування	Тип	Опис
Email	String	Email користувача
Pass	String	Пароль користувача

### 8.4.4 Приклад даних, що повертаються

Для JSON

```

{
  "Data": {
    [
      {
        "ContractorID": 1111,
        "ContractorName": "ФІРМА",
        "ContractorEDRPOU": "45121454778"
      },
      {
        "ContractorID": 2222,
        "ContractorName": "ФІРМА2",
        "ContractorEDRPOU": "89657234514"
      }
    ]
  }
}

```

### 8.4.5 Поля відповіді

Найменування	Опис
ErrorCode	Код помилки (0-помилки немає, відмінний від 0 –

	помилка)
IsError	Булевий результат (true – помилка створення замовлення, false – заказ створено)
ResultMessages	Текст помилки, або "ОК", при вдалому виконанні запиту
Data	Список платників/вантажоотримувачів юр. осіб
<b>Атрибутиплатника/вантажоотримувача юр. осіб</b>	
ContractorID	ID платника/вантажоотримувача юр. осіб
ContractorName	Назваплатника/вантажоотримувача юр. осіб
ContractorEDRPOU	ЄДРПОУ платника/вантажоотримувача юр. осіб

## 8.5 Отримання списку способів оплати доставки

### 8.5.1 Опис

Цей метод дозволяє отримати список існуючих оплат доставки.

### 8.5.2 Адреса сервіса

<https://connect.erc.ua/connectservice/api/Documents/GetDeliveryPaymentTypes/>

### 8.5.3 Вхідні параметри

Найменування	Тип	Опис
Email	String	Email користувача
Pass	String	Пароль користувача

### 8.5.4 Приклад даних, що повертаються

Для JSON

```

'Data': {
  [
    "DeliveryPaymentTypeID": 1,
    "DeliveryPaymentTypeName": "Грузополучатель «Двери – Склад»",
  ],
  [
    "DeliveryPaymentTypeID": 2,
    "DeliveryPaymentTypeName": "Отпр. «Двери – Двери»",
  ]
}

```

### 8.5.5 Поля відповіді

Найменування	Опис
ErrorCode	Код помилки (0-помилки немає, відмінний від 0 – помилка)
IsError	Булевий результат (true – помилка створення

	замовлення, false – заказ створено)
ResultMessages	Текст помилки, або "ОК", при вдалому виконанні запиту
Data	Список способів оплати
<b>Атрибути способів оплати</b>	
DeliveryPaymentTypeID	ID способу оплати
DeliveryPaymentTypeName	Назва способу оплати

## 8.6 Створення замовлення в режимі «Доставка»

### 8.6.1 Опис

Цей метод дозволяє створювати авто-замовлення і відправляти його в збірку в режимі «Доставка».

### 8.6.2 Адреса сервіса

<https://connect.erc.ua/connectservice/api/Documents/CreateOrder/>

### 8.6.3 Вхідні параметри

Найменування	Тип	Опис
Email	String	Email користувача
Pass	String	Пароль користувача
AddressID	Int	Код адреси (див. <u>Отримання списку адрес</u> п. 8.1)
DeliveryTimeID	Int	Код рейса доставки (див. <u>Отримання списку рейсів</u> п. 8.3)
INN	Int	ППН вантажоотримувача клієнта (вантажоотримувач повинен бути активним на дату замовлення), обов'язкове
ContractId	Int	Id контракту (опціонально) (див. <u>Отримання списку договорів</u> п. 7.3)
Version	Int	Версійність результату створення замовлення (опціональний, по-замовчуванню "1.0")
DocExtNum	String	Номер замовлення облікової системи дилера (опціонально)
WarehouseID	Int	Обов'язковий. Створення замовлення у розрізі складу. Припустимі значення: 1, 5, 95, 96

Specs		Масив специфікації код-позиції, кількість
Lang	String	Мова, на якій буде вивантажено результат (за замовчуванням - українська). Можливі значення: ru, ua, ru-ru, uk-ua (без урахування регістру)
<b>IsPaperWork</b>	Boolean	<b>Необхідно оформлення бухгалтерських документів (повідомлення Ведучого дилер- менеджера в ERC)*.</b> <b>Для використання цього параметру потрібно мати розширений доступ.</b> <b>Можливі значення: 1, 0</b>
EDRPOU	Int	ЄДРПОУ платника (опціонально)
PayerID	Int	Код платника (опціонально)
ConsigneeID	Int	ID отримувача - юридичної особи (опціонально, для рейсів Нової пошти)
ConsigneeEDRPOU	Int	ЄДРПОУ отримувача - юридичної особи (опціонально, для рейсів Нової пошти)
DeliveryPaymentTypeID	Int	Код способу оплати доставки
Comment	String	Коментар по замовленню (опціонально)

\* **Важливо!** При необхідності оформлення бухгалтерських документів (параметр **IsPaperWork**) **обов'язково** необхідно вказати **хоча б один з параметрів**: EDRPOU або PayerID.

#### 8.6.4 Приклад даних, що повертаються

Для JSON

- При параметрі Version"1.0"

```
[
{
  "DocId": 111111,
  "DocNum": "1/33333",
  "OrderNum": "22222",
  "IsDDP": true,
  "PaperworkCurrencyRate": 28,5
  "Specs": {
    "Ware1": 3
  },
  "IsError": true,
  "ErrorCode": 100,
  "ResultMessages": "Погодження Создан новый \"Заказ на отгрузку\"(1/33333) B2B! | Документутвержден"
},
]
```



```
{
  "DocId": 777777,
  "DocNum": "1/33334",
  "OrderNum": "22223",
  "IsDDP": true,
  "PaperworkCurrencyRate": 28,5
  "Specs": {
    "Ware2": 7,
    "Ware3": 1,
    "Ware4": 4
  },
  "IsError": false,
  "ErrorCode": 0,
  "ResultMessages": "Созданновый \"Заказнаотгрузку\"(1/33334) B2B! | Документ утвержден"
}
```

- При параметрі Version "1.1" розділ Specs буде мати наступний вигляд:

```
"Specs":
[
  {"w": " Ware1", "q": 1, "p": 1111, "s": 1111},
  {"w": " Ware2", "q": 2, "p": 22222, "s": 44444},
]
```

В результаті додається ціна (p) та сума (s).

### 8.6.5 Поля відповіді

Найменування	Опис
DocNum	Номер створеного замовлення (може бути пустим рядком, якщо замовлення не створено)
OrderNum	Номер замовлення B2B
IsDDP	Валюта замовлення. Можливі значення: true – гривневе замовлення, false – долларове замовлення
PaperworkCurrencyRate	Курс виписки
ErrorCode	Код помилки (0-помилки немає, відмінний від 0 – помилка)
IsError	Булевий результат (true – помилка створення замовлення, false – заказ створено)
ResultMessages	Текст помилки, або текст про вдале створення замовлення
Specs	Специфікація створеного документа або документа «на погодження»

### 8.6.6 Приклади виклику режиму «доставка»

Для PHP (Online запуск php скриптів: <https://ideone.com/>)

```
<?php
```

```

$Email = 'user@email.com';
$Pass = '****';
$AddressID = 1;
$DeliveryTimeID=1;
$Lang='ru';
$IsPaperWork=1;
$EDRPOU=11111111;

$specArray =
array(
    array(
        "w" =>"1M301-SPACEGRAY",
        "q" => 1
    ),
    array(
        "w" =>"E1003-SPACEGRAY",
        "q" => 4
    )
);

$data = array("Email" => $Email, "Pass" => $Pass, "INN" => $INN, "Lang" => $Lang,
"IsPaperWork" => $IsPaperWork, "EDRPOU" => $EDRPOU, "Specs" => $specArray);
$data_string = json_encode($data);

$ch = curl_init('https://connect.erc.ua/connectservice/api/Documents/CreateOrder');
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "POST");
curl_setopt($ch, CURLOPT_POSTFIELDS, $data_string);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
    'Content-Type: application/json',
    'Content-Length: ' . strlen($data_string)
));
//execute post
$result = curl_exec($ch);
//close connection
curl_close($ch);

echo $result;
?>

```

Для C#

```

public async Task<JObject> CallCreateOrder()
{
    var specList = new List<object>()
    {
        new { w = "1M301-SPACEGRAY", q = 1 },
        new { w = "E1003-SPACEGRAY", q = 1 },
    }
}

```

```
        new { w = "EO320-GOLD", q = 4 }
    };

    var specArrayJson = JArray.FromObject(specList);

    var jsonParam = new JObject
    {
        ["Email"] = "user@email.com",
        ["Pass"] = "****",
        ["INN"] = 111111111,
        ["Lang"] = "ru",
        ["IsPaperWork"] = 1,
        ["EDRPOU"] = 11111111
    };
    jsonParam.Add(new JProperty("Specs", specArrayJson));

    var credentials = CredentialCache.DefaultCredentials;
    var handler = new HttpClientHandler {Credentials = credentials};

    using (var client = new HttpClient(handler))
    {
        client.BaseAddress = new Uri("https://connect.erc.ua/connectservice/");
        client.DefaultRequestHeaders.Accept.Clear();
        client.DefaultRequestHeaders.Accept.Add(new
        MediaTypeWithQualityHeaderValue("application/json"));

        var response = client.PostAsJsonAsync("api/Documents/CreateOrder", jsonParam).Result;

        if (response.IsSuccessStatusCode)
        {
            {
                var jsonString = await response.Content.ReadAsStringAsync();
                try
                {
                    {
                        var retJson = JObject.Parse(jsonString);
                        return retJson;
                    }
                }
                catch (Exception ex)
                {
                    {
                        return null;
                    }
                }
            }
        }

        return null;
    }
}
```

Для JavaScript/AJAX

```

var specArray = [
    {'w': '1M301-SPACEGRAY', 'q': 1},
    {'w': 'E1003-SPACEGRAY', 'q': 1},
    {'w': 'EO320-GOLD', 'q': 4}
];

var postData = {
    Email: 'user@email.com',
    Pass: '****',
    AddressID: 1,
    DeliveryTimeID: 1,
    Lang: 'ru',
    IsPaperWork: 1,
    EDRPOU: 11111111,
    Specs: specArray
};

$.ajax({
    url: 'https://connect.erc.ua/connectservice/api/Documents/CreateOrder/',
    data: postData,
    type: 'POST',
    contentType: 'application/json',
    dataType: 'json',

    success: function(jdata) {
if (jdata.hasOwnProperty('d')) {
    jdata = jdata.d;
}
var errorCode = jdata.ErrorCode;
    var mess = jdata.ResultMessages;
},
    loadError: function(xhr, st, err) {
alert(xhr.responseText);
}
});

```

## 9. Отримання замовлень

### 9.1 Опис

Цей метод дозволяє отримати детальну інформацію про замовлення (специфікація замовлення, кількість відвантаженого товару, об'єм).

### 9.2 Адреса сервіса

<https://connect.erc.ua/connectservice/api/Documents/GetOrder/>

### 9.3 Вхідні параметри

Найменування	Тип	Опис
--------------	-----	------

Email	String	Email користувача
Pass	String	Пароль користувача
OrderNum	Int	Номер замовлення B2B (опціонально)
DocExtNum	String	Номер замовлення облікової системи дилера (опціонально)

## 9.4 Приклад даних, що повертаються

Для JSON

```
{
  "DocExtNum": "00011111_2020",
  "DocDate": "2020-03-25T14:59:00",
  "DocId": 11111111,
  "DocNum": "1/11111111",
  "OrderNum": "2222222",
  "IsDDP": true,
  "Specs": [
    {
      "Ware": "WK24EUP0",
      "WareName": "Електрочайник HOTPOINT ARISTON WK 24E UP0",
      "VendorGroupName": "Hotpoint-Ariston",
      "WareGroupName": "Електрочайники",
      "Quantity": 1,
      "Fact": 1,
      "Price": XXXX.0,
      "Sum": XXXX.0,
      "Volume": 0.014872,
      "Weight": 1.9
    }
  ],
  "IsError": false,
  "ErrorCode": 0,
  "ResultMessages": "OK",
  "Data": null
}
```

## 9.5 Поля відповіді

Найменування	Опис
ErrorCode	Код помилки (0-помилки немає, відмінний від 0 – помилка)
IsError	Булевий результат (true – помилка створення замовлення, false – заказ створено)
ResultMessages	Текст помилки, або "OK", при вдалому виконанні запиту
DocId	ID документа ERC

DocNum	Номер документа ERC
OrderNum	Номер замовлення B2B
IsDDP	Валюта замовлення. Можливі значення: true – гривневе замовлення, false – долларове замовлення
DocExtNum	Номер замовлення облікової системи дилера
Specs	Специфікація документа
Атрибути специфікації документа	
Ware	Код позиції
WareName	Назва позиції
VendorGroupName	Вендор
WareGroupName	Категорія
Quantity	Кількість
Fact	Фактично відвантажено
Price	Ціна
Sum	Сума
Volume	Об'єм
Weight	Вага

## 9.6 Приклади виклику

Для PHP (Online запуск php скриптів: <https://ideone.com/>)

```
<?php
$email = 'user@email.com';
$password = '****';
$docExtNum = 1111111111;

$data = array("Email" => $email, "Pass" => $password, "DocExtNum" => $docExtNum);
$data_string = json_encode($data);

$ch = curl_init('https://connect.erc.ua/connectservice/api/Documents/GetOrder');
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "POST");
curl_setopt($ch, CURLOPT_POSTFIELDS, $data_string);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
    'Content-Type: application/json',
    'Content-Length: ' . strlen($data_string)
));
//execute post
$result = curl_exec($ch);
//close connection
curl_close($ch);
echo $result;
?>
```

Для C#

```
public async Task<JObject> CallGetOrder()
{
    var jsonParam = new JObject
    {
        ["Email"] = "user@email.com",
        ["Pass"] = "****",
        ["DocExtNum"] = 1111111111
    };

    var credentials = CredentialCache.DefaultCredentials;
    var handler = new HttpClientHandler { Credentials = credentials };

    using (var client = new HttpClient(handler))
    {
        client.BaseAddress = new Uri("https://connect.erc.ua/connectservice/");
        client.DefaultRequestHeaders.Accept.Clear();
        client.DefaultRequestHeaders.Accept.Add(new
        MediaTypeWithQualityHeaderValue("application/json"));

        var response = client.PostAsJsonAsync("api/Documents/GetOrder", jsonParam).Result;

        if (response.IsSuccessStatusCode)
        {
            var jsonString = await response.Content.ReadAsStringAsync();
            try
            {
                var retJson = JObject.Parse(jsonString);
                return retJson;
            }
            catch (Exception ex)
            {
                return null;
            }
        }
    }

    return null;
}
```

Для JavaScript/AJAX

```
var postData = {
    Email: 'user@email.com',
    Pass: '****',
    DocExtNum: 1111111111
};
```

```
$.ajax({
  url: 'https://connect.erc.ua/connectservice/api/Documents/GetOrder/',
  data: postData,
  type: 'POST',
  contentType: 'application/json',
  dataType: 'json',

  success: function(jdata) {
    if (jdata.hasOwnProperty('d')) {
      jdata = jdata.d;
    }
    var errorCode = jdata.ErrorCode;
    var mess = jdata.ResultMessages;
  },
  loadError: function(xhr, st, err) {
    alert(xhr.responseText);
  }
});
```



## 10. Резерви

### 10.1 Опис

Цей метод дозволяє отримати інформацію про свої резерви (за період не більше 14 днів).

### 10.2 Адреса сервіса

<https://connect.erc.ua/connectservice/api/Documents/GetReserves>

### 10.3 Вхідні параметри

Найменування	Тип	Опис
Email	String	Email користувача
Pass	String	Пароль користувача
IsJson	Boolean	Не обов'язковий параметр (опціонально - при відсутності результат буде вивантажено в XML). Можливі значення: 0, 1, true, false
DateFrom	Data	Дата «з». Формат дати "dd.ММ.уууу" або "ММ-dd-уууу" або "ууууММdd".
DateTo	Data	Дата «по». Формат дати "dd.ММ.уууу" або "ММ-dd-уууу" або "ууууММdd".

### 10.4 Приклад даних, що повертаються

Результат доступний в XML і в JSON форматі.

Для XML

```

<a_Reserves>
  <a_Reserve Num="номер счета">
    <DateFrom>Дата начала резерва</DateFrom>
    <DateTo>Дата окончания резерва</DateTo>
    <Ware>Код позиции</Ware>
    <WareName>Найменування позиции</WareName>
    <Vendor> Вендор</Vendor>
    <Category>Категория</Category>
    <Quantity>Кол-во</Quantity>
    <Price>Цена</Price>
    <Sum>Сумма</Sum>
    <Currency>Валюта</Currency>
  </a_Reserve>
  ...
</a_Reserves>

```

Для JSON

```
[
  {
```

```

"DocNum": "1/11111111",
>DateFrom": "2020-05-01T00:00:00",
>DateTo": "2020-06-09T00:00:00",
>Days": 40,
>Town": "Київ",
>Spec": [
  {
    "Ware": "5E1100IUSB",
    "Name": "Джерело безперебійного живлення Eaton 5E1100IUSB",
    "VendorGroupName": "Eaton",
    "WareGroupName": "ДБЖ побутові та комп'ютерів",
    "Quantity": "4",
    "Price": XXXX.0,
    "Sum": XXXX.0,
    "Currency": "UAH"
  }
]
},
...
]

```

### 10.5 Поля відповіді

Найменування	Опис
a_Reserve Num	Номер рахунку
DateFrom	Дата початку резерва
DateTo	Дата закінчення резерва
Ware	Код позиції
WareName	Найменування позиції
Vendor	Вендор
Category	Категорія
Quantity	Кількість
Price	Ціна
Sum	Сума
Currency	Валюта

### 10.6 Приклади виклику

Для PHP (Online запуск php скриптів: <https://ideone.com/>)

```

<?php

>Email = 'user@email.com';
>Pass = '****';
>DateFrom= '01.03.2018';
>DateTo= '10.03.2018';

```

```

$data = array("Email" => $Email, "Pass" => $Pass, "DateFrom" => $DateFrom, "DateTo" =>
$dateTo);
$data_string = json_encode($data);

$ch = curl_init(https://connect.erc.ua/connectservice/api/Documents/GetReserves);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "POST");
curl_setopt($ch, CURLOPT_POSTFIELDS, $data_string);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
    'Content-Type: application/json',
    'Content-Length: ' . strlen($data_string)
));
curl_setopt($ch, CURLOPT_TIMEOUT, 50);
curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 50);

//execute post
$result = curl_exec($ch);

//close connection
curl_close($ch);

echo $result;

?>

```

Для C#

```

publicclass DocumentsPriceClient
{
    publicstring Email { get; set; }
    publicstring Pass { get; set; }
    publicDateTime DateFrom { get; set; }
    publicDateTime DateTo { get; set; }
}

var user = new DocumentsPriceClient() { Email = "user@email.com", Pass =
"****", DateFrom = DateTime.Now.AddDays(-10), DateTo = DateTime.Now };
using (var httpClient = new HttpClient())
{
    var requestUri =
"https://connect.erc.ua/connectservice/api/Documents/GetReserves";
    var response = httpClient.PostAsJsonAsync(requestUri, user).Result;
    var xml = response.Content.ReadAsStringAsync().Result;

    return response;
}

```

## Для AJAX

```
var json = JSON.stringify({ "Email": 'user@email.com', "Pass": '****', "DateFrom":  
'01.03.2018', "DateTo": '10.03.2018' });  
  
$.ajax({  
  url: 'https://connect.erc.ua/connectservice/api/Documents/GetReserves',  
  type: 'post',  
  contentType: 'application/json', // type of data  
  data: json,  
  dataType: 'xml',  
  success: function (jdata) {  
    if (jdata.hasOwnProperty('d')) {  
      jdata = jdata.d;  
    }  
    var xml = jdata.xml;  
  },  
  
  error: function(errorObj) {  
    alert(errorObj.responseText);  
  }  
}
```

## 11. Відвантажувальні накладні

### 11.1 Опис

Цей метод дозволяє отримати інформацію про відвантажувальні накладні (за період не більше 14 днів).

### 11.2 Адреса сервіса

<https://connect.erc.ua/connectservice/api/Documents/GetOutlays>

### 11.3 Вхідні параметри

Найменування	Тип	Опис
Email	String	Email користувача
Pass	String	Пароль користувача
IsJson	Boolean	Не обов'язковий параметр (опціонально - при відсутності результат буде вивантажено в XML). Можливі значення: 0, 1, true, false
DateFrom	Data	Дата «з». Формат дати "dd.ММ.уууу" або "ММ-dd-уууу" або "ууууММdd".
DateTo	Data	Дата «по». Формат дати "dd.ММ.уууу" або "ММ-dd-уууу" або "ууууММdd".

### 11.4 Приклад даних, що повертаються

Результат доступний в XML і в JSON форматі.

Для XML

```

<a_Outlays>
  <a_Outlay Num="номер накладной" RSB_Num="Номер учетной системы"
  Date="Дата накладной" Status="Статус документа">
    <Address>"Адрес доставки"</Address>
    <Contact Name="Грузополучатель" Document="Документы
    грузополучателя"/>
    <Sum>Сумма документа</Sum>
    <Currency>Валюта документа</Currency>
    <Specs>
      <Ware Code="Код позиции" Name="Найменування позиції"
      VendorGroupName="Вендор" WareGroupName="Категорія"
      Quantity="Кол-во" Fact="Отгружено" Price="Цена" Sum="Сумма"
      SN="Серийный номер позиции" Volume="Объем" Weight="Вес" />

      <Ware Code="Код позиции" Name="Найменування позиції"
      VendorGroupName="Вендор" WareGroupName="Категорія"
      Quantity="Кол-во" Fact="Отгружено" Price="Цена" Sum="Сумма"
      SN="Серийный номер позиции" Volume="Объем" Weight="Вес" />
      ...
    </Specs>
  </a_Outlay>

```

...

</a\_Outlays>

### Для JSON

```
[
  {
    "DocNum": "1/111111",
    "RSB_Num": "1/11111",
    "Date": "2020-03-20T18:10:00",
    "Status": "Відвантажено",
    "Address": " м. Київ, вул Святкова, буд.А",
    "ContactName": "Петренко Петро",
    "ContactDocument": "паспорт АА111111 от 23.04.99",
    "Sum": "X,00",
    "Currency": "USD",
    "Spec": [
      {
        "Ware": "31030014404",
        "Name": "Миша Genius NX-7010 WL Turquoise",
        "VendorGroupName": "GENIUS",
        "WareGroupName": "Миші бездротові",
        "Quantity": 1,
        "Fact": 1,
        "Price": " X,00",
        "Sum": " X,00",
        "SN": "X8D95236500045",
        "Volume": 0.00132,
        "Weight": 0.15
      }
    ]
  },
  ...
]
```

### 11.5 Поля відповіді

Найменування	Опис
a_Outlay Num	Номер накладної
RSB_Num	Номер облікової системи
Date	Дата накладної
Status	Статус документа
Address	Адреса доставки
Contact Name	Вантажоотримувач
Document	Документивантажоотримувача
Sum	Сума документу
Currency	Валюта документу

Specs	Специфікація
Атрибути специфікації	
Ware Code	Код позиції
Name	Найменування позиції
VendorGroupName	Вендор
WareGroupName	Категорія
Quantity	Кількість
Fact	Відвантажено
Price	Ціна
Sum	Сума
SN	Серійний номер позиції
Volume	Об'єм
Weight	Вага

## 11.6 Приклади виклику

Для PHP (Online запуск php скриптів: <https://ideone.com/>)

```
<?php

$Email = 'user@email.com';
$Pass = '****';
$dateFrom = '01.03.2018';
$dateTo = '10.03.2018';

$data = array("Email" => $Email, "Pass" => $Pass, "DateFrom" => $dateFrom, "DateTo" =>
$dateTo);
$data_string = json_encode($data);

$ch = curl_init(https://connect.erc.ua/connectservice/api/Documents/GetOutlays);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "POST");
curl_setopt($ch, CURLOPT_POSTFIELDS, $data_string);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
    'Content-Type: application/json',
    'Content-Length: ' . strlen($data_string)
));
curl_setopt($ch, CURLOPT_TIMEOUT, 50);
curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 50);

//execute post
$result = curl_exec($ch);

//close connection
```

```
curl_close($ch);

echo $result;

?>
```

## Для C#

```
publicclass SpecPriceClient
{
    publicstring Email { get; set; }
    publicstring Pass { get; set; }
    publicDateTime DateFrom { get; set; }
    publicDateTime DateTo { get; set; }
}

var user = new SpecPriceClient() { Email = "user@email.com", Pass = "****",
    DateFrom = DateTime.Now.AddDays(-10), DateTo = DateTime.Now };
using (var httpClient = new HttpClient())
{
    var requestUri =
"https://connect.erc.ua/connectservice/api/Documents/GetOutlays";
    var response = httpClient.PostAsJsonAsync(requestUri, user).Result;
    var xml = response.Content.ReadAsStringAsync().Result;

    return response;
}
```

## Для JavaScript/AJAX

```
var json = JSON.stringify({ "Email": 'user@email.com', "Pass": '****', "DateFrom":
'01.03.2018', "DateTo": '10.03.2018' });

$.ajax({
    url: "https://connect.erc.ua/connectservice/api/Documents/GetOutlays",
    type: 'post',
    contentType: 'application/json', // type of data
    data: json,
    dataType: 'xml',
    success: function (jdata) {
        if (jdata.hasOwnProperty('d')) {
            jdata = jdata.d;
        }
        var xml = jdata.xml;
    },

    error: function(errorObj) {
        alert(errorObj.responseText);
    }
}
```



## 12. Бухгалтерські рахунки-фактури

### 12.1 Опис

Цей метод дозволяє отримати інформацію про виписані рахунки-фактури (за період не більше 14 днів)

### 12.2 Адреса сервіса

<https://connect.erc.ua/connectservice/api/Documents/GetBills>

### 12.3 Вхідні параметри

Найменування	Тип	Опис
Email	String	Email користувача
Pass	String	Пароль користувача
IsJson	Boolean	Не обов'язковий параметр (опціонально - при відсутності результат буде вивантажено в XML). Можливі значення: 0, 1, true, false
DocExtNum	String	Номер замовлення облікової системи дилера (опціонально)
DateFrom	Data	Дата «з». Формат дати "dd.ММ.уууу" або "ММ-dd-уууу" або "ууууММdd".
DateTo	Data	Дата «по». Формат дати "dd.ММ.уууу" або "ММ-dd-уууу" або "ууууММdd".

### 12.4 Приклад даних, що повертаються

Результат доступний в XML і в JSON форматі.

Для XML

```

<a_Bills>
  <a_Bill Num="номер счета" Date="Дата счета">
    <Supplier Name="Найменування поставщика" EDRPOU="ЕДРПОУ
поставщика"/>
    <Payer Name="Найменування плательщика" EDRPOU="ЕДРПОУ
плательщика"/>
    <Agreement>Договор</Agreement>
    <AccountNum>Номер р.с. поставщика</AccountNum>
    <Bank Name=" Найменування банка"/>
    <SumWithoutVat>Сумма без НДС</SumWithoutVat>
    <SumVat>НДС по документу</SumVat>
    <Sign>Подпись(кем подписан документ)</Sign>
    <Specs>
    <Ware Code="Код позиции" Name="Найменування позиції"
VendorGroupName="Вендор" WareGroupName="Категорія"
Quantity="Кол-во" PriceWithoutWat="Цена без НДС" VAT="НДС"/>
    <Ware Code="Код позиции" Name="Найменування позиції"

```

```

VendorGroupName="Вендор" WareGroupName="Категория"
Quantity="Кол-во" PriceWithoutWat="Цена без НДС" VAT="НДС"
TNVD="код ТНВД"/>
...
</Specs>
</a_Bill>
...
</a_Bills>

```

### Для JSON

```

[
  {
    "DocNum": "1/111111",
    "Date": "2019-11-20T00:00:00",
    "SupplierName": "\\I-AP-CI\\",
    "SupplierEdrpou": "99999999",
    "PayerName": "ТОВ \\ФІРМА\\",
    "PayerEdrpou": "99999999",
    "Agreement": "54/85-99",
    "AccountNum": "260000000000001",
    "Bank": "ПАТ \\Банк\\",
    "SumWithoutVat": "XXXX,00",
    "SumVat": "XXX,00",
    "Sign": "Артеменко Артем",
    "Spec": [
      {
        "Ware": "BKN-3",
        "Name": "Обігрівач теплова гармата Ballu BKN-3, 2200Вт, 20м2, мех. керування, IP24, жовтий",
        "TNVD": "( 8516 29 91 00 )",
        "VendorGroupName": "Ballu",
        "WareGroupName": "Обігрівачі",
        "Quantity": 10.0,
        "PriceWithoutVat": "XXXX,00",
        "VAT": "XXX,00"
      }
    ]
  },
  ...
]

```

### 12.5 Поля відповіді

Найменування	Опис
Supplier Name	Найменуванняпостачальника
EDRPOU	ЄДРПОУ постачальника
Payer Name	Найменування платника
EDRPOU	ЄДРПОУ платника

Agreement	Договір
AccountNum	Номер р.р. постачальника
Bank Name	Найменування банку
SumWithoutVat	Сума без ПДВ
SumVat	ПДВ по документу
Sign	Підпис (ким підписан документ)
Specs	Специфікація рахунку
<b>Атрибути специфікації</b>	
Ware Code	Код позиції
Name	Найменування позиції
VendorGroupName	Вендор
WareGroupName	Категорія
Quantity	Кількість
PriceWithoutWat	Ціна без ПДВ
VAT	ПДВ
TNVD	Код ТНВЕД, для послуги виводиться значення "УСЛ", для віртуального товару - значення "ВТ", для не імпортного товару (внутрішні закупки) - значення "ВЗ"

## 12.6 Приклади виклику

Для PHP (Online запуск php скриптів: <https://ideone.com/>)

```

PHP:
<?php

$email = 'user@email.com';
$pass = '****';
$dateFrom = '01.03.2018';
$dateTo = '10.03.2018';

$data = array("Email" => $email, "Pass" => $pass, "DateFrom" => $dateFrom, "DateTo" =>
$dateTo);
$data_string = json_encode($data);

$ch = curl_init(https://connect.erc.ua/connectservice/api/Documents/GetBills);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "POST");
curl_setopt($ch, CURLOPT_POSTFIELDS, $data_string);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
    'Content-Type: application/json',
    'Content-Length: ' . strlen($data_string)
));
curl_setopt($ch, CURLOPT_TIMEOUT, 50);

```

```

curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 50);

//execute post
$result = curl_exec($ch);

//close connection
curl_close($ch);

echo $result;

?>

```

### Для C#

```

publicclass SpecPriceClient
{
    publicstring Email { get; set; }
    publicstring Pass { get; set; }
    publicDateTime DateFrom { get; set; }
    publicDateTime DateTo { get; set; }
}

var user = new SpecPriceClient() { Email = "user@email.com", Pass = "*****",
DateFrom = DateTime.Now.AddDays(-10), DateTo = DateTime.Now };
using (var httpClient = new HttpClient())
{
    var requestUri = "https://connect.erc.ua/connectservice/api/Documents/GetBills";
    var response = httpClient.PostAsJsonAsync(requestUri, user).Result;
    var xml = response.Content.ReadAsStringAsync().Result;

    return response;
}

```

### Для JavaScript/AJAX

```

var json = JSON.stringify({ "Email": 'user@email.com', "Pass": '*****', "DateFrom":
'01.03.2018', "DateTo": '10.03.2018' });

$.ajax({
    url: 'https://connect.erc.ua/connectservice/api/Documents/GetBills',
    type: 'post',
    contentType: 'application/json', // type of data
    data: json,
    dataType: 'xml',
    success: function (jdata) {
        if (jdata.hasOwnProperty('d')) {
            jdata = jdata.d;
        }
        var xml = jdata.xml;
    },
},

```

```
error: function(errorObj) {  
    alert(errorObj.responseText);  
}
```

## 13. Бухгалтерські видаткові накладні

### 13.1 Опис

Цей метод дозволяє отримати інформацію про виписані видаткові накладні (за період не більше 14 днів).

### 13.2 Адреса сервіса

<https://connect.erc.ua/connectservice/api/Documents/GetInvoices>

### 13.3 Вхідні параметри

Найменування	Тип	Опис
Email	String	Email користувача
Pass	String	Пароль користувача
IsJson	Boolean	Не обов'язковий параметр (опціонально - при відсутності результат буде вивантажено в XML). Можливі значення: 0, 1, true, false
DateFrom	Data	Дата «з». Формат дати "dd.ММ.уууу" або "ММ-dd-уууу" або "ууууММdd".
DateTo	Data	Дата «по». Формат дати "dd.ММ.уууу" або "ММ-dd-уууу" або "ууууММdd".

### 13.4 Приклад даних, що повертаються

Результат доступний в XML і в JSON форматі.

Для XML

```

<a_Invoices>
  <a_Invoice Num="номер накладной" Date="Дата накладной">
    <Supplier Name="Найменування поставщика" EDRPOU="ЄДРПОУ поставщика"/>
    <Payee Name="Найменування плательщика" EDRPOU="ЄДРПОУ плательщика"/>
    <Agreement>Договор</Agreement>
    <PowerOfAttorney>Доверенность</PowerOfAttorney>
    <SumWithoutVat>Сумма без НДС</SumWithoutVat>
    <SumVat>НДС по документу</SumVat>
    <Sign>Подпись(кем подписан документ)</Sign>
    <Specs>
      <Ware Code="Код позиции" Name="Найменування позиції" VendorGroupName="Вендор" WareGroupName="Категорія" Quantity="Кол-во" PriceWithoutWat="Цена без НДС" VAT="НДС"/>
      <Ware Code="Код позиции" Name="Найменування позиції" VendorGroupName="Вендор" WareGroupName="Категорія" Quantity="Кол-во" PriceWithoutWat="Цена без НДС" VAT="НДС" TNVD="код ТНВД"/>
    ...
  </a_Invoice>
</a_Invoices>

```

```

...
</Specs>
</a_Invoice>
...
</a_Invoices>

```

## Для JSON

```

[
  {
    "DocNum": "1/111111",
    "Date": "2019-12-12T00:00:00",
    "SupplierName": "\"I-AP-CI\"",
    "SupplierEdrpou": "99999999",
    "PayerName": "ТОВ \"Фірма\"",
    "PayerEdrpou": "99999999",
    "Agreement": "58-68/9",
    "PowerOfAttorney": "Ген. 88 від 03.07.2019",
    "SumWithoutVat": "XXXX,00",
    "SumVat": "XXX,00",
    "Sign": "Артеменко Артем",
    "Spec": [
      {
        "Ware": "BKN-3",
        "Name": "Обігрівач теплога гармата Ballu BKN-3, 2200Вт, 20м2, мех. керування, IP24, жовтий",
        "TNVD": "( 8516 29 91 00 )",
        "VendorGroupName": "Ballu",
        "WareGroupName": "Обігрівачі",
        "Quantity": 10.0,
        "PriceWithoutVat": "XXXX,00",
        "VAT": "XXX,00"
      }
    ]
  },
  ...
]

```

## 13.5 Поля відповіді

Найменування	Опис
a Invoice Num	Номер накладної
Date	Дата накладної
Supplier Name	Найменуванняпостачальника
EDRPOU	ЄДРПОУ постачальника
Payer Name	Найменуванняплатника
EDRPOU	ЄДРПОУ платника
Agreement	Договір
PowerOfAttorney	Довіренність

SumWithoutVat	Сума без ПДВ
SumVat	ПДВ по документу
Sign	Підпис (ким підписан документ)
Specs	Специфікація видаткової накладної
Атрибути специфікації	
Ware Code	Код позиції
Name	Найменування позиції
VendorGroupName	Вендор
WareGroupName	Категорія
Quantity	Кількість
PriceWithoutVat	Ціна без ПДВ
VAT	ПДВ
TNVD	Код ТНВЕД, для послуги виводиться значення "УСЛ", для віртуального товару -значення "ВТ", для не імпортного товару (внутрішні закупки) - значення "ВЗ"

### 13.6 Приклади виклику

Для PHP (Online запуск php скриптів: <https://ideone.com/>)

```
<?php

$email = 'user@email.com';
$pass = '****';
$dateFrom = '01.03.2018';
$dateTo = '10.03.2018';

$data = array("Email" => $email, "Pass" => $pass, "DateFrom" => $dateFrom, "DateTo" =>
$dateTo);
$data_string = json_encode($data);

$ch = curl_init(https://connect.erc.ua/connectservice/api/Documents/GetInvoices);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "POST");
curl_setopt($ch, CURLOPT_POSTFIELDS, $data_string);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
    'Content-Type: application/json',
    'Content-Length: ' . strlen($data_string)
));
curl_setopt($ch, CURLOPT_TIMEOUT, 50);
curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 50);

//execute post
```



```

$result = curl_exec($ch);

//close connection
curl_close($ch);

echo $result;

?>

```

Для С#

```

publicclass SpecPriceClient
{
    publicstring Email { get; set; }
    publicstring Pass { get; set; }
    publicDateTime DateFrom { get; set; }
    publicDateTime DateTo { get; set; }
}

var user = new SpecPriceClient() { Email = "user@email.com", Pass = "*****",
DateFrom = DateTime.Now.AddDays(-10), DateTo = DateTime.Now };
using (var httpClient = new HttpClient())
{
    var requestUri =
"https://connect.erc.ua/connectservice/api/Documents/GetInvoices";
    var response = httpClient.PostAsJsonAsync(requestUri, user).Result;
    var xml = response.Content.ReadAsStringAsync().Result;

    return response;
}

```

Для JavaScript/AJAX

```

var json = JSON.stringify({ "Email": 'user@email.com', "Pass": '*****', "DateFrom":
'01.03.2018', "DateTo": '10.03.2018' });

$.ajax({
    url: 'https://connect.erc.ua/connectservice/api/Documents/GetInvoices',
    type: 'post',
    contentType: 'application/json', // type of data
    data: json,
    dataType: 'xml',
    success: function (jdata) {
        if (jdata.hasOwnProperty('d')) {
            jdata = jdata.d;
        }
        var xml = jdata.xml;
    },

    error: function(errorObj) {
        alert(errorObj.responseText);
    }
});

```

```
}  
}
```

## 14. Акт зміни ціни

### 14.1 Опис

Цей метод дозволяє отримати інформацію про виписані акти зміни ціни (за період не більше 14 днів).

### 14.2 Адреса сервіса

<https://connect.erc.ua/connectservice/api/Documents/GetValuationAdjustments>

### 14.3 Вхідні параметри

Найменування	Тип	Опис
Email	String	Email користувача
Pass	String	Пароль користувача
IsJson	Boolean	Не обов'язковий параметр (опціонально - при відсутності результат буде вивантажено в XML). Можливі значення: 0, 1, true, false.
DateFrom	Data	Дата «з». Формат дати "dd.ММ.уууу" або "ММ-dd-уууу" або "ууууММdd".
DateTo	Data	Дата «по». Формат дати "dd.ММ.уууу" або "ММ-dd-уууу" або "ууууММdd".

### 14.4 Приклад даних, що повертаються

Результат доступний в XML і в JSON форматі.

Для JSON

```
[
  {
    "DocNum": "11111",
    "ExtNum": "1/11111",
    "Date": "2020-02-08T00:00:00",
    "SupplierName": "I-AP-CI",
    "SupplierEdrpou": null,
    "PayerName": "ФІРМА ",
    "PayerEdrpou": null,
    "SumWithoutVat": "",
    "SumVat": "-XXXX,00",
    "Spec": [
      {
        "Ware": null,
        "Name": "Принтер A4 Xerox Phaser 3020BI (Wi-Fi)",
        "Vendor": null,
        "SubCategory": null,
        "Quantity": 50,
        "Price": "XXXXX,00",

```

```

        "Sum": "XXXXX,00",
        "NewPrice": "XXXXX,00",
        "NewSum": ""
    }
}
}
...
]

```

### 14.5 Поля відповіді

Найменування	Опис
DocNum	Номер документа
ExtNum	Номер Акта
Date	Дата Акта
Supplier Name	Найменуванняпостачальника
SupplierEdrpou	ЄДРПОУ постачальника
Payer Name	Найменуванняплатника
PayerEdrpou	ЄДРПОУ платника
SumWithoutVat	Сума без ПДВ
SumVat	ПДВ по документу
Sign	Підпис (ким підписан документ)
Specs	Специфікація Акта
<b>Атрибути специфікації</b>	
Ware	Код позиції
Name	Найменування позиції
Vendor	Вендор
SubCategory	Категорія
Quantity	Кількість
Price	Ціна
Sum	Сума
NewPrice	Нова ціна
NewSum	Нова сума

### 14.6 Приклади виклику

Для PHP (Online запуск php скриптів: <https://ideone.com/>)

```

<?php

$email = 'user@email.com';
$pass = '****';

```

```

$dateFrom= '01.03.2018';
$dateTo= '10.03.2018';

$data = array("Email" => $Email, "Pass" => $Pass, "DateFrom" => $DateFrom, "DateTo" =>
$dateTo);
$data_string = json_encode($data);

$ch =
curl_init(https://connect.erc.ua/connectservice/api/Documents/GetValuationAdjustments);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "POST");
curl_setopt($ch, CURLOPT_POSTFIELDS, $data_string);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
'Content-Type: application/json',
'Content-Length: ' . strlen($data_string)
));
curl_setopt($ch, CURLOPT_TIMEOUT, 50);
curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 50);

//execute post
$result = curl_exec($ch);

//close connection
curl_close($ch);

echo $result;

?>

```

Для C#

```

publicclass SpecPriceClient
{
    publicstring Email { get; set; }
    publicstring Pass { get; set; }
    publicDateTime DateFrom { get; set; }
    publicDateTime DateTo { get; set; }
}

var user = new SpecPriceClient() { Email = "user@email.com", Pass = "*****",
DateFrom = DateTime.Now.AddDays(-10), DateTo = DateTime.Now };
using (var httpClient = new HttpClient())
{
    var requestUri =
"https://connect.erc.ua/connectservice/api/Documents/GetValuationAdjustments";
    var response = httpClient.PostAsJsonAsync(requestUri, user).Result;
    var xml = response.Content.ReadAsStringAsync().Result;
}

```

```
    return response;
}
```

### Для JavaScript/AJAX

```
var json = JSON.stringify({ "Email": 'user@email.com', "Pass": '****', "DateFrom":
'01.03.2018', "DateTo": '10.03.2018' });

$.ajax({
  url:
'https://connect.erc.ua/connectservice/api/Documents/GetValuationAdjustments',
  type: 'post',
  contentType: 'application/json', // type of data
  data: json,
  dataType: 'xml',
  success: function (jdata) {
    if (jdata.hasOwnProperty('d')) {
      jdata = jdata.d;
    }
    var xml = jdata.xml;
  },

  error: function(errorObj) {
    alert(errorObj.responseText);
  }
}
```

## 15.Замовлення товару Microsoft ESD

### 15.1 Опис

Цей метод дозволяє створити авто-замовлення, що містить товари з категорії Microsoft ESD. Параметри для замовлення аналогічні параметрам для замовлення API Замовлення-самовивіз (див. розділ 7).

**Важливо!** Специфікація повинна містити тільки ESD позиції, інакше буде помилказамовлення: *"Всі позиції повинні бути ESD."*

У специфікації вказувати коди товарів з категорії «Електронні ключі».

*Примітка:* Для настройки взаємодії з API замовлення ESD і тестування - є можливість замовити тестовий ключ.

Для цього необхідно замовити ключ для позиції ABC-77777 (в замовленні повинна бути тільки тестова позиція), при цьому для тестової позиції бухгалтерські рахунки не створюються.

### 15.2 Адреса сервіса

<https://connect.erc.ua/connectservice/api/Documents/CreateOrderESD>

### 15.3 Вхідні параметри

Найменування	Тип	Опис
Email	String	Email користувача
Pass	String	Пароль користувача
INN	Int	ПІН вантажоотримувача клієнта (вантажоотримувач повинен бути активним на дату замовлення)
ContractId	Int	Id контракту (опціонально) (див п. 7.3)
Version	Int	Версійність результату створення замовлення (опціональний, по-замовчуванню "1.0")
DocExtNum	String	Номер замовлення облікової системи дилера (опціонально)
Specs		Масив специфікації код-позиції, кількість
Lang	String	Мова, на якій буде вивантажено результат (за замовчуванням - українська). Можливі значення: ru, ua, ru-ru, uk-ua (без урахування регістру)
<b>IsPaperWork</b>	Boolean	<b>Необхідне оформлення бухгалтерських документів (повідомлення Ведучого дилера-менеджера в ERC).</b>

		<b>Для використання цього параметру потрібно мати розширений доступ. Можливі значення: 1, 0</b>
EDRPOU	Int	ЄДРПОУ платника
IsEnvelopeESD	Boolean	Необхідний конверт з ліценцією (друкована форма)

## 15.4 Приклад даних, що повертаються

Для JSON

- При параметрі Version "1.0"

```
[
  {
    "DocId": 111111,
    "DocNum": "1/33333",
    "OrderNum": "22222",
    "PaperworkCurrencyRate": 28,5
    "Specs": {
      "Ware1": 3
    },
    "IsError": true,
    "ErrorCode": 100,
    "ResultMessages": "Погодження Создан новый \"Заказ на отгрузку\"(1/33333) B2B! | Документутвержден"
  },
  {
    "DocId": 777777,
    "DocNum": "1/33334",
    "OrderNum": "22223",
    "PaperworkCurrencyRate": 28,5
    "Specs": {
      "Ware2": 7,
      "Ware3": 1,
      "Ware4": 4
    },
    "IsError": false,
    "ErrorCode": 0,
    "ResultMessages": "Созданновый \"Заказнаотгрузку\"(1/33334) B2B! | Документ утвержден"
  }
]
```

- При параметрі Version "1.1" розділ Specs буде мати наступний вигляд:

```
"Specs":
[
  {"w": " Ware1", "q": 1, "p": 1111, "s": 1111},
  {"w": " Ware2", "q": 2, "p": 22222, "s": 44444},
]
```

В результаті додається ціна (p) та сума (s)



**15.5 Поля відповіді**

Найменування	Опис
DocNum	Номер створеного замовлення (може бути пустим рядком, якщо замовлення не створено)
DocId	Номер документа (облікової системи ERC)
OrderNum	Номер замовлення B2B
DocExtNum	Номер в обліковій системі партнера
PaperworkCurrencyRate	Курс виписки
ErrorCode	Код помилки (0-все в порядку, відмінний від 0 – помилка)
IsError	Булевий результат (true – помилка створення замовлення, false – замовлення створено)
ResultMessages	Текст помилки, або текст про вдале створення замовлення: "Запит на отримання ключів ESD прийнятий, для отримання ключів використовуйте сервіс: <a href="https://connect.erc.ua/connectservice/api/Documents/GetESD">https://connect.erc.ua/connectservice/api/Documents/GetESD</a> параметри: Email, Pass, OrderNum або DocExtNum"
Specs	Специфікація створеного документа або документа «на погодження»

**15.6 Приклади виклику**

Дивись розділ 7.4.6.

## 16.Отримання ключа товару Microsoft ESD

### 16.1 Опис

Цей метод дозволяє отримувати ключі для товару Microsoft ESD в створених раніше замовленнях. Приклад створення замовлення див. розділ 15.

*Примітка:* Для тестування можливості отримання ключів необхідно використовувати код товару: ABC-77777, при цьому для тестової позиції бухгалтерські рахунки не створюються.

### 16.2 Адреса сервіса

<https://connect.erc.ua/connectservice/api/Documents/GetESD/>

### 16.3 Вхідні параметри

Найменування	Тип	Опис
Email	String	Email користувача
Pass	String	Пароль користувача
OrderNum	Int	Номер замовлення B2B з електронними ключами (ESD)
DocExtNum	String	Номер замовлення облікової системи дилера
Version	Int	Версійність результату створення замовлення (опціональний, по-замовчуванню "1.0")

### 16.4 Приклад даних, що повертаються

Для JSON

- При параметріVersion"1.0"

```
[
  {
    "DocId": 111111,
    "DocNum": "1/33333",
    "OrderNum": "22222",
    "Specs": {
      "Ware1": 3
    },
    "IsError": true,
    "ErrorCode": 100,
    "ResultMessages": "Погодження Создан новый \"Заказ на отгрузку\"(1/33333) B2B! |
Документутвержден"
  },
  {
    "DocId": 777777,
    "DocNum": "1/33334",
    "OrderNum": "22223",
    "Specs": {
```

```

    "Ware2": 7,
    "Ware3": 1,
    "Ware4": 4
  },
  "IsError": false,
  "ErrorCode": 0,
  "ResultMessages": "Созданновый \Заказнаотгрузку\"(1/33334) B2B! | Документ утвержден"
}
]

```

- При параметрі Version "1.1" розділ Specs буде мати наступний вигляд:

```

"Specs":
[
  {"w": " Ware1", "q": 1, "p": 1111, "s": 1111},
  {"w": " Ware2", "q": 2, "p": 22222, "s": 44444},
]

```

В результаті додається ціна (p) та сума (s)

Для JSON

```

"Data": [
  {
    "Ware": "T5D-03189",
    "WareName": "Microsoft Office Home and Business 2019 All Languages (электронный ключ)",
    "Lic": "*****_*****_*****_*****_*****"
  }
]

```

## 16.5 Поля відповіді

Найменування	Опис
DocNum	Номер створеного замовлення (може бути пустим рядком, якщо замовлення не створено)
DocId	Номер документа (облікової системи ERC)
OrderNum	Номер замовлення B2B
ErrorCode	Код помилки (0-все в порядку, відмінний від 0 – помилка)
IsError	Булевий результат (true – помилка отримання ключів, false – ключі отримані)
ResultMessages	Текст помилки, або текст про вдале отримання ключів
Specs	Специфікація створеного документа або документа «на погодження»
Data	Масив даних ліцензії ESD
<b>Атрибути даних ліцензії</b>	
Ware	Код товару

WareName	Найменування товару
Lic	Номер ключа ліцензії

## **16.6 Приклади виклику**

Дивись розділ 7.4.6.

## 17. Створення замовлення для доставки товару з зазначенням ТТН (дропшипінг)

### 17.1 Опис

Цей метод дозволяє створювати авто-замовлення із зазначенням ТТН (дропшипінг) і відправляти його в збірку. Також для цього замовлення можливе створення рахунку-фактури.

### 17.2 Адреса сервіса

<https://connect.erc.ua/connectservice/api/Documents/CreateOrder/>

### 17.3 Вхідні параметри

Найменування	Тип	Опис
Email	String	Email користувача
Pass	String	Пароль користувача
Declaration	Int	Номер декларації для дроп замовлення
ContractId	Int	Id контракту (опціонально) (див. Отримання списку договорів п. 7.3)
Version	Int	Версійність результату створення замовлення (опціональний, по-замовчуванню "1.0")
DocExtNum	String	Номер замовлення облікової системи дилера (опціонально)
WarehouseID	Int	Обов'язковий. Створення замовлення у розрізі складу. Припустимі значення: 1, 5, 95, 96
Specs		Масив специфікації код-позиції, кількість
Lang	String	Мова, на якій буде вивантажено результат (за замовчуванням - українська). Можливі значення: ru, ua, ru-ru, uk-ua (без урахування регістру)
<b>IsPaperWork</b>	Boolean	<b>Необхідно оформлення бухгалтерських документів (повідомлення Ведучого дилер-менеджера в ERC)*.</b> <b>Для використання цього параметру потрібно мати розширений доступ.</b> <b>Можливі значення: 1, 0</b>
EDRPOU	Int	ЄДРПОУ платника (опціонально)
PayerID	Int	Код платника (опціонально)

Comment	String	Коментар по замовленню (опціонально)
---------	--------	--------------------------------------

\* **Важливо!** При необхідності оформлення бухгалтерських документів (параметр **IsPaperWork**) **обов'язково** необхідно вказати **хоча б один з параметрів**: EDRPOU або PayerID.

#### 17.4 Приклад даних, що повертаються

Для JSON

- При параметрі Version "1.0"

```
[
  {
    "DocId": 111111,
    "DocNum": "1/33333",
    "OrderNum": "22222",
    "IsDDP": true,
    "PaperworkCurrencyRate": 28,5
    "Specs": {
      "Ware1": 3
    },
    "IsError": true,
    "ErrorCode": 100,
    "ResultMessages": "Погодження Создан новый \"Заказ на отгрузку\"(1/33333) B2B! | Документутвержден"
  },
  {
    "DocId": 777777,
    "DocNum": "1/33334",
    "OrderNum": "22223",
    "IsDDP": true,
    "PaperworkCurrencyRate": 28,5
    "Specs": {
      "Ware2": 7,
      "Ware3": 1,
      "Ware4": 4
    },
    "IsError": false,
    "ErrorCode": 0,
    "ResultMessages": "Созданновый \"Заказнаотгрузку\"(1/33334) B2B! | Документ утвержден"
  }
]
```

- При параметрі Version "1.1" розділ Specs буде мати наступний вигляд:

```
"Specs":
[
  {"w": " Ware1", "q": 1, "p": 1111, "s": 1111},
  {"w": " Ware2", "q": 2, "p": 22222, "s": 44444},
]
```

В результаті додається ціна (p) та сума (s)

**17.5 Поля відповіді**

Найменування	Опис
DocNum	Номер створеного замовлення (може бути пустим рядком, якщо замовлення не створено)
OrderNum	Номер замовлення B2B
IsDDP	Валюта замовлення. Можливі значення: true – гривневе замовлення, false – долларове замовлення
PaperworkCurrencyRate	Курс виписки
ErrorCode	Код помилки (0-помилки немає, відмінний від 0 – помилка)
IsError	Булевий результат (true – помилка створення замовлення, false – заказ створено)
ResultMessages	Текст помилки, або текст про вдале створення замовлення
Specs	Специфікація створеного документа або документа «на погодження»

**17.6 Приклади виклику режим «Дропшипінг»**

Для PHP (Online запуск php скриптів: <https://ideone.com/>)

```
<?php
$Email = 'user@email.com';
$Pass = '****';
$Declaration = 11111111;
$Lang='ru';
$IsPaperWork=1;
$EDRPOU=11111111;

$specArray =
array(
    array(
        "w" =>"1M301-SPACEGRAY",
        "q" => 1
    ),
    array(
        "w" =>"E1003-SPACEGRAY",
        "q" => 4
    )
);

$data = array("Email" => $Email, "Pass" => $Pass, "INN" => $INN, "Lang" => $Lang,
"IsPaperWork" => $IsPaperWork, "EDRPOU" => $EDRPOU, "Specs" => $specArray);
$data_string = json_encode($data);
```

```

$ch = curl_init('https://connect.erc.ua/connectservice/api/Documents/CreateOrder');
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "POST");
curl_setopt($ch, CURLOPT_POSTFIELDS, $data_string);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
    'Content-Type: application/json',
    'Content-Length: ' . strlen($data_string)
));
//execute post
$result = curl_exec($ch);
//close connection
curl_close($ch);

echo $result;
?>

```

### Для C#

```

public async Task<JObject> CallCreateOrder()
{
    var specList = new List<object>()
    {
        new { w = "1M301-SPACEGRAY", q = 1 },
        new { w = "E1003-SPACEGRAY", q = 1 },
        new { w = "EO320-GOLD", q = 4 }
    };

    var specArrayJson = JObject.FromObject(specList);

    var jsonParam = new JObject
    {
        ["Email"] = "user@email.com",
        ["Pass"] = "*****",
        ["Declaration"] = 1111111111,
        ["Lang"] = "ru",
        ["IsPaperWork"] = 1,
        ["EDRPOU"] = 11111111
    };
    jsonParam.Add(new JProperty("Specs", specArrayJson));

    var credentials = CredentialCache.DefaultCredentials;
    var handler = new HttpClientHandler { Credentials = credentials };

    using (var client = new HttpClient(handler))
    {
        client.BaseAddress = new Uri("https://connect.erc.ua/connectservice/");
        client.DefaultRequestHeaders.Accept.Clear();
        client.DefaultRequestHeaders.Accept.Add(new

```



```

MediaTypeWithQualityHeaderValue("application/json"));

var response = client.PostAsJsonAsync("api/Documents/CreateOrder", jsonParam).Result;

if (response.IsSuccessStatusCode)
    {
var jsonString = await response.Content.ReadAsStringAsync();
try
    {
var retJson = JObject.Parse(jsonString);
return retJson;
    }
catch (Exception ex)
{
return null;
    }
}

return null;
}

```

### Для JavaScript/AJAX

```

var specArray = [
    { 'w': '1M301-SPACEGRAY', 'q': 1},
    { 'w': 'E1003-SPACEGRAY', 'q': 1},
    { 'w': 'EO320-GOLD', 'q': 4}
];

var postData = {
    Email: 'user@email.com',
    Pass: '****',
    Declaration: 1111111,
    Lang: 'ru',
    IsPaperWork: 1,
    EDRPOU: 11111111,
    Specs: specArray
};

$.ajax({
    url: 'https://connect.erc.ua/connectservice/api/Documents/CreateOrder/',
    data: postData,
    type: 'POST',
    contentType: 'application/json',
    dataType: 'json',

    success: function(jdata) {

```

```
if (jdata.hasOwnProperty('d')) {  
    jdata = jdata.d;  
}  
var errorCode = jdata.ErrorCode;  
    var mess = jdata.ResultMessages;  
},  
loadError: function(xhr, st, err) {  
alert(xhr.responseText);  
    }  
});
```

### Додаткова інформація

- Обмін відбувається в форматі JSON, для спец-прайса - в форматі XML і JSON.
- Для формату XML кількість запитів необмежено (відсутнє блокування за часті звернення).
- Замовлення обробляються в режимі реального часу.
- Залишки по складу і ціни на товар оновлюються раз в режимі реального часу.
- Є можливість отримання товару в розрізі вендорів, категорій.
- Є можливість тестового (перевірочного) отримання ключів для товару категорії Microsoft ESD.