

# API по работе интернет магазинов с Delivery.

## Версия 3.2.

### Оглавление

<b>Введение</b> .....	3
<b>1. Представительства</b> .....	3
1.1 <i>Получение списка областей - метод GetRegionList.</i> .....	3
1.2 <i>Получение списка городов – метод GetAreasList.</i> .....	4
1.3 <i>Получение списка представительств - метод GetWarehousesList.</i> .....	5
1.4 <i>Получение подробной информации о представительстве - метод GetWarehousesInfo.</i> .....	7
1.5 <i>Поиск представительств - метод GetFindWarehouses.</i> .....	8
1.6 <i>Получение списка представительств с подробной информацией по городу - метод GetWarehousesListInDetail.</i> .....	9
<b>2. Квитанции</b> .....	11
2.1 <i>Поиск квитанции - метод GetReceiptDetails.</i> .....	11
2.2 <i>Расчет сроков доставки - метод GetDateArrival.</i> .....	12
<b>3. Расчет стоимости перевозки</b> .....	14
3.1 <i>Модели данных для обмена информацией:</i> .....	14
3.2 <i>Вывод справочника категорий доп. услуг и входящих в них доп. услуг - метод GetDopUslugiClassification.</i> .....	16
3.3 <i>Вывод справочника категорий тарифа - метод GetTariffCategory.</i> .....	17
3.4 <i>Вывод справочника схем доставки - метод GetDeliveryScheme.</i> .....	18
3.5 <i>Расчет стоимости перевозки - метод PostReceiptCalculate.</i> .....	19
<b>4. Связь с пользователем</b> .....	23
4.1 <i>Модели данных для обмена информацией:</i> .....	23
4.2 <i>Получение новостей компании - метод GetNews.</i> .....	24
4.3 <i>Получение тем сообщений - метод GetMessagesTheme.</i> .....	25
4.4 <i>Отправка сообщения - метод PostContactsMessage.</i> .....	26
4.5 <i>Отправка оценки работы компании - метод PostServiceRate.</i> .....	27
4.6 <i>Отправка оценки работы компании - метод PostServiceRate.</i> .....	28
<b>5. Личный кабинет, регистрация</b> .....	29
5.1 <i>Авторизация:</i> .....	29
5.2 <i>Выход</i> .....	29
5.3 <i>Получение информации о пользователе.</i> .....	30

5.4	Получение квитанций пользователя – метод <i>GetUserReceipt</i> . .....	31
<b>6.</b>	<b>Оформление квитанции</b> .....	34
6.1	Получение доступа через API ключ и выбор формата данных. ....	34
6.2	Получение списка платежных карт клиента. ....	36
6.3	Получение списка расчетных счетов клиента. ....	37
6.4	Получение списка категорий отправляемого груза. ....	38
6.5	Оформление квитанции - метод <i>PostCreateReceipts</i> . ....	39
6.6	Деактивация единиц груза - метод <i>PostDeactivateEg</i> . ....	42
6.7	Получение документов в PDF - метод <i>GetPdfDocument</i> . ....	43
6.8	Получение списка отправителей (дочерних или родительских организаций клиента) - метод <i>GetSenderList</i> . ....	44
6.9	Получение схем доставки - метод <i>GetDeliveryScheme</i> . ....	45
6.10	Получение стоимости страхования - метод <i>GetInsuranceCost</i> . ....	46
6.11	Получение категорий груза - метод <i>GetCargoCategory</i> . ....	47
6.12	Получение доступных валют - метод <i>GetCurrency</i> . ....	48
6.13	Получение списка плательщиков - метод <i>GetPayer</i> . ....	49
6.14	Получение справочника категорий тарифа - метод <i>GetTariffCategory</i> . ....	50
6.15	Получение адресов клиента - метод <i>GetClientAddress</i> . ....	52
6.16	Получение возможных получателей клиента- метод <i>GetPosibleReciver</i> . ....	53
6.17	Получение вида оплаты клиента - метод <i>GetClientPaymentType</i> . ....	54
6.18	Получение списка складов/потоматов - метод <i>GetWarehousesListByCity</i> . ....	55
6.19	Получение полной информации по квитанции - метод <i>GetFullReceiptInformation</i> . ....	57
6.20	Создание адреса или получателя - метод <i>PostCreateAddressOrClient</i> . ....	60
6.21	Получение информации по стикеру квитанции - метод <i>GetStickers</i> . ....	62
6.22	Объединение квитанций в один заказ на забор - метод <i>PostAddReceiptIntoPickUpRequest</i> . 63	
6.23	Получение реестра отправки - метод <i>SendingRegistr</i> . ....	64
<b>7.</b>	<b>Операции с логами квитанции</b> .....	65
7.1	Получение логов по квитанции. ....	65
	<b>Дополнительные справочники</b> .....	66
	Справочник статусов квитанции .....	66
	Справочник валют .....	66
	Справочник кодов операции .....	66

# Введение

Формат для обмена данными json.

Адрес веб сервиса <http://www.delivery-auto.com/api/>

API предыдущей версии доступно по ссылке <http://www.delivery-auto.com/api/v1/> или <http://www.delivery-auto.com/api/>

Выходные параметры имеют формат:

```
{
  "status": true,
  "message": "",
  "data": Данные
}
```

При успешном выполнении команды status == true.

В случае возникновения исключения status == false, переменная message содержит описание ошибки.

Для ввода русскоязычных полей необходимо использовать кодировку Windows-1251.

## 1. Представительства

### 1.1 Получение списка областей - метод *GetRegionList*.

**GET api/v4/Public/GetRegionList?culture={culture}&country={country}**

#### Входные параметры

---

Наименование	Описание
culture	Культура; Допустимые значения (en-US, ru-RU, uk-UA). (String)
country	Id страны (1-Украина, 2-Россия, null - все)

---

#### Выходные параметры

Представляет собой json. Коллекция объектов {id, name, externalId}.

id - id области

name - Наименование области

externalId - Id области

#### Формат выходных параметров

*application/json, text/json*

#### Пример:

```
{
  "status": true,
  "message": "",
  "data": [
    {
      "id": -1,
      "name": "ВСЕ"
      "externalId": "00000000-0000-0000-0000-000000000000"
    }
  ]
}
```

```

    },
    {
      "id": 3898,
      "name": "Винницкая область",
      "externalId": "c8ad84fe-cf49-e211-9515-00155d012d0d"
    }
  ]
}

```

## 1.2 Получение списка городов – метод *GetAreasList*.

### **GET**

**api/v4/Public/GetAreasList?culture={culture}&fl\_all={fl\_all}&regionId={regionId}&country={country}**

### **Входные параметры**

---

Наименование	Описание
culture	Культура; Допустимые значения (en-US, ru-RU, uk-UA). (String)
fl_all	Необязательный параметр. Флажок позволяющий показывать все города где есть возможность оказания услуг компании (bool)
regionId	Id области
country	Id страны (1-Украина, 2-Россия)

---

### **Выходные параметры**

Представляет собой json. Коллекция объектов {id, name, RegionId, IsWarehouse, ExtracityPickup, ExtracityShipping, RAP, RAS, regionName, country, districtName}.

id - id города

name - Наименование города

RegionId – id области

IsWarehouse - Флаг склада (1-в городе есть склад)

ExtracityPickup - true = Осуществляется забор за город

ExtracityShipping - true = Осуществляется доставка за город

RAP - true = Региональный забор

RAS - true = Региональная доставка

regionName – название области

country – код страны (1 –Украина)

districtName – название района

## Формат выходных параметров

*application/json, text/json*

### Пример:

```
{
  "status": true,
  "message": "",
  "data": [
    {
      "id": "2d481888-1429-e311-8b0d-00155d037960",
      "name": "Бап",
      "RegionId": "c8ad84fe-cf49-e211-9515-00155d012d0d",
      "IsWarehouse": true,
      "ExtracityPickup": true,
      "ExtracityShipping": true,
      "RAP": false,
      "RAS": false,
      "regionName": "Винницкая область",
      "regionId": 3898,
      "country": 1,
      "districtName": "Александрыйский"
    },
    {
      "id": "45481888-1429-e311-8b0d-00155d037960",
      "name": "Березина",
      "RegionId": "c8ad84fe-cf49-e211-9515-00155d012d0d",
      "IsWarehouse": false,
      "ExtracityPickup": true,
      "ExtracityShipping": true,
      "RAP": false,
      "RAS": false,
      "regionName": "Винницкая область",
      "regionId": 3898,
      "country": 1,
      "districtName": "Арцизский"
    }
  ]
}
```

### 1.3 Получение списка представительств - метод *GetWarehousesList*.

#### **GET**

**[api/v4/Public/GetWarehousesList?culture={culture}&includeRegionalCenters={includeRegionalCenters}&CityId={CityId}&RegionId={RegionId}&country={country}](#)**

#### **Входные параметры**

---

Наименование	Описание
culture	Культура; Допустимые значения (en-US, ru-RU, uk-UA). (String)
includeRegionalCenters	Выводить офисы? (Boolean, Значение по умолчанию false)

---

Наименование	Описание
CityId	Id города. (Guid?, Значение по умолчанию null)
RegionId	Id области. (Guid?, Значение по умолчанию null)
country	Id страны (1-Украина, 2-Россия)

---

## Выходные параметры

Представляет собой json. Коллекция объектов {id, name, Latitude, Longitude, CityId}.

id - id склада

name - Наименование склада

Latitude - Широта

Longitude – Долгота

CityId – Id города

### Формат выходных параметров

*application/json, text/json*

### Пример:

```
{
  "status": true,
  "message": "",
  "data": [
    {
      "id": "1c828aa6-70c8-e211-9902-00155d037919",
      "name": "АВДЕЕВКА",
      "Latitude": 37.7349460000,
      "Longitude": 48.1386240000,
      "CityId": "b62a29dd-8c63-e611-80fd-000d3a204dc8"
    },
    {
      "id": "e627c8fd-d549-e211-9515-00155d012d0d",
      "name": "АЛЕКСАНДРИЯ",
      "Latitude": 33.1150260000,
      "Longitude": 48.6727020000,
      "CityId": "44f54093-d12a-e311-8b0d-00155d037960"
    },
    {
      "id": "ad3a6d45-b249-e211-ab75-00155d012d0d",
      "name": "АЖИУШТА",
      "Latitude": 34.4151950000,
      "Longitude": 44.6864660000,
      "CityId": "03690b6c-d649-e211-9515-00155d012d0d"
    }
  ]
}
```

## 1.4 Получение подробной информации о представительстве - метод *GetWarehousesInfo*.

**GET api/v4/Public/GetWarehousesInfo?culture={culture}&WarehousesId={WarehousesId}**

### Входные параметры

---

Наименование	Описание
culture	Культура; Допустимые значения (en-US, ru-RU, uk-UA). (String)
WarehousesId	Id представительства. (Guid)

---

### Выходные параметры

Представляет собой json. Коллекция объектов {id,name,address,operatingTime,Phone, EmailStorage,Latitude,Longitude,Office, CityId, CityName, IsWarehouse, RcPhoneSecurity, RcPhoneManagers, RcPhone, RcName, WarehouseForDeliveryId, WarehouseType}.

- id - id представительства
- name - Наименование представительства
- address - Адрес представительства
- operatingTime - Время работы представительства
- Phone - Телефонные номера представительства
- EmailStorage - Email представительства
- Latitude - Широта
- Longitude - Долгота
- Office - Признак офиса
- CityId – id города
- CityName – название города
- IsWarehouse – является ли складом
- RcPhoneSecurity – телефон отдела СБ
- RcPhoneManagers – телефон отдела менеджеров
- RcPhone – телефон регионального центра
- RcName – наименование регионального центра
- WarehouseForDeliveryId – склад для доставки
- WarehouseType – тип склада

### Формат выходных параметров

*application/json, text/json*

### Пример:

```
{
  "status": true,
  "message": "",
  "data": {
    "id": "e627c8fd-d549-e211-9515-00155d012d0d",
    "name": "АЛЕКСАНДРИЯ",
    "address": "ул. Дибровы, 16",
    "operatingTime": "ПН-ПТ: 9:00-18:00, СБ: 9:00-15:00",
    "Phone": "(067) 620-72-76, (05235) 7-12-44",
    "EmailStorage": "als@delivery-auto.com.ua",
```

```

    "Latitude": 33.1150260000,
    "Longitude": 48.6727020000,
    "Office": false,
    "CityId": "1e8e7257-a82a-e311-8b0d-00155d037960",
    "CityName": "Александрия",
    "IsWarehouse": true,
    "RcPhoneSecurity": "(067) 627-67-95",
    "RcPhoneManagers": "(047) 444-63-99",
    "RcPhone": "(047) 444-63-99",
    "RcName": "Центральный Региональный Центр - 2",
    "WarehouseForDeliveryId": null,
    "WarehouseType": 0
  }
}

```

## 1.5 Поиск представительств - метод `GetFindWarehouses`.

### **GET**

**`api/v4/Public/GetFindWarehouses?culture={culture}&Longitude={Longitude}&Latitude={Latitude}&count={count}&includeRegionalCenters={includeRegionalCenters}&CityId={CityId}`**

### **Входные параметры**

Наименование	Описание
culture	Культура; Допустимые значения (en-US, ru-RU, uk-UA). (String)
Longitude	Долгота. (Double)
Latitude	Широта. (Double)
count	Количество возвращаемых представительств. (Integer)
includeRegionalCenters	Выводить офисы. (Boolean)
CityId	Id города

### **Выходные параметры**

Представляет собой json. Коллекция объектов {id,name,distance,latitude,longitude, cityName, address, IsWarehouse, Phone, working\_time, WarehouseType, IsRegionalCentre}.

- id - id представительства
- name - Наименование представительства
- distance - Расстояние
- latitude - Широта
- longitude – Долгота
- cityName – название города
- address – адрес склада
- IsWarehouse – является ли складом



Phone – телефон склада  
working\_time – рабочее время  
WarehouseType – 0 – склад, 2-почтомат, 3-склад с наложенным платежом наличным  
IsRegionalCentre – является ли региональным центром  
IsCashOnDelivery – есть ли на складде услуга наложенного платежа

### Формат выходных параметров

*application/json, text/json*

#### Пример:

```
{
  "status": true,
  "message": "",
  "data": [
    {
      "id": "bbb06f68-3529-e311-8b0d-00155d037960",
      "name": "Курахово",
      "distance": 18.3,
      "longitude": 47.9876930000,
      "latitude": 37.2770390000,
      "cityName": "Курахово",
      "address": "ул. Березовая, 6 (ул. Чалова)",
      "IsWarehouse": true,
      "phone": "(067) 557-89-25",
      "working_time": "ПН-ПТ 9:00-18:00, СБ 9:00-17:00",
      "WarehouseType": 0,
      "IsRegionalCentre": false,
      "IsCashOnDelivery": true
    },
  ]
}
```

## 1.6 Получение списка представительств с подробной информацией по городу - метод *GetWarehousesListInDetail*.

### GET

**api/v4/Public/GetWarehousesListInDetail?culture={culture}&CityId={CityId}&onlyWarehouses={onlyWarehouses}&country={country}**

### Входные параметры

---

Наименование	Описание
culture	Культура; Допустимые значения (en-US, ru-RU, uk-UA). (String)
CityId	Id города. (Guid?, Значение по умолчанию null)
onlyWarehouses	Флаг – только склады (bool, значение по умолчанию false)
country	Id страны (1-Украина, 2-Россия)

---

## Выходные параметры

Представляет собой json. Коллекция объектов {id,name,address,operatingTime,Phone, EmailStorage,Latitude,Longitude,Office, CityName, IsWarehouse, RcPhoneSecurity, RcPhoneManagers, RcPhone, RcName, WarehouseForDeliveryId, WarehouseType}.

id - id представительства

name - Наименование представительства

CityId - id города

address - Адрес представительства

operatingTime - Время работы представительства

Phone - Телефонные номера представительства

EmailStorage - Email представительства

Latitude - Широта

Longitude - Долгота

Office - Признак офиса

CityName – Название города

IsWarehouse – Является ли складом

RcPhoneSecurity – Телефон службы безопасности

RcPhoneManagers – Телефон отдела менеджеров

RcPhone – Телефон регионального центра

RcName – Название регионального центра

WarehouseForDeliveryId – Id склада доставки

WarehouseType – тип склада

### Формат выходных параметров

*application/json, text/json*

### Пример:

```
{
  "status": true,
  "message": "",
  "data": [
    {
      "id": "11fb447a-4a97-e411-bf7a-000d3a200160",
      "name": "КУРАХОВО",
      "address": "ул. Березовая, 6 (ул. Чалова)",
      "operatingTime": "ПН-ПТ 9:00-18:00, СБ 9:00-17:00",
      "Phone": "(067) 557-89-25",
      "EmailStorage": "kur@delivery-auto.com.ua",
      "latitude": 37.2796560000,
      "longitude": 47.9885500000,
      "Office": null,
      "CityId": "bbb06f68-3529-e311-8b0d-00155d037960",
      "CityName": "Курахово",
      "IsWarehouse": true,
      "RcPhoneSecurity": "(044) 238-88-56",
      "RcPhoneManagers": "(067) 620-80-17",
      "RcPhone": "(044) 23-888-50",
      "RcName": "Восточный Региональный Центр - 1",
      "WarehouseForDeliveryId": null,
      "WarehouseType": null
    }
  ]
}
```

## 2. Квитанции

### 2.1 Поиск квитанции - метод *GetReceiptDetails*.

**GET api/v4/Public/GetReceiptDetails?culture={culture}&number={number}**

#### Входные параметры

---

Наименование	Описание
culture	Культура; Допустимые значения (en-US, ru-RU, uk-UA). (String)
number	Номер квитанции. (String)

---

#### Выходные параметры

Представляет собой json. Коллекция объектов {id, number, SendDate, ReceiveDate, SenderWarehouseName, RecipientWarehouseName, Discount, TotalCost, Status, Weight, Volume, Sites, PaymentStatus, Currency, InsuranceCost, InsuranceCurrency, PushStateCode, CodCost, CodCurrency, SenderPhone, ReceiverPhone, CitySendName, CityReceiveName, StatusesDecoding, SafetyDealMoneyStatus}.

id - id Квитанции

number - Номер квитанции

SendDate - Дата отправки

ReceiveDate - Дата получения

SenderWarehouseName - Склад отправки

RecipientWarehouseName - Склад получения

Discount - Сумма скидки

TotalCost - Общая стоимость квитанции

Status - Текущий статус квитанции

Weight - Общий вес груза

Volume - Объем груза

PaymentStatus - Статус оплаты

Currency – Валюта квитанции

InsuranceCost – Стоимость страхования

InsuranceCurrency – Валюта страхования

PushStateCode – Статус отправки

CodCost – Сумма наложенного платежа

CodCurrency – Валюта наложенного платежа

SenderPhone – Телефон отправителя

ReceiverPhone – Телефон получателя

CitySendName – Город отправления

CityReceiveName – Город получения

StatusesDecoding – Статус квитанции

SafetyDealMoneyStatus – Статус денежных средств по Безопасной сделке

#### Формат выходных параметров

*application/json, text/json*

#### Пример:

```
{
  "status": true,
  "message": ""
}
```

```

"data": {
  "id": "045905c9-b17b-4ccb-8e85-8ec7f5b548e2",
  "number": "0830047053",
  "SendDate": "2014-06-05T09:54:20",
  "ReceiveDate": "2014-06-07T09:54:20",
  "SenderWarehouseName": "КИЕВ-2",
  "ReceipientWarehouseName": "ЧЕРНОВЦЫ-2",
  "Discount": 0.0,
  "TotalCost": 24.500,
  "Status": 6,
  "Weight": 4.000,
  "Volume": 0.070000,
  "Sites": "1", //Кол-во мест
  "PaymentStatus": false
  "Currency": 100000000,
  "InsuranceCost": null,
  "InsuranceCurrency": null,
  "PushStateCode": null,
  "codCost":null,
  "codCurrency":null,
  "SenderPhone":null,
  "ReceiverPhone":null,
  "CitySendName":null,
  "CityReceiveName":null,
  "DeliveryType":null,
  "StatusesDecoding": "Выдана"
}
}

```

## 2.2 Расчет сроков доставки - метод *GetDateArrival*.

### **GET**

**api/v4/Public/GetDateArrival?areasSendId={areasSendId}&areasResiveld={areasResiveld}&dateSend={dateSend}&currency={currency}&warehouseSendId={warehouseSendId}&warehouseResiveld={warehouseResiveld}**

### **Входные параметры**

Наименование	Описание
areasSendId	Id Города отправления. (Guid)
areasResiveld	Id Города прибытия. (Guid)
dateSend	Дата отправления. (String)
currency	Код вылюты (int)
warehouseSendId	Id Склада отправления. (Guid?, Значение по умолчанию null)
warehouseResiveld	Id Склада прибытия. (Guid?, Значение по умолчанию null)

## Выходные параметры

Представляет собой json. Коллекция объектов {arrivalDate}.  
arrivalDate - Дата прибытия

### Формат выходных параметров

*application/json, text/json*

### Пример:

```
{
  "status": true,
  "message": "",
  "data": {
    "arrivalDate": "09.07.2014"
  }
}
```

## 3. Расчет стоимости перевозки

### 3.1 Модели данных для обмена информацией:

```
class CalculatorModel //Основная модель калькулятора
{
    string culture; //Культура
    string areasSendId; //Id города отправления
    string areasResiveId; //Id города получения
    string warehouseSendId; //Id склада отправления
    string warehouseResiveId; // Id склада получения
    string areasSendIdName; //Наименование города отправления
    string areasResiveIdName; //Наименование города получения
    string warehouseSendIdName; //Наименование склада отправления
    string warehouseResiveIdName; //Наименование склада получения
    double CashOnDeliveryValue; //Сумма наложенного платежа
    int CashOnDeliveryValuta; //Валюта наложенного платежа
    double InsuranceValue; //Страховая стоимость груза
    decimal InsuranceCost; //Стоимость страховки
    DateTime? dateSend; //Дата отправки
    DateTime? dateResive; //Дата получения
    int climbingToFloor; //Доставка на этаж
    int descentFromFloor; //Спуск с этажа
    int deliveryScheme; //Схема доставки
    List<CategoryModel> category; //Перечисления категорий груза
    List<DopUslugaClassificationModel> dopUslugaClassifier; //Перечисление доп. услуг
    decimal? categorySumma;
    decimal? allSumma; //Общая стоимость перевозки
    bool status; //Статус расчета
    bool denyIssue; //Запрет выдачи
    bool EconomDelivery; //Экономная доставка, флаг
    bool EconomPickUp; //Экономный забор, флаг
    bool IsGidrobort; //Гидроборт, флаг
    bool IsOverSize; //Негабарит, флаг
    bool isPostomat; //Запрет выдачи, флаг
    string comment; //Описание расчета
    decimal? SummaryTransportCost; //Стоимость перевозки склад склад
    decimal? SummaryDuCost; //Стоимость доп. услуг
    decimal? SummaryOformlenieCost; //Стоимость оформления
    int currency; //Валюта
    int viewType;
}

class CategoryModel //Модель категории груза
{
    string categoryId; //Id категории груза
    string categoryIdName; //Наименование категории груза
    int classification;
    int countPlace; //Количество мест
    double? helf; //Вес
    double? size; //Объем
    double? height;
    double? lenght;
    double? width;
    double? helfTarif; //Тариф за кг
    double? egTarif; //Тариф за единицу груза
    double? oformlenie; //Стоимость оформления за место
    double? oformlenieCost; //Общая стоимость оформления
    double? deliveryCost; //Стоимость перевозки
    double? documentCost;
    string comment; //Ход расчета
}
```

```
public class DopUslugaClassificationModel //Модель категории доп. услуг
{
    int classification; //Код категории
    string name; //Наименование категории
    List<DopUslugaModel> dopUsluga; //Перечисление доп. услуг
}
```

```
public class DopUslugaModel //Модель доп. услуги
{
    string uslugId; //Id доп. услуги
    string name; //Наименование доп. услуги
    decimal? cost; //Стоимость доп. услуги
    int count; //Количество услуг
    int classification;
    decimal? minWidth;
    decimal? maxWidth;
    decimal summa; //Общая стоимость доп. услуги
    string comment;
    int currency; //Валюта
}
```

### 3.2 Вывод справочника категорий доп. услуг и входящих в них доп. услуг - метод *GetDopUslugiClassification*.

#### **GET**

**[api/v4/Public/GetDopUslugiClassification?culture={culture}&currency={currency}&CitySendId={CitySendId}&CityReceiveId={CityReceiveId}&formalization={formalization}](#)**

#### **Входные параметры**

---

Наименование	Описание
culture	Культура; Допустимые значения (en-US, ru-RU, uk-UA). (String)
CitySendId	Id города отправления. (Guid)
CityReceiveId	Id города получения. (Guid)
currency	Код валюты (int)
formalization	Выводить доп. услуги для расета стоимости(false) или для оформления(true). Необязательный параметр, по умолчанию false. (bool)

---

#### **Выходные параметры**

Представляет собой json. Коллекция объектов DopUslugaClassificationModel

#### **Формат выходных параметров**

*application/json, text/json*

#### **Пример:**

```
{
  "status": true,
  "message": "",
  "data": [
    {
      "classification": 100000005,
      "name": "Упаковочные материалы",
      "dopUsluga": [
        {
          "uslugaId": "2b4247c9-be8c-e211-be60-00155d037919",
          "name": "Доупаковка МАХI",
          "cost": 9.00000000,
          "count": 0,
          "classification": 0,
          "minWidth": null,
          "maxWidth": null,
          "summa": 0.0,
          "comment": null
        }
      ],
      "currency": 100000001
    }
  ],
}
```



```

    {
      "uslugaId": "3e9cde5d-bf8c-e211-be60-00155d037919",
      "name": "Доупаковка MIDI",
      "cost": 6.00000000,
      "count": 0,
      "classification": 0,
      "minWidth": null,
      "maxWidth": null,
      "summa": 0.0,
      "comment": null
      "currency": 100000001
    }
  ]
}

```

### 3.3 Вывод справочника категорий тарифа - метод *GetTariffCategory*.

#### **GET**

**api/v4/Public/GetTariffCategory?CitySendId={CitySendId}&CityReceiveId={CityReceiveId}&WarehouseReceiveId={WarehouseReceiveId}&culture={culture}**

#### **Входные параметры**

---

Наименование	Описание
culture	Культура; Допустимые значения (en-US, ru-RU, uk-UA). (String)
CitySendId	Город отправления. (Guid)
CityReceiveId	Город получения. (Guid)
WarehouseReceiveId	Склад получения. (Guid)

---

#### **Выходные параметры**

Представляет собой json. Коллекция объектов {id, name, MaxWidth, MaxSize, MinSize, MinWidth, Length, Width, Height, RequiredWeight, RequiredSize}

- Id – id Категории тарифа
- name – Наименование категории тарифа
- MinWidth – Минимальная ширина
- MaxSize – Максимальный размер
- MinSize – Минимальный размер
- MaxWidth – Максимальная ширина
- Length – Длина
- Width – Ширина
- Height – Высота
- RequiredWeight – Допустимая ширина
- RequiredSize – Допустимый размер

## Формат выходных параметров

*application/json, text/json*

### Пример:

```
{
  "status": true,
  "message": "",
  "data": [
    {
      "MinWidth": null,
      "MaxWidth": null,
      "MinSize": null,
      "MaxSize": null,
      "Length": null,
      "Width": null,
      "Height": null,
      "RequiredWeight": null,
      "RequiredSize": null,
      "id": "00000000-0000-0000-0000-000000000000",
      "name": "Груз"
    },
    {
      "MinWidth": 0.0000,
      "MaxWidth": 2000.0000,
      "MinSize": 0.4100,
      "MaxSize": 2.4000,
      "Length": null,
      "Width": null,
      "Height": null,
      "RequiredWeight": true,
      "RequiredSize": true,
      "id": "62c7b796-e648-e211-ab75-00155d012d0d",
      "name": "\"Американка-1\" 1,0м x 1,2м x 2м"
    }
  ]
}
```

### 3.4 Вывод справочника схем доставки - метод *GetDeliveryScheme*.

#### **GET**

**[api/v4/Public/GetDeliveryScheme?CitySendId={CitySendId}&CityReceiveId={CityReceiveId}&WarehouseReceiveId={WarehouseReceiveId}&culture={culture}](#)**

#### **Входные параметры**

---

Наименование	Описание
culture	Культура; Допустимые значения (en-US, ru-RU, uk-UA). (String)
CitySendId	Id города отправления. (Guid)
CityReciveId	Id города получения. (Guid)

---

Наименование	Описание
WarehouseReceiveId	Id склада получения. (Guid)

## Выходные параметры

Представляет собой json. Коллекция объектов {id,name }  
 Id – id Схемы доставки  
 name – Наименование схемы доставки

### Формат выходных параметров

*application/json, text/json*

### Пример:

```
{
  "status": true,
  "message": "",
  "data": [
    {
      "name": "Warehouse-Warehouse",
      "id": 0
    },
    {
      "name": "Door-Door",
      "id": 1
    },
    {
      "name": "Warehouse-Door",
      "id": 2
    },
    {
      "name": "Door-Warehouse",
      "id": 3
    }
  ]
}
```

## 3.5 Расчет стоимости перевозки - метод PostReceiptCalculate.

### POST api/v4/Public/PostReceiptCalculate

#### Входные параметры

Наименование	Описание
input	Модель описывающая входные и выходные параметры калькулятора (CalculatorModel)

#### Пример входных данных:

```
{
  culture: "ru-RU", //Культура
  areasSendId: "4fc948a7-3729-e311-8b0d-00155d037960", //Город отправления
  areasResiveId: "e3ac6f68-3529-e311-8b0d-00155d037960", //Город прибытия
}
```

```

warehouseSendId: "1c828aa6-70c8-e211-9902-00155d037919", //Склад отправления
warehouseResiveId: "d908c5e1-b36b-e211-81e9-00155d012a15", //Склад прибытия
InsuranceValue: 1000000, //Страховая стоимость груза
CashOnDeliveryValue: 5000, //Стоимость наложенного платежа
dateSend: "06.06.2014", //Дата отправления
deliveryScheme: 2, //Схема доставки
category: [ //Массив категорий груза
  {
    categoryId: "00000000-0000-0000-0000-000000000000", //Id категории груза
    countPlace: 1, //Количество мест
    helf: 1, //Вес груза
    size: 1 // Объем груза
  }
],
dopUslugaClassificator: [
  {
    dopUsluga: [ //Массив доп. услуг
      {
        uslugId: "2b4247c9-be8c-e211-be60-00155d037919", //Id Доп. услуги
        count: 1 //Количество доп. услуг
      },
      {
        uslugId: "3e9cde5d-bf8c-e211-be60-00155d037919",
        count: 5
      }
    ]
  }
]
}

```

## Выходные параметры

### Формат выходных параметров

*application/json, text/json*

### Пример:

```

{
  "status": true,
  "message": "",
  "data":
  {
    "SummaryTransportCost":230, //Стоимость перевозки склад-склад
    "SummaryDuCost":99, //Стоимость доп. услуг
    "SummaryOfornlenieCost":1.5, //Стоимость оформления
    "EconomyDiscountCardValuta": 0, //Экономия на дисконтной карте
    "EconomyEconomTarifValuta": 0, //Экономия на эконом-тарифе
    "EconomyGlobalDiscountValuta": 0, //Экономия на глобальной скидке
    "EconomyIndTarifValuta": 0, //Экономия на индивидуальном тарифе
    "EconomyAktsiyaValuta": 0, //Экономия на акционной скидке от направления/склада
    "EconomySpecTarifValuta": 0, //Экономия на эконом-тарифе
    "EconomySummary": 16.5, //Суммарная экономия на квитанции
    "EconomyVesovojTarifValuta": 16.5, //Экономия на весовом тарифе
    "culture":"ru-RU", //Культура
    "areasSendId":"4fc948a7-3729-e311-8b0d-00155d037960", //Id города отправления
    "areasResiveId":"e3ac6f68-3529-e311-8b0d-00155d037960", //Id города прибытия
    "warehouseSendId":"1c828aa6-70c8-e211-9902-00155d037919", //Id склада отправления
    "warehouseResiveId":"d908c5e1-b36b-e211-81e9-00155d012a15", //Id склада прибытия
    "areasSendIdName":"Авдеевка", //Наименование города отправления
    "areasResiveIdName":"Артемовск", //Наименование города прибытия
    "warehouseSendIdName":"АВДЕЕВКА", //Наименование склада отправления
    "warehouseResiveIdName":"АРТЕМОВСК", //Наименование склада прибытия
  }
}

```

```

"InsuranceValue":1000000, //Страховая стоимость груза
"InsuranceCost":4000, //Стоимость страховки
"dateSend":"01.06.2014", //Дата отправки
"dateResive":"04.06.2014", //Дата получения
"deliveryScheme":2, //Схема перевозки
"category":[ //Массив категорий
  {
    "categoryId":"00000000-0000-0000-0000-000000000000", //Id категории груза
    "categoryIdName":"Груз", //Наименование категории груза
    "classification":0,
    "countPlace":1, //Количество мест
    "help":1, //Вес
    "size":1, //Объем
    "height":0,
    "length":0,
    "width":0,
    "helpTarif":0.98, //Тариф за вес
    "egTarif":230, //Тариф за единиц груза
    "oformlenie":1.5, //Стоимость оформления
    "oformlenieCost":1.5, //Общая стоимость оформления
    "deliveryCost":230, //Стоимость перевозки
    "documentCost":230,
    "comment":"Расчет по общему тарифу за объем\r\nСтоимость перевозки
составляет 230,00(грн./м3)*1,0000(м3) = 230,00 грн.\r\nПри скидке в 0% сумма перевозки со
скидкой составляет 230,00 грн.\r\n"
  }
],
"dopUslugaClassificator":[ //Массив категорий доп. услуг
  {
    "classification":100000005, //Код категории
    "name":"Упаковочные материалы", //Наименование категории
    "dopUsluga":[ //Массив доп. услуг внутри категории
      {
        "uslugId":"2b4247c9-be8c-e211-be60-00155d037919", //Id доп.
        "name":"Доупаковка МАХI", //Наименование доп. услуги
        "cost":9.00000000, //Стоимость доп. услуги
        "count":1, //Количество услуг
        "classification":100000005, //Код категории
        "minWidth":null,
        "maxWidth":null,
        "summa":9, //Общая стоимость доп. услуги
        "comment":null
      },
      {
        "uslugId":"3e9cde5d-bf8c-e211-be60-00155d037919",
        "name":"Доупаковка MIDI",
        "cost":6.00000000,
        "count":5,
        "classification":100000005,
        "minWidth":null,
        "maxWidth":null,
        "summa":30,
        "comment":null
      }
    ]
  }
],
  {
    "classification":100000014,
    "name":"Забор/доставка",
    "dopUsluga":[
      {
        "uslugId":"5bfb9362-04a9-e211-9619-00155d037919",
        "name":"Доставка до 500 кг",
        "cost":60.00000000,
        "count":1,
        "classification":100000014,
        "minWidth":null,
        "maxWidth":null,
        "summa":60,
        "comment":null
      }
    ]
  }
]

```

```
    }
  ]
},
"categorySumma":null,
"allSumma":4330.5, //Общая стоимость перевозки
"status":true, //Статус выполнения
"comment":"Расчет по общему тарифу за объем Стоимость перевозки составляет
230,00(грн./м3)*1,0000(м3) = 230,00 грн. При скидке в 0% сумма перевозки со скидкой
составляет 230,00 грн. В стоимость квитанции включена доп.услуга \u0027Доупаковка
МАХИ\u0027, ценой 9,00 грн./ед., количеством 1 ед. Общая сумма заказанной доп.услуги =
9,00 грн. В стоимость квитанции включена доп.услуга \u0027Доупаковка МІDІ\u0027, ценой
30,00 грн./ед., количеством 5 ед. Общая сумма заказанной доп.услуги = 150,00 грн. В
стоимость квитанции включена доп.услуга \u0027Доставка до 500 кг\u0027, ценой 60,00
грн./ед., количеством 1 ед. Общая сумма заказанной доп.услуги = 60,00 грн. Стоимость
перевозки склад-склад 230,00 грн. Стоимость перевозки склад-склад со скидкой 230,00 грн.
Стоимость доп.услуг: 99,00 грн. Стоимость страхования: 4000,00 грн. Общая стоимость
квитанции: 329,00 грн. Стоимость всех услуг округлена до 0,50 грн. ",
"viewType":0
}
```

## 4. Связь с пользователем

### 4.1 Модели данных для обмена информацией:

```
public class RateServicesModel
{
    Guid OfficeId; //Id представительства
    int WarehousePlacing; //Расположение склада
    int CargoReceiveSpeed; //Скорость приема груза
    int CargoOutputSpeed; //Скорость выдачи груза
    int DocumentsIssuanceSpeed; //Скорость оформления документов
    int DeliverySpeed; //Скорость перевозки груза
    int TarrifsRate; //Тарифы на перевозки (склад-склад)
    int CargoLoadTarrifs; //Тарифы на забор доставку груза
    int WorkersCulture; //Культура обслуживания работниками склада
    int QualityInGeneral; //Качество услуг филиала в целом
    string YourRecomendations; //Ваши пожелания и рекомендации
    string ClientNumber; //Штрих-код клиента
    string Name; //Имя клиента
    string LastName; //Фамилия клиента
    string SecondName; //Отчество клиента
    string Phone; //Номер телефона клиента
    string Email; //Электронная почта клиента
    string CompanyName; //Наименование компании
}

class PickupCargoModel
{
    string ContactName; //Контактное лицо
    string Name; //Название организации/ФИО
    string PhoneNumber; //Номер телефона
    string Email; //Электронный адрес
    string Area; //Область
    string City; //Город
    string Address; //Адрес
    string AccessMode; //Наличие пропускного режима
    int? Weight; //Вес груза
    int? Size; //Объем груза
    int? Quantity; //Количество мест
    string Date; //Дата забора
    string Time; //Желательное время
    string Note; //Примечание
    bool? IsFloor; //Спуск с этажа?
    string Floor; //Этаж
    string ToCity; //Город получения
}

class ContactsMessageModel
{
    string ReceiptNumber; //Номер квитанции
    string Name; //ФИО
    string Phone; //Телефон
    string Email; //Электронная почта
    string Subject; //Тема сообщения
    Guid? Warehouse //Id склада
    string Message; //Сообщение
    string CategoryName; //Имя категории сообщения
}
```

## 4.2 Получение новостей компании - метод GetNews.

**GET api/v4/Public/GetNews?culture={culture}&count={count}&page={page}**

### Входные параметры

---

Наименование	Описание
culture	Культура; Допустимые значения (en-US, ru-RU, uk-UA). (String)
count	Количество возвращаемых записей на странице (int)
page	Номер страницы новости. (int, Значение по умолчанию = 1)

---

### Выходные параметры

Представляет собой json. Коллекция объектов {NewsItemId, Title, ShortContent, Content, PublishDate, ImageName, ImageUrl, ImageContent, WarehousesId}.

- NewsItemId – Id новости
- Title – тема новости
- ShortContent – Краткое содержание новости
- Content – Содержание новости
- PublishDate – Дата публикации,
- ImageName – Наименование изображения
- ImageUrl – Ссылка на изображение
- ImageContent – содержание файла изображения
- WarehousesId – Id склада

### Формат выходных параметров

*application/json, text/json*

### Пример:

```
{
  "status": true,
  "message": "",
  "data": [
    {
      "NewsItemId": 111289,
      "Title": "Переезд представительства №6 в городе Киев",
      "ShortContent": "Переезд представительства №6 в городе Киев",
      "Content": "<p><strong>Уважаемые
клиенты!&nbsp;</strong></p>\r\n\r\n<p>Обратите Ваше внимание, что с 30.06.2014г.
представительство №6 в г. Киев будет расположено по новому адресу:
&nbsp;</p>\r\n\r\n<p>г. Киев: ул. Радищева, 12/16, тел: (044) 501-81-45, (067)
620-05-07.</p>\r\n\r\n<p>Заезд с переулка Радищева</p>\r\n\r\n<p><img alt=\"\"
src=\"/userfs/images/%d0%ba%d0%b8%d0%b5%d0%b2-6(1).jpg\" style=\"height:350px;
width:597px\" /></p>\r\n",
      "PublishDate": "2014-06-30T00:00:00",
      "ImageName": null,
      "ImageUrl": null,
      "ImageContent": null
    }
  ]
}
```



```

    "WarehousesId": "0b44d5c2-8ee8-e311-9747-00155d015206"
  },
  {
    "NewsItemId": 111288,
    "Title": "27.06.2014г. представительство в г. Рубежное работает до 16:00",
    "ShortContent": "27.06.2014г. представительство в г. Рубежное работает до 16:00",
    "Content": "<p><span style=\"font-family:arial,Helvetica,sans-serif; font-size:14px\">Уважаемые клиенты!</span><br />\r\n<span style=\"font-family:arial,Helvetica,sans-serif; font-size:14px\">&nbsp;</span><br />\r\n<span style=\"font-family:arial,Helvetica,sans-serif; font-size:14px\">По техническим причинам 27.06.2014г. наше представительство в г. Рубежное, будет работать до 16:00.</span></p>\r\n\r\n<p><span style=\"font-family:arial,Helvetica,sans-serif; font-size:14px\">О возобновлении работы представительства &nbsp;<br>будет сообщено дополнительно.</span></p>\r\n",
    "PublishDate": "2014-06-27T00:00:00",
    "ImageName": null,
    "ImageUrl": null,
    "ImageContent": null
    "WarehousesId": null
  },
]
}

```

### 4.3 Получение тем сообщений - метод *GetMessagesTheme*.

#### **GET api/v4/Public/GetMessagesTheme?culture={culture}**

#### **Входные параметры**

Наименование	Описание
culture	Культура; Допустимые значения (en-US, ru-RU, uk-UA). (String)

#### **Выходные параметры**

Представляет собой json. Коллекция объектов {Id, Name}.

Id – Id темы

Name – Наименование темы

#### **Формат выходных параметров**

*application/json, text/json*

#### **Пример:**

```

{
  "status": true,
  "message": "",
  "data": [
    {
      "Id": "AGREEMENT",
      "Name": "Заключение договора"
    },
    {
      "Id": "CARGO_DAMAGE",
      "Name": "Задержка, утеря, повреждение груза"
    }
  ],
}

```

```
{
  "Id": "ACCEPTANCE_DOCUMENT",
  "Name": "Получение акта выполненных работ и налоговых накладных"
}
]
```

#### 4.4 Отправка сообщения - метод *PostContactsMessage*.

##### **POST api/v4/Public/PostContactsMessage**

##### **Входные параметры**

---

Наименование	Описание
Input	Модель описывающая входные и выходные параметры отправки сообщения (ContactsMessageModel)

---

##### **Пример входных данных:**

```
{
  "ReceiptNumber": "1234567890",
  "Name": "Name",
  "Phone": "0501112233",
  "Email": "name@name.ua",
  "Subject": "COMPLIANTS",
  "Message": "message text"
}
```

##### **Выходные параметры**

##### **Формат выходных параметров**

*application/json, text/json*

##### **Пример:**

```
{
  "status": true,
  "message": ""
}
```

## 4.5 Отправка оценки работы компании - метод PostServiceRate.

### **POST api/v4/Public/PostServiceRate**

#### **Входные параметры**

---

Наименование	Описание
input	Модель описывающая входные и выходные параметры (RateServicesModel)

---

#### **Пример входных данных:**

```
{
  "OfficeId": "1c828aa6-70c8-e211-9902-00155d037919",
  "WarehosePlacing": 3,
  "CargoReceiveSpeed": 4,
  "CargoOutputSpeed": 5,
  "DocumentsIssuanceSpeed": 6,
  "DeliverySpeed": 7,
  "TarrifsRate": 8,
  "CargoLoadTarrifs": 9,
  "WorkersCulture": 10,
  "QualityInGeneral": 11,
  "YourRecomendations": "sample string 12",
  "ClientNumber": "1234567890",
  "Name": "name",
  "LastName": "last name",
  "SecondName": "second name",
  "Phone": "123456",
  "Email": "name@name.ru",
  "CompanyName": "test"
}
```

#### **Выходные параметры**

##### **Формат выходных параметров**

*application/json, text/json*

##### **Пример:**

```
{
  "status": true,
  "message": ""
}
```

## 4.6 Отправка оценки работы компании - метод PostServiceRate.

### POST api/v4/Public/PostPickUpCargo

#### Входные параметры

---

Наименование	Описание
input	Модель описывающая входные и выходные параметры (PickUpCargoModel)

---

#### Пример входных данных:

```
{
  "ContactName": "contact name",
  "Name": "name",
  "PhoneNumber": "123456",
  "Email": "name@name.ru",
  "Area": "Ar. Krim",
  "City": "donetsk",
  "Address": "test",
  "AccessMode": "1",
  "Weight": 1,
  "Size": 2,
  "Quantity": 3,
  "Date": "01.01.2014",
  "Time": "15:00",
  "Note": "sample string 11",
  "IsFloor": true,
  "Floor": "10",
  "ToCity": "qwe"
}
```

#### Выходные параметры

##### Формат выходных параметров

*application/json, text/json*

##### Пример:

```
{
  "status": true,
  "message": ""
}
```

## 5. Личный кабинет, регистрация

### 5.1 Авторизация:

#### POST api/v4/Public/PostLogin

#### Входные параметры

---

Наименование	Описание
model	Модель описывающая входные и выходные параметры (LoginModel)

---

#### Пример входных данных:

```
{
  UserName: "iv.iv.ivankov@gmail.com", //Логин пользователя
  Password: "123456", //Пароль пользователя
  RememberMe: true //Запомнить меня
};
```

#### Выходные параметры

#### Формат выходных параметров

*application/json, text/json*

#### Пример:

```
{
  "status": true,
  "message": ""
}
или
{
  "status": false,
  "code": 401,
  "message": "Неверный логин или пароль",
}
или
{
  "status": false,
  "code": 0,
  "message": ex.Message
}
```

### 5.2 Выход

#### POST api/v4/Public/PostLogoff

Метод требует авторизации

#### Входные параметры

Отсутствуют

#### Выходные параметры

## Формат выходных параметров

*application/json, text/json*

### Пример:

```
{
  "status": true,
  "message": ""
}
```

## 5.3 Получение информации о пользователе.

### GET api/v4/Public/GetUserInfo?culture={culture}

Метод требует авторизации

### Входные параметры

---

Наименование	Описание
culture	Культура; Допустимые значения (en-US, ru-RU, uk-UA). (String)

---

### Выходные параметры

Представляет собой json. Объект { AccessLevel, ClientNumber, CrmUserId, Email, Id, UserName, SmsPhoneNumber}.

- AccessLevel – Уровень доступа
- ClientNumber – Штрих-код клиента
- CrmUserId – Id клиента в CRM
- Email – Адрес электронной почты
- Id – Id клиента в базе сайта
- UserName – Имя пользователя
- SmsPhoneNumber – Номер для СМС пользователя

## Формат выходных параметров

*application/json, text/json*

### Пример:

```
{
  "status": true,
  "message": "",
  "data": {
    "AccessLevel": "Полный доступ",
    "ClientNumber": "00022558",
    "CrmUserId": "abcdefab-0123-4567-89ab-0123456789ab",
    "Email": "test@test.ts",
    "Id": "45023",
    "UserName": "!! Тестовый клиент для сайту",
    "SmsPhoneNumber": 662332658
  }
}
```

## 5.4 Получение квитанций пользователя – метод *GetUserReceipt*.

**GET api/v4/Public/GetUserReceipt?page={page}&rows={rows}&type={type}&culture={culture}&detail={detail}**

Метод требует авторизации

### Входные параметры

---

Наименование	Описание
page	Страница квитанций (int)
rows	Количество выводимых строк (int)
type	Тип квитанций (0-отправки, 1-получения) (int)
culture	Культура; Допустимые значения (en-US, ru-RU, uk-UA). (String)
detail	Надо ли возвращать информацию по доп. услугам и возможным получателям. Необязательный параметр, по умолчанию true (bool)

---

### Выходные параметры

Представляет собой json. Объект { id, number, SendDate, ReceiveDate, SenderWarehouseName, RecipientWarehouseName, TotalCost, Status, StatusesDecoding, Weight, Volume, PaymentStatus, Currency, CanChangeRecipient, LockShipping, IsPrivate, IsAllowDeny, Sender, Recipient, Payer, StatedValue, Sites, PriceWarehouseWarehouse, AuxServicesList, InsuranceCost, InsuranceValue, PossibleReceivers, codCost, codCurrency, codName, codWarehouse, isGiveMoney, SafetyDealMoneyStatus }.

- id – id квитанции,
- number – Номер квитанции,
- total – Количество страниц с квитанциями,
- SendDate – Дата отправки квитанции,
- ReceiveDate – Дата получения квитанции,
- SenderWarehouseName – Склад отправки квитанции,
- RecipientWarehouseName – Склад получения квитанции,
- TotalCost – Общая стоимость,
- PartnerNumber – Номер декларации партнёра,
- Type – тип квитанции,
- Status – Статус квитанции,
- StatusesDecoding – Текстовый статус квитанции,
- Weight – Вес,
- Volume – Объем,
- PaymentStatus – Статус оплаты,
- Currency – Валюта,

CanChangeReceipient – Можно ли менять отправителя,  
LockShipping – Запрет на выдачу,  
IsPrivate – Закрыта ли для просмотра,  
IsAllowDeny – Доступна для снятия или установления запрета на выдачу,  
Sender – Имя отправителя,  
Receipient – Имя получателя,  
Payer – Имя плательщика,  
StatedValue – Заявленная стоимость,  
Sites – Количество мест,  
PriceWarehouseWarehouse – Цена перевозки без скидки и доп. услуг,  
AuxServicesList – Список доп. услуг,  
InsuranceCost – Стоимость страховки,  
InsuranceCurrency - Валюта страховки,  
PossibleReceivers – Список возможных получателей,  
codCost – заявленная стоимость груза,  
codCurrency – валюта наложенного платежа,  
codName – имя получателя средств наложенного платежа,  
codWarehouse – склад получения средств наложенного платежа,  
isGiveMoney – выданы ли деньги получателю наложенного платежа  
SafetyDealMoneyStatus – статус денежных средств по Безопасной сделке

#### Формат выходных параметров

*application/json, text/json*

#### Пример:

```
{
  "status": true,
  "message": "",
  "data": [
    {
      "id": "c26032e6-fd57-4b8a-827b-eb93a736a80b",
      "number": "9900043094",
      "SendDate": "2015-10-28T00:00:00",
      "ReceiveDate": "2015-10-29T00:00:00",
      "SenderWarehouseName": "КИЕВ-1",
      "ReceipientWarehouseName": "КИЕВ-1",
      "Status": "8",
      "StatusesDecoding": "Зарезервирована",
      "TotalCost": 278.000,
      "PartnerNumber": "",
      "Weight": 100.000,
      "Volume": 1.000000,
      "PaymentStatus": false,
      "Currency": 100000000,
      "CanChangeReceipient": false,
      "LockShipping": false,
      "IsPrivate": 0,
      "IsAllowDeny": true,
      "Sender": "!! Иванков Иван Тест",
      "Receipient": "!! Новый Клиент Киев",
      "Payer": "!! Иванков Иван Тест",
      "StatedValue": 1000.000,
      "Sites": "2",
      "PriceWarehouseWarehouse": 262.000,
      "codCost": 1000.000,
      "codCurrency": 100000000,
    }
  ]
}
```



```
"codName": "иван иванович Иванов",
"codPhone": "0669854652",
"codWarehouse": "ДНІПРО-1Центр З/Д (Лівий бер.)",
"AuxServicesList": [
  {
    "Id": "3d39b06b-c80b-4ca8-be5c-4ad3e59439b5",
    "ReceiptId": null,
    "Name": "Оформление багажа",
    "Count": 2,
    "Summ": 6.00000
  },
  {
    "Id": "0b23e486-9362-4615-b12c-63fd813ffdef",
    "ReceiptId": null,
    "Name": "Возврат паллет",
    "Count": 1,
    "Summ": 10.00000
  }
],
"InsuranceCost": 0.0,
"InsuranceCurrency": 100000000,
"PossibleReceivers": [
  {
    "Id": "35ba4253-b1fe-4cdc-b5da-820a7e891e3f",
    "ReceiptId": null,
    "Name": " Козлов, ФОП"
  }
],
"PushStateCode": 0
}
]
```

## 6. Оформление квитанции

### 6.1 Получение доступа через API ключ и выбор формата данных.

Для использования этих методов, требующих авторизации через API, необходимо дополнительно передавать api ключ. Пользователю выдается пара значений публичный и секретный api ключи. Для авторизации пользователю необходимо передать в заголовке запроса в параметре "HMACAuthorization" публичный ключ, текущее время сервера и хеш код, сформированный путем шифрования по алгоритму HmacSHA1. Примеры использования приведены ниже.

#### Пример на Javascript.

```
var apiKey = 'CDBFE2D5-BF02-4C0D-B7D6-5CF277761C50';
var apiSecretKey = '6c131f01b99dfac3529d0cd68b1d6649';

var getHMAC = function (key, timestamp) {
    var hash = CryptoJS.HmacSHA1(key + timestamp, apiSecretKey);
    return hash.toString();
};

var data = {
    "egs": [
        {
            "Id": "f6ee49fa-3e29-e311-8b0d-00155d037960",
            "PartnerNumber": "123456"
        }
    ]
};

$.ajax({
    url: 'http://www.delivery-auto.com/api/v4/Public/PostDeactivateEg',
    type: "POST",
    data: data,
    dataType: 'json',
    beforeSend: function (request) {
        request.setRequestHeader('HMACAuthorization', 'amx ' + apiKey + ':' + timestamp + ':' +
getHMAC(apiKey, timestamp));
    },
    success: function (data) {
        debugger;
        if (data.status == true) {
            debugger;
        }
    },
    error: errorMessageFunc
});
```

#### Пример на C#.

```
public string getHMAC(string publicKey, TimeSpan timestamp, string secretKey) {
    string message = publicKey + timestamp.Milliseconds.ToString();
    System.Text.ASCIIEncoding encoding = new System.Text.ASCIIEncoding();
    HMACSHA1 hmacsha1 = new HMACSHA1(encoding.GetBytes(secretKey));
    byte[] hashmessage = hmacsha1.ComputeHash(encoding.GetBytes(message));
    return ByteToString(hashmessage);
}

public static string ByteToString(byte[] buff)
```

```

{
    string sbinary = "";
    for (int i = 0; i < buff.Length; i++)
    {
        sbinary += buff[i].ToString("X2"); // hex format
    }
    return (sbinary);
}

public ActionResult TestJSApi()
{
    string dataString = @"{
        ""egs"": [
            {
                ""Id"": ""f6ee49fa-3e29-e311-8b0d-00155d037960"",
                ""PartnerNumber"": ""123456""
            }
        ]
    }";

    var apiKey = "CDBFE2D5-BF02-4C0D-B7D6-5CF277761C50";
    var apiSecretKey = "6c131f01b99dfac3529d0cd68b1d6649";

    DateTime myDate1 = new DateTime(1970, 1, 9, 0, 0, 00);
    DateTime myDate2 = DateTime.Now;
    var timestamp = myDate2.Subtract(myDate1);
    var HMAC = getHMAC(apiKey, timestamp, apiSecretKey);

    var request = (HttpRequest)WebRequest.Create("http://www.delivery-
auto.com/api/v4/Public/PostDeactivateEg");

    var data = System.Text.Encoding.UTF8.GetBytes(dataString);

    request.Method = "POST";
    request.ContentType = "text/json";
    request.ContentLength = data.Length;
    request.Headers["HMACAuthorization"] = string.Format("amx {0}:{1}:{2}", apiKey,
timestamp.Milliseconds.ToString(), HMAC);

    using (var stream = request.GetRequestStream())
    {
        stream.Write(data, 0, data.Length);
    }

    var response = (HttpWebResponse)request.GetResponse();

    var responseString = new StreamReader(response.GetResponseStream()).ReadToEnd();
}

```

По умолчанию данные выводятся формате json. Чтобы изменить тип входных и выходных данных на xml необходимо в адресную строку передать параметр type=xml. Например //api/v3/Public/GetClientCards?type=xml.

## 6.2 Получение списка платежных карт клиента.

### GET api/v4/Public/GetClientCards

Метод требует авторизации через API

### Входные параметры

Отсутствуют

### Выходные параметры

Представляет собой список объектов. Объект { id, name }.

id – Id карты

name – Номер карты

### Формат выходных параметров

*application/json, text/json*

### Пример:

**В формате json.**

```
{
  "data": [
    {
      "id": "3dafcedb-904b-4210-ae46-2af2acd385ef",
      "name": "5104*****8490"
    },
    {
      "id": "c05e1eae-80e2-4380-a2f2-5ddbd34a07ce",
      "name": "5211*****9950"
    },
    {
      "id": "42b137f9-eb86-424b-b2ed-60b5b4659299",
      "name": "5168*****8737"
    }
  ],
  "status": true,
  "message": ""
}
```

**В формате xml.**

```
<ApiResponse>
  <status>true</status>
  <message/>
  <data>
    <DirectoryItem>
      <id>3dafcedb-904b-4210-ae46-2af2acd385ef</id>
      <name>5104*****8490</name>
    </DirectoryItem>
    <DirectoryItem>
      <id>c05e1eae-80e2-4380-a2f2-5ddbd34a07ce</id>
      <name>5211*****9950</name>
    </DirectoryItem>
    <DirectoryItem>
      <id>42b137f9-eb86-424b-b2ed-60b5b4659299</id>
      <name>5168*****8737</name>
    </DirectoryItem>
  </data>
</ApiResponse>
```

## 6.3 Получение списка расчетных счетов клиента.

### **GET api/v4/Public/GetClientInvoices**

Метод требует авторизации через API

#### **Входные параметры**

Отсутствуют

#### **Выходные параметры**

Представляет собой список объектов. Объект { id, name }.

id – Id карты

name – Номер карты

#### **Формат выходных параметров**

*application/json, text/json*

#### **Пример:**

**В формате json.**

```
{
  "data": [
    {
      "id": "f38964d0-f2a5-e411-b119-000d3a200160",
      "name": "77777777777777777777"
    }
  ],
  "status": true,
  "message": ""
}
```

**В формате xml.**

```
<ApiResponse>
  <status>true</status>
  <message/>
  <data>
    <DirectoryItem>
      <id>f38964d0-f2a5-e411-b119-000d3a200160</id>
      <name>77777777777777777777</name>
    </DirectoryItem>
  </data>
</ApiResponse>
```

## 6.4 Получение списка категорий отправляемого груза.

### GET api/v4/Public/GetCargoCategory?culture={culture}

#### Входные параметры

---

Наименование	Описание
culture	Культура; Допустимые значения (en-US, ru-RU, uk-UA). (String)

---

#### Выходные параметры

Представляет собой список объектов. Объект { id, name }.

id – Id карты

name – Номер карты

#### Формат выходных параметров

*application/json, text/json*

#### Пример:

##### **В формате json.**

```
{
  "status": true,
  "message": "",
  "data": [
    {
      "id": "0656bab4-e62c-e411-bd10-000d3a200936",
      "name": "Автоаксессуары"
    },
    {
      "id": "f306d03b-9e36-e311-8b0d-00155d037960",
      "name": "Автозапчасти, автомототехника"
    }
  ]
}
```

##### **В формате xml.**

```
<ApiResponse>
  <status>true</status>
  <message/>
  <data>
    <DirectoryItem>
      <id>0656bab4-e62c-e411-bd10-000d3a200936</id>
      <name>Автоаксессуары</name>
    </DirectoryItem>
    <DirectoryItem>
      <id>f306d03b-9e36-e311-8b0d-00155d037960</id>
      <name>Автозапчасти, автомототехника</name>
    </DirectoryItem>
  </data>
</ApiResponse>
```

## 6.5 Оформление квитанции - метод PostCreateReceipts.

### POST api/v4/Public/PostCreateReceipts

Метод требует авторизации через API

#### Входные параметры

---

Наименование	Описание
input	Модель описывающая входные и выходные параметры квитанции (RegistrationReceiptsModel)

---

#### Пример входных данных:

```
var data = {
  "culture": "ru-RU", //Культура
  "flSave": "true", //Флаг сохранения
  "debugMode": "false", //Флаг режима дебага
  "receiptsList": [
    {
      "areasSendId": "f6ee49fa-3e29-e311-8b0d-00155d037960", //Город отправки
      "areasResiveId": "ebc7639a-db2a-e311-8b0d-00155d037960", //Город получения
      "warehouseSendId": "6b3b6d45-b249-e211-ab75-00155d012d0d", //Склад отправки
      "warehouseResiveId": "ab3b6d45-b249-e211-ab75-00155d012d0d", //Склад получения

      "dateSend": "2015-06-24T00:00:00", //Дата отправки
      "deliveryScheme": 1, //Схема доставки 0- Склад-склад, 1- Двери двери, 2-
Склад-двери, 3- Двери-склад
      "receiverName": "Иванков Иван Иванович", //Получатель; для физ.лица - Ф.И.О.
      "receiverPhone": "0500000000", //Телефон получателя
      "receiverType": false, //false - физ. лицо, true - юр. лицо
      "currency": 100000000, //Валюта квитанции
      "InsuranceValue": 10000.0, //Застрахованная стоимость груза
      "senderId": "cdbfe2d5-bf02-4c0d-b7d6-5cf277761c50", //Id отправителя
      "payerInsuranceId": "1aa70d22-1209-e511-b3b5-000d3a200160", //плательщик
страховки
      "payerId": "1aa70d22-1209-e511-b3b5-000d3a200160", //если плательщиком надо
передать третье лицо

      "payerType": 1, //Тип плательщика 0-отправитель, 1-получатель
      "paymentType": 0, //Тип оплаты 1-безнал., 0-нал.
      "paymentTypeInsuranse": 0, //Тип оплаты 1-безнал., 0-нал.

      "deliveryAddress": "Науки, 5", //Адрес доставки
      "deliveryContactName": "Дмитрий", //Контактное лицо для доставки
      "deliveryContactPhone": "0500000000", //Телефон дл доставки
      "DeliveryComment": "Комментарий при доставке", //Комментарий при доставке
      "ReturnDocuments": true, //Флаг возврат документа при доставке
      "climbingToFloor": 4, //Подъем на этаж
      "EconomDelivery": false, //Эконом доставка
      "IsOverSize": false, //Негабарит, для Доставки груза
      "IsGidrobort": false, //Гидроборт, для Доставки груза
      "EconomPickUp": false, //Эконом забор, для Забора груза
      "ExpressPickUp": false, //Экспресс забор, для Забора груза
      "CustomsCost": 1000.0, //Таможенна стоимость груза
      "CustomsCurrency": 100000000, //Валюта для таможи
      "CustomsDocuments": true, //Флаг наличия документов для таможи
      "CustomsDescriptions": "Опиание для таможи", // Опиание для таможи

      "cashOnDeliveryType": 0, //Тип наложенного платежа 0-плетажна карта, 1-
расчетый счет, 2-наличными, 3-Безопасная сделка
    }
  ]
}
```

```
"CashOnDeliveryValuta": 100000000, //Валюта наложенного платежа
"CashOnDeliveryValue": 1000.0, //Сумма наложенного платежа
"CashOnDeliveryCardId": "B08ACA89-B6C8-4014-ABF8-3EA61B18E5DA", //Id платежной
карты
    "CashOnDeliveryPayerAccountId": "1aa70d22-1209-e511-b3b5-000d3a200160"
//плательщик наложенного платежа

    "CashOnDeliveryRasschSchetId": "00000000-0000-0000-0000-000000000000", //Id
расчетного счета (при Безопасной сделке поле обязательно для заполнения)
    "CashOnDeliveryDescription": "описание", //Описание наложенного платежа

    "CashOnDeliveryWarehouseId": "6b3b6d45-b249-e211-ab75-00155d012d0d", //Склад
выплаты наложенного платежа

    "CashOnDeliverySenderFullName": "Иванов Иван Иванович", //Ф.И.О. получателя
наложенного платежа
    "CashOnDeliverySenderPhone": "0501234567", //Телефон получателя наложенного
платежа
    "CashOnDeliveryReceiverFullName": "Петров Петр Петрович", //Ф.И.О. плательщика
наложенного платежа
    "CashOnDeliveryReceiverPhone": "0671234567", //Телефон плательщика наложенного
платежа

    "pickUpDate": "2015-06-30T00:00:00", //Дата забора
    "pickUpContactName": "Василий", //контактное лицо для забора
    "pickUpContactPhone": "0500000000", //Контактный телефон для забора
    "pickUpAddressId": "a5eaf714-fb60-e411-b421-000d3a200936", //Адрес забора
    "descentFromFloor": 4, //спуск с этажа

    "category": [
        {
            "categoryId": "00000000-0000-0000-0000-000000000000", //Категория тарифа
            "cargoCategoryId": "0307d03b-9e36-e311-8b0d-00155d037960", //Категория
отправляемого груза

            "countPlace": 10, //Количество мест
            "helf": 100.0, //Вес
            "size": 2.0, //Объем
            "isEconom": true, //Экономная, но более долгая доставка
            "PartnerNumber": "123456" //Номер декларации партнера
        },
        {
            "categoryId": "00000000-0000-0000-0000-000000000000",
            "cargoCategoryId": "0f07d03b-9e36-e311-8b0d-00155d037960",
            "countPlace": 10,
            "helf": null,
            "size": null,
            "isEconom": true,
            "PartnerNumber": "123457"
        }
    ]
},
{
    "areasSendId": "f6ee49fa-3e29-e311-8b0d-00155d037960", //Город отправки
    "areasResiveId": "4577d856-322b-e311-8b0d-00155d037960", //Город получения
    "warehouseSendId": "6b3b6d45-b249-e211-ab75-00155d012d0d", //Склад отправки
    "warehouseResiveId": "efbecb4b-da49-e211-9515-00155d012d0d", //Склад получения

    "dateSend": "2015-06-24T00:00:00", //Дата отправки
    "deliveryScheme": 3, //Схема доставки 0- Склад-склад, 1- Двери двери, 2-
Склад-двери, 3- Двери-склад
    "possibleResiverReceipt_1": "07b98959-52ab-40a0-9ce7-ab7ee678d809", //Id
получателя
    "possibleResiverReceipt_2": "07b98959-52ab-40a0-9ce7-ab7ee678d809", //Id
возможного получателя
}
```



```

    "possibleResiverReceipt_3": "07b98959-52ab-40a0-9ce7-ab7ee678d809", //Id
возможного получателя
    "possibleResiverReceipt_4": "07b98959-52ab-40a0-9ce7-ab7ee678d809", //Id
возможного получателя

    "currency": 100000000, //Валюта квитанции
    "InsuranceValue": 10000.0, //Застрахованная стоимость груза

    "payerType": 0, //Тип плательщика 0-отправитель, 1-получатель
    "paymentType": 0, //Тип оплаты 1-безнал., 0-нал.
    "paymentTypeInsuranse": 0, //Тип оплаты 1-безнал., 0-нал.

    "pickUpDate": "2015-06-30T00:00:00", //Дата забора
    "pickUpContactName": "Василий", //контактное лицо для забора
    "pickUpContactPhone": "0500000000", //Контактный телефон для забора
    "pickUpAddressId": "a5eaf714-fb60-e411-b421-000d3a200936", //Адрес забора
    "descentFromFloor": 4, //спуск с этажа

    "category": [
        {
            "categoryId": "00000000-0000-0000-0000-000000000000", //Категория тарифа
            "cargoCategoryId": "0307d03b-9e36-e311-8b0d-00155d037960", //Категория
отправляемого груза
            "countPlace": 1, //Количество мест
            "helf": 20.0, //Вес
            "size": 1.0, //Объем
            "isEconom": true,
            "PartnerNumber": "1234567"
        }
    ]
}

```

Отправителем является клиент, который авторизовался перед вызовом метода. SenderId может принять значение вашего Id (публичный ключ) или Id вашей родительской или дочерней организации.

У данного метода обязательными полями являются все, связанные с отправителем, получателем, плательщиком; дата, город, склад отправления и получения; категория груза. Если какие-то из необходимых полей не будут заполнены, то метод вернёт предупреждение и квитанция создана не будет.

При отладке модулей создания квитанций рекомендуется использовать **debugMode**.

Для определения получателя квитанции необходимо или указать его Id или создать нового. В случае если указывается уже существующий то его Id передается в параметре **possibleResiverReceipt\_1**. Для создания нового получателя используется три параметра **receiverName**, **receiverType** и **receiverPhone**.

Аналогична ситуация с полями адреса доставки и адреса забора. Создать новый адрес забора и получения можно передав их в текстовом формате в поля **pickUpAdress** и **deliveryAddress** соответственно. Или передав их **Id** в поля **pickUpAdressId** и **deliveryAddressId**.

В качестве массива можно передавать квитанции объединенные одним заказом на забор, то есть с совпадающим городом отправления, адресом и датой отправления.

## Выходные параметры

### Формат выходных параметров

*application/json, text/json*

#### Пример:

```
{
  "status": true,
  "message": [],
  "receipts": [
    {
      "Id": "f5a947f6-adcf-49e8-be46-a49d69621ae2",
      "Number": "9900000000", //Номер квитанции
      "TotalCost": 97.0, //Стоимость перевозки
      "InsuranceCost": 7.0, //Стоимость страхования
      "ComissionGM": 30.0, //Комиссия наложенного платежа наличными: % от суммы + 10 грн
      за услугу наложенного платежа
      "Comment": "", //Комментарий
      "egs": [//Массив единиц груза
        {
          "Id": "3fed9940-b094-4236-a3b0-728a83123eca",
          "PartnerNumber": null,
          "Number": "9900000000002002151"
        }
      ]
    }
  ]
}
```

#### В формате xml.

```
<RegistrationReceiptsOutputModel>
  <status>true</status>
  <message/>
  <receipts>
    <ReceiptsOutputModel>
      <Id>0656bab4-e62c-e411-bd10-000d3a200936</id>
      <PartnerNumber>123456</PartnerNumber>
    </ReceiptsOutputModel>
    <ReceiptsOutputModel>
      <Id>f306d03b-9e36-e311-8b0d-00155d037960</id>
      <PartnerNumber>123457</PartnerNumber>
    </ReceiptsOutputModel>
  </receipts>
</RegistrationReceiptsOutputModel>
```

## 6.6 Деактивация единиц груза - метод PostDeactivateEg.

### POST api/v4/Public/PostDeactivateEg

#### Входные параметры

---

Наименование	Описание
input	Модель описывающая входные и выходные параметры (ReceiptsOutputModel)

---

#### Пример входных данных:

```
{
  "egs": [
    {
      "Id": "f6ee49fa-3e29-e311-8b0d-00155d037960",
      "PartnerNumber": "123456"
    }
  ]
}
```

## Выходные параметры

### Формат выходных параметров

[application/json](#), [text/json](#)

### Пример:

**В формате json.**

```
{
  "status": true,
  "message": "",
  "data": []
}
```

**В формате xml.**

```
<ApiResult>
  <status>true</status>
  <message/>
  <data />
</ApiResult>
```

## 6.7 Получение документов в PDF - метод *GetPdfDocument*.

Метод требует авторизации через API ключ

**GET api/v4/Public/GetPdfDocument?number={number}&type={type}**

### Входные параметры

---

Наименование	Описание
number	Номер квитанции
type	Тип документа: 0 - Печать квитанции, 1 - Печать стикеров единиц груза на годексе, 2 - Печать стикеров единиц груза на одном листе, 4 – Печать на одном листе 95x95 нескольких квитанций (номера указываются через точку с запятой, н-р, ?number=9900112233;9900223344&type=4)

---

## Выходные параметры

### Формат выходных параметров

[application/json](#), [text/json](#)

## Пример:

### **В формате json.**

```
{
  "status": true,
  "message": "",
  "file": "EKJRLJFSDWEKNLVSJDIFJS"
}
```

### **В формате xml.**

```
<ApiResponseFile>
  <status>true</status>
  <message/>
  <file>EKJRLJFSDWEKNLVSJDIFJS</file>
</ApiResponseFile>
```

\* Значение поля file возвращает код зашифрованный в base64

## 6.8 Получение списка отправителей (дочерних или родительских организаций клиента) - метод *GetSenderList*.

### **GET api/v4/Public/GetSenderList**

Метод требует авторизации через API

### **Входные параметры**

Отсутствуют.

### **Выходные параметры**

#### **Формат выходных параметров**

*application/json, text/json*

## Пример:

### **В формате json.**

```
{
  "data": [
    {
      "id": "cdbfe2d5-bf02-4c0d-b7d6-5cf277761c50",
      "name": "!! Иванков Иван Тест",
      "cityId": "16617DF3-A42A-E311-8B0D-00155D037960",
      "cityName": "Київ"
    },
    {
      "id": "c11d0fff-b75d-e411-b4c0-000d3a200936",
      "name": "!! Иванков Получатель 99 Авдеевка",
      "cityId": "4FC948A7-3729-E311-8B0D-00155D037960",
      "cityName": "Авдіївка"
    }
  ],
  "status": true,
  "message": ""
}
```

### **В формате xml.**

```
<ApiResponseFile>
  <status>true</status>
```

```

<message/>
data>
  <DirectoryItem>
    <id>cdbfe2d5-bf02-4c0d-b7d6-5cf277761c50</id>
    <name>!! Иванков Иван Тест</name>
    <cityId>16617DF3-A42A-E311-8B0D-00155D037960</cityId >
    <cityName>Київ</cityName >
  </DirectoryItem>
  <DirectoryItem>
    <id>c11d0fff-b75d-e411-b4c0-000d3a200936</id>
    <name>!! Иванков Получатель 99 Авдеевка </name>
    <cityId>4FC948A7-3729-E311-8B0D-00155D037960</cityId >
    <cityName>Авдіївка</cityName >
  </DirectoryItem>
</data>
</ApiResponseFile>

```

## 6.9 Получение схем доставки - метод *GetDeliveryScheme*.

### **GET**

**api/Public/GetDeliveryScheme?CitySendId={CitySendId}&CityReceiveId={CityReceiveId}&WarehouseReceiveId={WarehouseReceiveId}&culture={culture}**

### Входные параметры

Наименование	Описание
CitySendId	Id города отправления. (Guid)
CityReceiveId	Id города получения. (Guid)
WarehouseReceiveId	Id склада получения. (Guid)
culture	Культура; Допустимые значения (en-US, ru-RU, uk-UA). (String)

### Выходные параметры

#### Формат выходных параметров

*application/json, text/json*

#### Пример:

**В формате json.**

```

{
  "data": [
    {
      "id": "1",
      "name": "Двери-двери"
    },
    {

```

```

        "id": "2",
        "name": "Склад-двери"
    }
],
"status": true,
"message": ""
}

```

**В формате xml.**

```

<ApiResult>
  <status>true</status>
  <message/>
  <data>
    <DirectoryItem>
      <id>1</id>
      <name>Двери-двери</name>
    </DirectoryItem>
    <DirectoryItem>
      <id>2</id>
      <name>Склад-двери</name>
    </DirectoryItem>
  </data>
</ApiResult>

```

## 6.10 Получение стоимости страхования - метод *GetInsuranceCost*.

**GET api/v4/Public/GetInsuranceCost?CitySendId={CitySendId}&CityReceiveId={CityReceiveId}&WarehouseSendId={WarehouseSendId}&WarehouseReceiveId={WarehouseReceiveId}&InsuranceValue={InsuranceValue}&InsuranceCurrency={InsuranceCurrency}**

### Входные параметры

Наименование	Описание
CitySendId	Id города отправления. (Guid)
CityReceiveId	Id города получения. (Guid)
WarehouseSendId	Id склада отправления. (Guid)
WarehouseReceiveId	Id склада получения. (Guid)
InsuranceValue	Заявленная стоимость груза. (double)
InsuranceCurrency	Валюта оплаты страхования. (int)

## Выходные параметры

### Формат выходных параметров

*application/json, text/json*

#### Пример:

##### В формате json.

```
{
  "Value": 40.0,
  "MinValue": 10000.0,
  "status": true,
  "message": null
}
```

##### В формате xml.

```
<ApiResponseInsurance>
  <status>true</status>
  <Value>40</Value>
  <MinValue>10000</MinValue>
</ApiResponseInsurance>
```

Value - стоимость страхования

MinValue - минимальная стоимость заявленного груза для данного направления

## 6.11 Получение категорий груза - метод *GetCargoCategory*.

**GET api/v4/Public/GetCargoCategory?TariffCategoryId={TariffCategoryId}&culture={culture}**

### Входные параметры

---

Наименование	Описание
TariffCategoryId	Id категории тарифа. (Guid)
culture	Культура; Допустимые значения (en-US, ru-RU, uk-UA). (String)

---

## Выходные параметры

### Формат выходных параметров

*application/json, text/json*

#### Пример:

##### В формате json.

```
{
  "data": [
    {
      "id": "0f07d03b-9e36-e311-8b0d-00155d037960",
      "name": "Документы"
    }
  ],
}
```

```
"status": true,  
"message": ""  
}
```

**В формате xml.**

```
<ApiResult>  
  <status>true</status>  
  <message/>  
  <data>  
    <DirectoryItem>  
      <id>0f07d03b-9e36-e311-8b0d-00155d037960</id>  
      <name>Документы</name>  
    </DirectoryItem>  
  </data>  
</ApiResult>
```

## 6.12 Получение доступных валют - метод *GetCurrency*.

### **GET**

**api/v4/Public/GetCurrency?CitySendId={CitySendId}&CityReceiveId={CityReceiveId}&PayerType={PayerType}&PayerId={PayerId}&culture={culture}**

### **Входные параметры**

---

Наименование	Описание
CitySendId	Id города отправления. (Guid)
CityReceiveId	Id города получения. (Guid)
PayerType	Тип плательщика 0 – отправитель, 1 – получатель, 2 – третье лицо (int).
PayerId	Id плательщика. (Guid?)
culture	Культура; Допустимые значения (en-US, ru-RU, uk-UA). (String)

---

### **Выходные параметры**

#### **Формат выходных параметров**

*application/json, text/json*

#### **Пример:**

**В формате json.**

```
{  
  "data": [  
    {  
      "id": "1000000000",  
      "name": "Гривня"  
    }  
  ]  
}
```



```
    }
  ],
  "status": true,
  "message": ""
}
```

**В формате xml.**

```
<ApiResponse>
  <status>true</status>
  <message/>
  <data>
    <DirectoryItem>
      <id>100000000</id>
      <name>Гривня</name>
    </DirectoryItem>
  </data>
</ApiResponse>
```

### 6.13 Получение списка плательщиков - метод *GetPayer*.

#### **GET**

**api/v4/Public/GetPayer?CitySendId={CitySendId}&CityReceiveId={CityReceiveId}&ClientSenderId={ClientSenderId}&ClientReceiverId={ClientReceiverId}&PayerType={PayerType}**

#### **Входные параметры**

---

Наименование	Описание
CitySendId	Id города отправления. (Guid)
CityReceiveId	Id города получения. (Guid)
ClientSenderId	Id клиента отправителя. (Guid)
ClientReceiverId	Id клиента получателя. (Guid?)
PayerType	Тип плательщика 0 – отправитель, 1 – получатель, 2 – третье лицо (int?).

---

#### **Выходные параметры**

##### **Формат выходных параметров**

*application/json, text/json*

##### **Пример:**

**В формате json.**

```
{
  "data": [
    {
```

```

        "id":"c11d0fff-b75d-e411-b4c0-000d3a200936",
        "name":"!! Иванков Получатель 99 Авдеевка"
    },
    {
        "id":"cdbfe2d5-bf02-4c0d-b7d6-5cf277761c50",
        "name":"!! Иванков Иван Тест"
    }
],
"status":true,
"message":""
}

```

**В формате xml.**

```

<ApiResult>
  <status>true</status>
  <message/>
  <data>
    <DirectoryItem>
      <id>c11d0fff-b75d-e411-b4c0-000d3a200936</id>
      <name>!! Иванков Получатель 99 Авдеевка</name>
    </DirectoryItem>
    <DirectoryItem>
      <id>cdbfe2d5-bf02-4c0d-b7d6-5cf277761c50</id>
      <name>!! Иванков Иван Тест</name>
    </DirectoryItem>
  </data>
</ApiResult>

```

## 6.14 Получение справочника категорий тарифа - метод *GetTariffCategory*.

**GET api/v4/Public/GetTariffCategory?CitySendId={CitySendId}&CityReceiveId={CityReceiveId}&WarehouseReceiveId={WarehouseReceiveId}&culture={culture}**

Метод требует авторизации через API ключ

### Входные параметры

Наименование	Описание
CitySendId	Id города отправления. (Guid)
CityReceiveId	Id города получения. (Guid)
WarehouseReceiveId	Id склада получения. (Guid)
culture	Культура; Допустимые значения (en-US, ru-RU, uk-UA). (String)

## Выходные параметры

### Формат выходных параметров

*application/json, text/json*

#### Пример:

**В формате json.**

```
{
  "data": [
    {
      "MinWidth": 0.0000, //Минимальный вес
      "MaxWidth": 2000.0000, //Максимальный вес
      "MinSize": 0.4100, //Минимальный объем
      "MaxSize": 2.4000, //Максимальный объем
      "Length": null, //Максимальная длина
      "Width": null, //Максимальная ширина
      "Height": null, //Максимальная высота
      "RequiredWeight": true, //Требуется ввод веса
      "RequiredSize": true, //Требуется ввод объема
      "id": "62c7b796-e648-e211-ab75-00155d012d0d",
      "name": "\"Американка-1\" 1,0м x 1,2м x 2м"
    }
  ],
  "status": true,
  "message": ""
}
```

**В формате xml.**

```
<ApiResult>
  <status>true</status>
  <message/>
  <data>
    <DirectoryItem>
      <id>00000000-0000-0000-0000-000000000000</id>
      <name>Груз</name>
    </DirectoryItem>
    <DirectoryItem>
      <id>62c7b796-e648-e211-ab75-00155d012d0d</id>
      <name>"Американка-1" 1,0м x 1,2м x 2м</name>
    </DirectoryItem>
    <DirectoryItem>
      <id>e9e885b1-e648-e211-ab75-00155d012d0d</id>
      <name>"Американка-2" 1,2м x1,2м x 2 м</name>
    </DirectoryItem>
  </data>
</ApiResult>
```

## 6.15 Получение адресов клиента - метод *GetClientAddress*.

**GET api/v4/Public/GetClientAddress?CityId={CityId}&ClientId={ClientId}**

### Входные параметры

---

Наименование	Описание
CityId	Id города. (Guid)
ClientId	Id клиента. (Guid)

---

### Выходные параметры

#### Формат выходных параметров

*application/json, text/json*

#### Пример:

##### **В формате json.**

```
{
  "data": [
    {
      "id": "a5eaf714-fb60-e411-b421-000d3a200936",
      "name": "!! для забора Киев 2 дом 2 кв "
    },
    {
      "id": "8a8522be-0064-e411-b2e7-000d3a200936",
      "name": "ленина дом 2 кв "
    }
  ],
  "status": true,
  "message": ""
}
```

##### **В формате xml.**

```
<ApiResponse>
  <status>true</status>
  <message/>
  <data>
    <DirectoryItem>
      <id>a5eaf714-fb60-e411-b421-000d3a200936</id>
      <name>!! для забора Киев 2 дом 2 кв</name>
    </DirectoryItem>
    <DirectoryItem>
      <id>8a8522be-0064-e411-b2e7-000d3a200936</id>
      <name>ленина дом 2 кв</name>
    </DirectoryItem>
  </data>
</ApiResponse>
```

## 6.16 Получение возможных получателей клиента- метод *GetPosibleReciver.*

### **GET**

**api/v4/Public/GetPosibleReciver?CityReceiveId={CityReceiveId}&ClientSenderId={ClientSenderId}**

Метод требует авторизации через API ключ

### **Входные параметры**

---

Наименование	Описание
CityReceiveId	Id города получения. (Guid)
ClientSenderId	Id клиента отправителя. (Guid)

---

### **Выходные параметры**

**Формат выходных параметров**  
*application/json, text/json*

**Пример:**

**В формате json.**

```
{
  "data": [
    {
      "id": null,
      "name": ""
    },
    {
      "id": "f26e1d30-ea7e-4d87-967b-7be00002b51d",
      "name": " Козлов, ФОП"
    },
    {
      "id": "ce30c74a-7253-400e-b98f-f7a19a811731",
      "name": " Скаско Олена Миколаївна"
    }
  ],
  "status": true,
  "message": ""
}
```

**В формате xml.**

```
<ApiResult>
  <status>true</status>
  <message/>
  <data>
    <DirectoryItem>
      <name/>
```

```
</DirectoryItem>
<DirectoryItem>
  <id>f26e1d30-ea7e-4d87-967b-7be00002b51d</id>
  <name> Козлов, ФОР</name>
</DirectoryItem>
<DirectoryItem>
  <id>ce30c74a-7253-400e-b98f-f7a19a811731</id>
  <name> Скаско Олена Николаївна</name>
</DirectoryItem>
</data>
</ApiResponse>
```

## 6.17 Получение вида оплаты клиента - метод *GetClientPaymentType*.

**GET api/v4/Public/GetClientPaymentType?ClientId={ClientId}**

### Входные параметры

---

Наименование	Описание
ClientId	Id клиента. (Guid)

---

### Выходные параметры

#### Формат выходных параметров

*application/json, text/json*

#### Пример:

**В формате json.**

```
{
  "data": true,
  "status": true,
  "message": ""
}
```

**В формате xml.**

```
<ApiResponseBool>
  <status>true</status>
  <message/>
  <data>true</data>
</ApiResponseBool>
```

data - вида оплаты клиента (bool). True = наличный, False = безналичный

## 6.18 Получение списка складов/потомаатов - метод *GetWarehousesListByCity*.

### **GET**

**api/v4/Public/GetWarehousesListByCity?CityId={CityId}&DirectionType={DirectionType}&culture={culture}**

### **Входные параметры**

---

Наименование	Описание
CityId	Id города. (Guid)
DirectionType	Тип направления 0 - склады отправления, 1 – склады получения. (int)
culture	Культура; Допустимые значения (en-US, ru-RU, uk-UA). (String)

---

### **Выходные параметры**

Представляет собой json. Коллекция объектов {id,name,address,operatingTime,Phone, EmailStorage,Latitude,Longitude,Office, CityId, CityName, IsWarehouse, RcPhoneSecurity, RcPhoneManagers, RcPhone, RcName, WarehouseForDeliveryId}.

id - id представительства

name - Наименование представительства

address - Адрес представительства

operatingTime - Время работы представительства

Phone - Телефонные номера представительства

EmailStorage - Email представительства

latitude - Широта

longitude - Долгота

Office - Признак офиса

CityId – id города

CityName – название города

IsWarehouse – является ли складом

IsCashOnDelivery – наличие кассы наложенного платежа

RcPhoneSecurity – телефон отдела СБ

RcPhoneManagers – телефон отдела менеджеров

RcPhone – телефон регионального центра

RcName – наименование регионального центра

WarehouseForDeliveryId – склад для доставки

## Формат выходных параметров

*application/json, text/json*

### Пример:

#### **В формате json.**

```
{
  "data": [
    {
      "id": "efbecb4b-da49-e211-9515-00155d012d0d",
      "name": "ХАРЬКОВ-1",
      "CityId": "4577d856-322b-e311-8b0d-00155d037960",
      "latitude": 36.4295970000,
      "longitude": 49.9120550000,
      "address": "ул. Роганская, 147",
      "operatingTime": "ПН-ПТ: 8:00-20:00, СБ: 9:00-17:00",
      "Phone": "(050) 471-15-64, (067) 620-72-00, (067) 620-72-27, (057) 754-60-17
(ф), Логисты - (067) 620-08-33, (050) 477-30-30",
      "EmailStorage": "xl@delivery-auto.com.ua",
      "IsWarehouse": true,
      "IsCashOnDelivery": true,
      "CityName": "Харьков",
      "RcPhoneSecurity": "(067) 621-00-81",
      "RcPhoneManagers": "(057) 754-60-16 (ф), (057) 719-70-80",
      "RcPhone": "(057) 731-41-34, (067) 628-47-14",
      "RcName": "Восточный Региональный Центр - 3"
    }
  ],
  "status": true,
  "message": ""
}
```

#### **В формате xml.**

```
<ApiResult>
  <status>true</status>
  <message/>
  <data>
    <DirectoryItem>
      <id>bdf546c-cb16-e211-89ed-00155d053b5d</id>
      <name>КИЕВ-1</name>
      <CityId>4577d856-322b-e311-8b0d-00155d037960</CityId>
      <latitude>36.4295970000</latitude>
      <longitude>49.9120550000</longitude>
      <address>ул. Роганская, 147</address>
      <operatingTime>ПН-ПТ: 8:00-20:00, СБ: 9:00-17:00</operatingTime>
      <Phone>(050) 471-15-64, (067) 620-72-00</Phone>
      <EmailStorage>xl@delivery-auto.com.ua</EmailStorage>
      <IsWarehouse>true</IsWarehouse>
      <IsCashOnDelivery>true</IsCashOnDelivery>
      <CityName>Харьков</CityName>
      <RcPhoneSecurity>(067) 621-00-81</RcPhoneSecurity>
      <RcPhoneManagers>(057) 754-60-16 (ф)</RcPhoneManagers>
      <RcPhone>(057) 731-41-34, (067) 628-47-14</RcPhone>
      <RcName>Восточный Региональный Центр - 3</RcName>
    </DirectoryItem>
  </data>
</ApiResult>
```



## 6.19 Получение полной информации по квитанции - метод *GetFullReceiptInformation*.

**GET api/v4/Public/GetFullReceiptInformation?culture={culture}&number={number}**

### Входные параметры

---

Наименование	Описание
culture	Культура; Допустимые значения (en-US, ru-RU, uk-UA). (String)
number	Номер квитанции (String)

---

### Выходные параметры

#### Формат выходных параметров

*application/json, text/json*

#### Пример:

**В формате json.**

```
{
  "status": true,
  "message": "",
  "data": {
    "areasSendId": "ebc7639a-db2a-e311-8b0d-00155d037960",
    "areasSend": "Симферополь",
    "areasReceiveId": "16617df3-a42a-e311-8b0d-00155d037960",
    "areasReceive": "Киев",
    "warehouseSendId": "ab3b6d45-b249-e211-ab75-00155d012d0d",
    "warehouseSend": "СИМФЕРОПОЛЬ-1",
    "warehouseReceiveId": "0c51680d-e932-e211-a357-00155d053b5d",
    "warehouseReceive": "КИЕВ-11",
    "deliveryScheme": 1,
    "number": "9900034260",
    "sender": "!! Иванков Иван Тест",
    "senderId": "cdbfe2d5-bf02-4c0d-b7d6-5cf277761c50",
    "receiver": "!! Новый Клиент Киев",
    "receiverId": "e05db89f-5166-e411-b380-000d3a200936",
    "payer": "!! Новый Клиент Киев",
    "payerId": "e05db89f-5166-e411-b380-000d3a200936",
    "paymentType": 0,
    "dateSend": "2015-09-14T00:00:00",
    "dateReceive": "2015-09-28T00:00:00",
    "state": 8,
    "Currency": 100000000,
    "partnerNumber": "2323",
    "paymentStatus": false,
    "paymentDate": null,
    "lockShipping": false,
    "totalCountPlace": "2",
    "totalWeight": 1.010,
    "totalSize": 1.010000,
  }
}
```

```
"warehouseWarehouseAmount": 2982.000,
"discountAmount": 0.0,
"lossesDiscountAmount": 0.00,
"totalAmount": 7682.500,
"insuranceValue": 10000.000,
"SafetyDealMoneyStatus": "Повернені",
"duArray": [
  {
    "uslugaId": "f8e3c68a-100d-e411-8c28-00155d015206",
    "name": "Доставка до 500 кг ОБЛ",
    "count": 1,
    "cost": 68.50000,
    "addressId": "b366f3a5-5166-e411-b380-000d3a200936",
    "address": "ул. Щетинина дом 2 кв ",
    "type": 2
  },
  {
    "uslugaId": "fc45b052-ebda-44bf-b186-603b18046448",
    "name": "Подъем на этаж до 5 кг",
    "count": 4,
    "cost": 0.00000,
    "addressId": null,
    "address": null,
    "type": 4
  },
  {
    "uslugaId": "8105e479-351f-e511-9ab9-000d3a200160",
    "name": "Доупаковка MIDI // темная",
    "count": 2,
    "cost": 18.00000,
    "addressId": null,
    "address": null,
    "type": null
  },
  {
    "uslugaId": "c35cb0ca-01a0-e411-b119-000d3a200160",
    "name": "Наложный платеж",
    "count": 1,
    "cost": 10.00000,
    "addressId": null,
    "address": null,
    "type": null
  },
  {
    "uslugaId": "989299b0-120d-e411-8c28-00155d015206",
    "name": "Забор груза до 500 кг Крым",
    "count": 1,
    "cost": 180.00000,
    "addressId": "35f85f41-2681-e411-bf77-000d3a200160",
    "address": "ул. Щорса дом 33 кв ",
    "type": 1
  },
  {
    "uslugaId": "449b53aa-1215-43ac-a37f-a4ed48a0d953",
    "name": "Спуск с этажа до 5кг",
    "count": 5,
    "cost": 0.00000,
    "addressId": null,
    "address": null,
    "type": 3
  },
  {
    "uslugaId": "8321e03d-7f4a-e211-8b7c-00155d012d0d",
    "name": "Оформление багажа",
```

```
"count": 2,
"cost": 6.00000,
"addressId": null,
"address": null,
"type": null
},
{
  "uslugaId": "f5f718c5-7c4a-e211-b373-00155d012d0d",
  "name": "Возврат паллет",
  "count": 1,
  "cost": 10.00000,
  "addressId": null,
  "address": null,
  "type": null
},
{
  "uslugaId": "7c4fbf45-dc0c-e411-8c28-00155d015206",
  "name": "Возврат документов ОБЛ",
  "count": 1,
  "cost": 40.00000,
  "addressId": null,
  "address": null,
  "type": 5
},
{
  "uslugaId": "804a3931-0bbd-e411-87fc-000d3a200160",
  "name": "Таможенные услуги (юрлицо)",
  "count": 1,
  "cost": 4368.00000,
  "addressId": null,
  "address": null,
  "type": 6
}
],
"possibleReceiverArray": [],
"receiptsArray": [
  {
    "number": "9900034261",
    "state": 8,
    "receiptType": 10,
    "paymentType": 1,
    "paymentStatus": false,
    "paymentDate": null,
    "currency": 100000000,
    "payerId": "e05db89f-5166-e411-b380-000d3a200936",
    "payer": "!! Новый Клиент Киев",
    "totalAmount": 10000.000,
    "clientCardId": "2548d998-7b98-4bd9-902a-671729662fc2",
    "clientCard": "5168*****6956",
    "codSender": "Тестов Тест Тестович",
    "codSenderPhone": "0501112233",
    "isGiveMoney": false,
    "codWarehouse": "Киев-14",
    "codCity": "КИЕВ"
  },
  {
    "number": "9900034262",
    "state": 8,
    "receiptType": 6,
    "paymentType": 1,
    "paymentStatus": false,
    "paymentDate": null,
    "currency": 100000000,
    "payerId": "cdbfe2d5-bf02-4c0d-b7d6-5cf277761c50",
```

```

        "payer": "!! Иванов Иван Тест",
        "totalAmount": 40.000,
        "clientCardId": null,
        "clientCard": "",
        "codSender": "",
        "codSenderPhone": "",
        "isGiveMoney": false,
        "codWarehouse": "",
        "codCity": ""
    }
],
"egArray": [
    {
        "cargoGregoryId": "0656bab4-e62c-e411-bd10-000d3a200936",
        "cargoGregory": "Автоаксессуары",
        "count": 1,
        "weight": 1.000,
        "size": 1.000000,
        "isEconomy": false,
        "cost": 2952.000
    },
    {
        "cargoGregoryId": "0f07d03b-9e36-e311-8b0d-00155d037960",
        "cargoGregory": "Документы",
        "count": 1,
        "weight": 0.010,
        "size": 0.010000,
        "isEconomy": false,
        "cost": 30.000
    }
]
}
}
}

```

## 6.20 Создание адреса или получателя - метод PostCreateAddressOrClient.

### POST api/v4/Public/PostCreateAddressOrClient

#### Входные параметры

Наименование	Описание
input	Модель описывающая входные параметры (ClientModel)

#### Пример входных данных:

```

{ //Модель для создания возможного получателя
  "AccountId": "",
  "ClientType": "false", //Тип клиента false-физ. лицо true-юр. лицо
  "Name": "", //Наименование организации (юр. лицо)
  "SecondName": "Тестовый", //Фамилия
  "FirstName": "Клиент", //Имя
  "LastName": "ДляСайта", //Отчество
  "CityId": "16617df3-a42a-e311-8b0d-00155d037960", //Id города
  "Egrpo": "", //ОКПО клиента
  "PhoneNumber": "0509996665", //Телефон Клиента
  "Street": "ул. Щорса", //Улица
  "House": "17", //Дом
  "Appartament": "3", //Квартира
  "senderId": "c11d0fff-b75d-e411-b4c0-000d3a200936" //Id отправителя
}

```

```
{ //Модель для создания адреса
  "AccountId": "1541a45b-1a56-e511-89e5-000d3a200160", //Id клиента
  "CityId": "16617df3-a42a-e311-8b0d-00155d037960", //Id города
  "Street": "ул. Васильковская", //Улица
  "House": "17", //Дом
  "Appartement": "3", //Квартира
  "senderId": "c11d0fff-b75d-e411-b4c0-000d3a200936" //Id отправителя
}
```

## Выходные параметры

### Формат выходных параметров

*application/json, text/json*

### Пример:

**В формате json.**

```
{
  "status": true,
  "data": {
    "address": {
      "Id": 98240,
      "Street": "ул. Щорса",
      "House": "17",
      "Appartement": "3",
      "AccountId": "1541a45b-1a56-e511-89e5-000d3a200160",
      "CityId": "16617df3-a42a-e311-8b0d-00155d037960",
      "Territoria": null,
      "StateCode": 0,
      "EntityId": "b280f4a4-1a56-e511-89e5-000d3a200160",
      "Index": null
    },
    "account": {
      "Id": 278147,
      "AccountId": "1541a45b-1a56-e511-89e5-000d3a200160",
      "ClientType": false,
      "Name": "!! Тестовый Клиент Для сайта 102",
      "FirstName": "Клиент",
      "LastName": "Для сайта 102",
      "SecondName": "!! Тестовый",
      "PaymentType": true,
      "CityId": "16617df3-a42a-e311-8b0d-00155d037960",
      "Egrpо": "",
      "Inn": "",
      "Kpp": "",
      "OwnershipCode": 100000066,
      "PhoneNumber": "0509996665",
      "SmsPhoneNumber": "0509996665",
      "ParentAccountId": null,
      "ParentAccountName": "",
      "StateCode": 0,
      "CountryCode": "38",
      "MasterId": null
    }
  }
}
```

## 6.21 Получение информации по стикеру квитанции - метод GetStickers.

Метод требует авторизации через API ключ

**GET api/v4/Public/GetStickers?number={number}**

### Входные параметры

---

Наименование	Описание
number	Номер квитанции

---

### Выходные параметры

Формат выходных параметров

*application/json, text/json*

Пример:

**В формате json.**

```
{
  "data": [
    {
      "barcode": "9900000126001011160",
      "categoryName": "Детские товары",
      "receiptNumber": "9900000126",
      "receiver": "!! Иванков Получатель Александрия",
      "dateSend": "2014-11-08T11:53:01",
      "dateReceive": "2014-11-12T00:00:00",
      "warehouseSend": "МАРИУПОЛЬ-1",
      "warehouseReceive": "ОЛЕКСАНДРИЯ",
      "totalPlaces": "11",
      "rang": 1,
      "econom": false,
      "delivery": true,
      "postomat": false
    },
    ...
  ],
  "status": true,
  "message": ""
}
```

**В формате xml.**

```
<ApiResponseStickers>
  <status>true</status>
  <message/>
  <data>
    <Sticker>
      <barcode>9900000126001011160</barcode>
      <categoryName>Детские товары</categoryName>
      <receiptNumber>9900000126</receiptNumber>
      <receiver>!! Иванков Получатель Александрия</receiver>
      <dateSend>2014-11-08T11:53:01</dateSend>
      <dateReceive>2014-11-12T00:00:00</dateReceive>
      <warehouseSend>МАРИУПОЛЬ-1</warehouseSend>
      <warehouseReceive>ОЛЕКСАНДРИЯ</warehouseReceive>
      <totalPlaces>11</totalPlaces>
      <rang>1</rang>
    </Sticker>
  </data>
</ApiResponseStickers>
```

```
        <econom>false</econom>
        <delivery>true</delivery>
        <postomat>false</postomat>
    </Sticker>
</data>
</ApiResponseStickers>
```

## 6.22 Объединение квитанций в один заказ на забор - метод *PostAddReceiptIntoPickUpRequest*.

### **POST api/v4/Public/PostAddReceiptIntoPickUpRequest**

Метод требует авторизации через API

#### **Входные параметры**

---

Наименование	Описание
input	Модель описывающая входные и выходные параметры квитанции (ExptractReceiptModel)

---

#### **Пример входных данных:**

```
var data = {
    "pickUpContactName": "Василий", //контактное лицо для забора
    "pickUpContactPhone": "0500000000", //Контактный телефон для забора
    "pickUpAddress": "ул. Науки, 50", //Адрес забора если он не создан
    "descentFromFloor": 4, //спуск с этажа
    "pickUpAddressId": "a5eaf714-fb60-e411-b421-000d3a200936", //Адрес забора если он создан ранее
    "receiptNumberList": ["9900070818", "9900070822", "9900070823"] //Массив номеров квитанций
}
```

В качестве массива можно передавать квитанции объединенные одним городом отправления, датой отправления.

#### **Выходные параметры**

##### **Формат выходных параметров**

*application/json, text/json*

##### **Пример:**

###### **В формате json.**

```
{
    "status": true,
    "message": "",
    "data": []
}
```

###### **В формате xml.**

```
<ApiResponse>
    <status>true</status>
    <message/>
    <data />
</ApiResponse>
```

## 6.23 Получение реестра отправки - метод *SendingRegistr*.

### GET ru-RU/SharedForms/SendingRegister

#### Входные параметры

---

Наименование	Описание
id	Номер заявки на забор груза

---

#### Выходные параметры

##### Формат выходных параметров

*Файл с расширением html*

##### Пример ссылки:

<http://www.delivery-auto.com.ua/ru-RU/SharedForms/SendingRegister?id=35334>

##### Выходной файл:

Register\_35334.html



## 7. Операции с логами квитанции

### 7.1 Получение логов по квитанции.

**GET /api/v4/Public/GetUnidersalLogsByReceiptNumber?number={number}&culture={culture}**

Метод требует авторизации

#### Входные параметры

---

Наименование	Описание
number	Номер квитанции (string)
culture	Культура; Допустимые значения (en-US, ru-RU, uk-UA). (String)

---

#### Выходные параметры

Представляет собой список объектов json. Объект {id, CreatedOn, WarehouseId, WarehouseName, OperationCode, OperationName }.

CreatedOn – Дата создания

WarehouseId – Id склада

WarehouseName – Наименование склада

OperationCode – Код операции

OperationName – Наименование операции

#### Формат выходных параметров

*application/json, text/json*

#### Пример:

```
{
  "status": true,
  "message": "",
  "calculatorModel": [
    {
      "CreatedOn": "2014-12-22T13:12:39.393",
      "WarehouseId": "bdf546c-cb16-e211-89ed-00155d053b5d",
      "WarehouseName": "КИЕВ-1",
      "OperationCode": 100000002,
      "OperationName": "Оформление квитанции на складе",
      "Number": null
    },
    {
      "CreatedOn": "2014-12-23T08:11:20.593",
      "WarehouseId": "265757bd-3ed7-e211-bafa-00155d037932",
      "WarehouseName": "РОВНО-ТРАНЗИТ",
      "OperationCode": 100000061,
      "OperationName": "Выгрузка груза из машины",
      "Number": null
    }
  ]
}
```

# Дополнительные справочники

## Справочник статусов квитанции

Код	Значение
0	Выдана. Груз выдан получателю в полном объеме.
1	Частично выдана. Груз выдан получателю не в полном объеме.
2	Оформлена. Груз прибыл на склад отправления.
3	Утилизирована
4	Продана
5	Отменена
6	В пути. Груз находится в дороге.
7	Доступна к выдаче. Груз прибыл на склад получения.
8	Зарезервирована
9	Передресована на другой склад
10	На выгрузке на складе
11	Груз прибыл в город назначения
12	Готовится к доставке курьером
13	Доставляется курьером

## Справочник валют

Код	Значение
100000000	Гривна
100000001	Рубль

## Справочник кодов операции

Код операции	Описание
100000002	Оформление квитанции на складе
100000013	Изменение даты прибытия квитанции
100000016	Переадресация квитанции между складами одного города
100000018	Погрузка груза в автомобиль
100000026	Изменение даты получения квитанции при доставке
100000059	Погрузка груза в автомобиль
100000060	Выгрузка груза из машины
100000061	Выгрузка груза из машины
100000062	Погрузка груза в автомобиль
100000070	Отмена выдачи квитанции
100000072	Выдача груза клиенту
100000079	Погрузка груза в автомобиль
100000082	Переадресация квитанции между городами
100000111	Оформление квитанции на складе
100000115	Оформление возвратной квитанции на складе
100000122	Отмена выдачи квитанции
100000125	Выдача груза клиенту
100000132	Выгрузка груза из машины