

API по работе интернет магазинов с Delivery.

Версия 3.4.

Оглавление

Введение	3
1. Представительства	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>Получение списка складов - метод GetWarehousesListByCity.</i>	8
1.6 <i>Поиск ближайших представительств - метод GetFindWarehouses.</i>	10
1.7 <i>Получение списка представительств с подробной информацией по городу - метод GetWarehousesListInDetail.</i>	12
2. Квитанции	14
2.1 <i>Поиск квитанции - метод GetReceiptDetails.</i>	14
2.2 <i>Расчет сроков доставки - метод GetDateArrival.</i>	15
3. Расчет стоимости перевозки	17
3.1 <i>Модели данных для обмена информацией</i>	17
3.2 <i>Вывод справочника категорий доп. услуг и входящих в них доп. услуг - метод GetDopUslugiClassification.</i>	18
3.3 <i>Вывод справочника категорий тарифа - метод GetTariffCategory.</i>	19
3.4 <i>Получение категорий груза - метод GetCargoCategory.</i>	21
3.5 <i>Вывод справочника схем доставки - метод GetDeliveryScheme.</i>	22
3.6 <i>Расчет стоимости перевозки - метод PostReceiptCalculate.</i>	23
3.7 <i>Получение стоимости страхования - метод GetInsuranceCost.</i>	26
4. Связь с пользователем	27
4.1 <i>Модели данных для обмена информацией:</i>	27
4.2 <i>Получение новостей компании - метод GetNews.</i>	28
4.3 <i>Получение тем сообщений - метод GetMessagesTheme.</i>	29
4.4 <i>Отправка оценки работы компании - метод PostServiceRate.</i>	30
4.5 <i>Отправка заказа автомобиля - метод PostPickUpCargo.</i>	31
5. Личный кабинет, регистрация	32
5.1 <i>Авторизация – метод PostLogin</i>	32

5.2	<i>Выход – метод PostLogoff</i>	32
5.3	<i>Получение информации о пользователе – метод GetUserInfo.</i>	33
5.4	<i>Получение квитанций пользователя – метод GetUserReceipt.</i>	34
6.	Оформление квитанции	38
6.1	<i>Получение доступа через API ключ и выбор формата данных.</i>	38
6.2	<i>Получение списка платежных карт клиента – метод GetClientCards.</i>	39
6.3	<i>Получение списка расчетных счетов клиента – метод GetClientInvoices.</i>	40
6.4	<i>Оформление квитанции - метод PostCreateReceipts.</i>	41
6.5	<i>Деактивация единиц груза - метод PostDeactivateEg.</i>	50
6.6	<i>Получение документов в PDF - метод GetPdfDocument.</i>	51
6.7	<i>Получение списка отправителей (дочерних или родительских организаций клиента) - метод GetSenderList.</i>	51
6.8	<i>Получение доступных валют - метод GetCurrency.</i>	53
6.9	<i>Получение списка плательщиков - метод GetPayer.</i>	54
6.10	<i>Получение адресов клиента - метод GetClientAddress.</i>	56
6.11	<i>Получение возможных получателей клиента- метод GetPosibleReciver.</i>	57
6.12	<i>Получение вида оплаты клиента - метод GetClientPaymentType.</i>	59
6.13	<i>Получение полной информации по квитанции - метод GetFullReceiptInformation.</i>	59
6.14	<i>Создание адреса или получателя - метод PostCreateAddressOrClient.</i>	64
6.15	<i>Получение информации по стикеру квитанции - метод GetStickers.</i>	66
6.16	<i>Объединение квитанций в один заказ на забор – метод PostAddReceiptIntoPickUpRequest</i>	67
6.17	<i>Получение реестра отправки - метод SendingRegister.</i>	68
7.	Операции с логами квитанции	69
7.1	<i>Получение логов по квитанции – метод GetUnidersalLogsByReceiptNumber</i>	69
8.	Дополнительные справочники	70
8.1	<i>Справочник статусов квитанции</i>	70
8.2	<i>Справочник валют</i>	70
8.3	<i>Справочник кодов операции</i>	70
8.4	<i>Справочник типов квитанций</i>	71

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	Тип данных	Значение по умолчанию	Описание
culture	String	uk-UA	Культура; Допустимые значения (en-US, ru-RU, uk-UA).
country	Integer?	null	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}®ionId={regionId}&country={country}&cityName={cityName}

Входные параметры

Наименование	Тип данных	Значение по умолчанию	Описание
culture	String	uk-UA	Культура; Допустимые значения (en-US, ru-RU, uk-UA).
fl_all	Boolean	false	Флаг, позволяющий показывать все города, где есть возможность оказания услуг компании
regionId	Integer?	null	Id области
country	Integer?	null	Id страны (1-Украина, 2-Россия, null - все)
cityName	String	null	Название города на языке, выбранном в параметре "culture"

Выходные параметры

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

regionId – id 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": "Александрийский"
    },
    {
      "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	Тип данных	Значение по умолчанию	Описание
culture	String	uk-UA	Культура; Допустимые значения (en-US, ru-RU, uk-UA)

Наименование	Тип данных	Значение по умолчанию	Описание
includeRegionalCenters	Boolean	false	Выводить офисы?
CityId	Guid?	null	Id города.
RegionId	Guid?	null	Id области.
country	Integer?	null	Id страны (1-Украина, 2-Россия, null - все)

Выходные параметры

Представляет собой json. Коллекция объектов {id, name, address, Latitude, Longitude, CityId, LatitudeCorrect, LongitudeCorrect, IsCashOnDelivery, CenterPickUpDelivery}.

id – id склада

name – Наименование склада

address – Адрес склада

Latitude – Широта (некорректна, так как перепутана местами с долготой)

Longitude – Долгота (некорректна, так как перепутана местами с широтой)

CityId – Id города

LatitudeCorrect – Корректная широта

LongitudeCorrect – Корректная долгота

IsCashOnDelivery – есть ли на складе услуга наложенного платежа

CenterPickUpDelivery – есть ли на складе центр забора доставок

Формат выходных параметров

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

Пример:

```
{
  "status": true,
  "message": "",
  "data": [
    {
      "id": "1c828aa6-70c8-e211-9902-00155d037919",
      "name": "АВДЕЕВКА",
      "address": "пр. Индустриальный, 1",
      "Latitude": 37.7081000000,
      "Longitude": 48.1624700000,
      "CityId": "4fc948a7-3729-e311-8b0d-00155d037960",
      "LatitudeCorrect": 48.1624700000,
      "LongitudeCorrect": 37.7081000000,
      "IsCashOnDelivery": true,
      "CenterPickUpDelivery": false
    },
    {
      "id": "e627c8fd-d549-e211-9515-00155d012d0d",
      "name": "АЛЕКСАНДРИЯ",
      "address": "ул. Дибровы, 16",
      "Latitude": 33.1150260000,
```

```

        "Longitude": 48.6727020000,
        "CityId": "1e8e7257-a82a-e311-8b0d-00155d037960",
        "LatitudeCorrect": 48.6727020000,
        "LongitudeCorrect": 33.1150260000,
        "IsCashOnDelivery": true,
        "CenterPickUpDelivery": false
    }
]
}

```

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

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

Входные параметры

Наименование	Тип данных	Значение по умолчанию	Описание
culture	String	uk-UA	Культура; Допустимые значения (en-US, ru-RU, uk-UA).
WarehousesId	Guid	*	Id представительства.

Выходные параметры

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

- id – id представительства
- name – Наименование представительства
- address – Адрес представительства
- operatingTime – Время работы представительства
- Phone – Телефонные номера представительства
- EmailStorage – Email представительства
- Latitude – Широта (некорректна так как перепутана местами с долготой)
- Longitude – Долгота (некорректна так как перепутана местами с широтой)
- LatitudeCorrect – Корректная широта
- LongitudeCorrect – Корректная 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
- IsCashOnDelivery – есть ли на складе услуга наложенного платежа

WarehouseType – тип склада

CenterPickUpDelivery – есть ли на складе центр забора доставок

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,
    "latitudeCorrect": 48.6727020000,
    "longitudeCorrect": 33.1150260000,
    "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,
    "IsCashOnDelivery": true,
    "WarehouseType": 3,
    "CenterPickUpDelivery": false
  }
}
```

1.5 Получение списка складов - метод *GetWarehousesListByCity*.

GET

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

Входные параметры

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

Выходные параметры

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

id – id представительства
name – Наименование представительства
address – Адрес представительства
operatingTime – Время работы представительства
Phone – Телефонные номера представительства
EmailStorage – Email представительства
Latitude – Широта (некорректна так как перепутана местами с долготой)
Longitude – Долгота (некорректна так как перепутана местами с широтой)
LatitudeCorrect – Корректная широта
LongitudeCorrect – Корректная 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
IsCashOnDelivery – есть ли на складе услуга наложенного платежа
WarehouseType – тип склада
CenterPickUpDelivery – есть ли на складе центр забора доставок

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,
    "latitudeCorrect": 48.6727020000,
    "longitudeCorrect": 33.1150260000,
    "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,
    "IsCashOnDelivery": true,
    "WarehouseType": 3,
    "CenterPickUpDelivery": false
  }
}

```

1.6 Поиск ближайших представительств - метод *GetFindWarehouses*.

GET

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

Входные параметры

Наименование	Тип данных	Значение по умолчанию	Описание
culture	String	uk-UA	Культура; Допустимые значения (en-US, ru-RU, uk-UA).
Longitude	Double	*	Долгота точки, от которой производится поиск
Latitude	Double	*	Широта точки, от которой производится поиск
count	Integer	1	Количество возвращаемых представительств в порядке увеличения расстояния от указанной точки.
includeRegionalCenters	Boolean	false	Выводить офисы.
CityId	Integer?	null	Id города
type	Integer?	null	Тип склада (0-склад, 3-склад с наложенным платежом наличным)
country	Integer?	null	Id страны (1-Украина, 2-Россия, null - все)

Выходные параметры

Представляет собой json. Коллекция объектов {id, name, distance, latitude, longitude, LatitudeCorrect, LongitudeCorrect, cityName, address, IsWarehouse, Phone, working_time, WarehouseType, IsRegionalCentre}.
 id – id представительства

name – Наименование представительства
distance – Расстояние до указанной точки
latitude – Широта (некорректна так как перепутана местами с долготой)
longitude – Долгота (некорректна так как перепутана местами с широтой)
latitudeCorrect – Корректная широта
longitudeCorrect – Корректная долгота
cityName – название города
address – адрес склада
IsWarehouse – является ли складом
phone – телефон склада
working_time – рабочее время
WarehouseType – 0-склад, 3-склад с наложенным платежом наличным
IsRegionalCentre – является ли региональным центром

Формат выходных параметров

application/json, text/json

Пример:

```
{
  "status": true,
  "message": "",
  "data": [
    {
      "id": "11fb447a-4a97-e411-bf7a-000d3a200160",
      "name": "КУРАХОВО",
      "distance": 20.7,
      "longitude": 47.9902660000,
      "latitude": 37.2778887000,
      "longitudeCorrect": 37.2778887000,
      "latitudeCorrect": 47.9902660000,
      "cityName": "Курахово",
      "address": "ул. Грушева, 9/1",
      "IsWarehouse": true,
      "phone": "0675578925",
      "working_time": "ПН-ПТ 9:00-18:00, СБ 9:00-17:00",
      "WarehouseType": 3,
      "IsRegionalCentre": false
    }
  ]
}
```

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

Выходные параметры

Представляет собой json. Коллекция объектов {id, name, address, operatingTime, Phone, EmailStorage, latitude, longitude, latitudeCorrect, longitudeCorrect, Office, CityId, CityName, IsWarehouse, RcPhoneSecurity, RcPhoneManagers, RcPhone, RcName, WarehouseForDeliveryId, IsCashOnDelivery, WarehouseType, CenterPickUpDelivery}.

id – id представительства

name – Наименование представительства

address – Адрес представительства

operatingTime – Время работы представительства

Phone – Телефонные номера представительства

EmailStorage – Email представительства

latitude – Широта (некорректна так как перепутана местами с долготой)

longitude – Долгота (некорректна так как перепутана местами с широтой)

latitudeCorrect – Корректная широта

longitudeCorrect – Корректная долгота

Office – Признак офиса

CityId – id города

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

IsCashOnDelivery – есть ли на складе услуга наложенного платежа

WarehouseType – тип склада

CenterPickUpDelivery – есть ли на складе центр забора доставок

Формат выходных параметров

application/json, text/json

Пример:

```
{
  "status": true,
  "message": "",
  "data": [
    {
      "id": "1c828aa6-70c8-e211-9902-00155d037919",
      "name": "АВДЕЕВКА",
      "address": "пр. Индустриальный, 1",
      "operatingTime": "ПН-ПТ 9:00-18:00, СБ 9:00-15:00",
      "Phone": "0676959349",
      "EmailStorage": "avdiyivka@delivery-auto.com.ua",
      "latitude": 37.7081000000,
      "longitude": 48.1624700000,
      "latitudeCorrect": 48.1624700000,
      "longitudeCorrect": 37.7081000000,
      "Office": null,
      "CityId": "4fc948a7-3729-e311-8b0d-00155d037960",
      "CityName": "Авдеевка",
      "IsWarehouse": true,
      "RcPhoneSecurity": "(044) 238-88-56",
      "RcPhoneManagers": "(067) 627-67-58",
      "RcPhone": null,
      "RcName": "Восточный Региональный Центр - 1",
      "WarehouseForDeliveryId": null,
      "IsCashOnDelivery": true,
      "WarehouseType": 3,
      "CenterPickUpDelivery": false
    }
  ]
}
```

2. Квитанции

2.1 Поиск квитанции - метод *GetReceiptDetails*.

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

Входные параметры

Наименование	Тип данных	Значение по умолчанию	Описание
culture	String	uk-UA	Культура; Допустимые значения (en-US, ru-RU, uk-UA).
number	String	*	Номер квитанции.

Выходные параметры

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

id - id Квитанции

number – Номер квитанции

SendDate – Дата отправки

ReceiveDate – Дата получения

CreatedData – Дата создания

SenderWarehouseName – Склад отправки

RecipientWarehouseName – Склад получения

Discount – Сумма скидки

TotalCost – Общая стоимость квитанции

Status – Текущий статус квитанции

Weight – Общий вес груза

Volume – Объем груза

Sites – Количество мест

PaymentStatus – Статус оплаты

Currency – Валюта квитанции

InsuranceCost – Стоимость страхования

InsuranceValue – Заявленная стоимость

InsuranceCurrency – Валюта страхования

PushStateCode – Статус отправки

CodCost – Сумма наложенного платежа

CodCurrency – Валюта наложенного платежа

Type – Тип квитанции (см. справочник 8.4)

SenderPhone – Телефон отправителя

ReceiverPhone – Телефон получателя

CitySendName – Город отправления

CityReceiveName – Город получения

DeliveryType – Схема доставки (см. пункт 3.4)

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",
    "CreatedDate": "2014-06-05T06:52:50",
    "SenderWarehouseName": "КИЕВ-02",
    "ReceipientWarehouseName": "ЧЕРНОВЦЫ-2",
    "Discount": 0.0,
    "TotalCost": 24.500,
    "Status": 0,
    "Weight": 4.0,
    "Volume": 0.07,
    "Sites": "1",
    "cargoCategory": "",
    "PaymentStatus": true,
    "Currency": 100000000,
    "InsuranceCost": null,
    "InsuranceValue": null,
    "InsuranceCurrency": null,
    "PushStateCode": null,
    "codCost": null,
    "codCurrency": null,
    "Type": 2,
    "DateArrivalExpress": null,
    "SenderPhone": null,
    "ReceiverPhone": null,
    "CitySendName": null,
    "CityReceiveName": null,
    "DeliveryType": null,
    "StatusesDecoding": "Выдана",
    "codSender": null,
    "SafetyDealMoneyStatus": null,
    "InsuranceInfo": null
  }
}
```

2.2 Расчет сроков доставки - метод *GetDateArrival*.

GET

[api/v4/Public/GetDateArrival?areasSendId={areasSendId}&areasResiveld={areasResiveld}&dateSend={dateSend}¤cy={currency}&warehouseSendId={warehouseSendId}&warehouseResiveld={warehouseResiveld}](#)

Входные параметры

Наименование	Тип данных	Значение по умолчанию	Описание
areasSendId	Guid	*	Id Города отправления.
areasResiveId	Guid	*	Id Города прибытия.
dateSend	String	*	Дата отправления.
currency	Integer	100000000 (гривна)	Код валюты.
warehouseSendId	Guid?	null	Id Склада отправления.
warehouseResiveId	Guid?	null	Id Склада прибытия.

Выходные параметры

Представляет собой json. Коллекция объектов {arrivalDate}.

arrivalDate – Дата прибытия в формате YYYY-MM-DDThh:mm:ss±hh

sendDate – Дата отправки в формате YYYY-MM-DDThh:mm:ss±hh

arrivalDateStr – Дата прибытия в формате DD.MM.YYYY

sendDateStr – Город отправления в формате DD.MM.YYYY

Формат выходных параметров

application/json, text/json

Пример:

```
{
  "status": true,
  "message": "",
  "data": {
    "arrivalDate": "2021-09-20T16:00:00+03:00",
    "sendDate": "2021-09-17T20:00:00+03:00",
    "weightSummary": null,
    "volumeSummary": null,
    "arrivalDateStr": "20.09.2021",
    "sendDateStr": "17.09.2021"
  }
}
```

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<DopUsluguClassificationModel> dopUsluguClassifier; //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; //Высота
    double? lenght; //Длинна
    double? width; //Ширина
    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; //Минимальный вес
    decimal? maxWidth; //Максимальный вес
    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}¤cy={currency}&CitySendId={CitySendId}&CityReceiveId={CityReceiveId}&formalization={formalization}

Входные параметры

Наименование	Тип данных	Значение по умолчанию	Описание
culture	String	uk-UA	Культура; Допустимые значения (en-US, ru-RU, uk-UA).
currency	Integer	100000000	Код валюты
CitySendId	Guid	*	Id города отправления.
CityReceiveId	Guid	*	Id города получения.
formalization	Boolean	false	Выводить доп. услуги для расчета стоимости(false) или для оформления(true).

Выходные параметры

Представляет собой json. Коллекция объектов DopUslugaClassificationModel

Формат выходных параметров

application/json, text/json

Пример:

```
{
  "status": true,
  "message": "",
  "data": [
    {
      "classification": 1000000005,
      "name": "Упаковочные материалы",
      "dopUsluga": [
        {
          "uslugId": "2b4247c9-be8c-e211-be60-00155d037919",
          "name": "Доупаковка MAXI",
          "cost": 9.000000000,
          "count": 0,
          "classification": 0,
          "minWidth": null,
          "maxWidth": null,
          "summa": 0.0,
          "comment": null,
          "currency": 1000000001
        },
        {
          "uslugId": "3e9cde5d-bf8c-e211-be60-00155d037919",
          "name": "Доупаковка MIDI",
          "cost": 6.000000000,
          "count": 0,
          "classification": 0,
          "minWidth": null,
          "maxWidth": null,
          "summa": 0.0,
          "comment": null,
          "currency": 1000000001
        }
      ]
    }
  ]
}
```

3.3 Вывод справочника категорий тарифа - метод *GetTariffCategory*.

GET

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

Входные параметры

Наименование	Тип данных	Значение по умолчанию	Описание
culture	String	uk-UA	Культура; Допустимые значения (en-US, ru-RU, uk-UA).

Наименование	Тип данных	Значение по умолчанию	Описание
CitySendId	Guid	*	Город отправления.
CityReceiveId	Guid	*	Город получения.
WarehouseReceiveId	Guid	*	Склад получения.

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 – Минимальный вес
- MaxSize – Максимальный размер
- MinSize – Минимальный размер
- MaxWidth – Максимальный вес
- Length – Длина
- Width – Ширина
- Height – Высота
- RequiredWeight – Допустимый вес
- 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 Получение категорий груза - метод *GetCargoCategory*.

GET [api/v4/Public/GetCargoCategory?TariffCategoryId={TariffCategoryId}&culture={culture}](#)

Входные параметры

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

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>

```

3.5 Вывод справочника схем доставки - метод *GetDeliveryScheme*.

GET

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

Входные параметры

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

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
    }
  ]
}
```

3.6 Расчет стоимости перевозки - метод PostReceiptCalculate.

POST api/v4/Public/PostReceiptCalculate

Входные параметры

Наименование	Тип данных	Описание
input	CalculatorModel (см. пункт 4.1)	Модель описывающая входные и выходные параметры калькулятора.

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
      "help": 1, //Вес груза
      "size": 1 // Объем груза
    }
  ],
  "dopUslugaClassifier": [
    {
      "dopUsluga": [ //Массив доп. услуг
        {
          "uslugaId": "2b4247c9-be8c-e211-be60-00155d037919", //Id Доп. услуги
          "count": 1 //Количество доп. услуг
        },
        {
          "uslugaId": "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
"EconomyVesovojTarifValuta": 16.5, //Экономия на весовом тарифе
"EconomyEconomTarifValuta": 0, //Экономия на эконом-тарифе
"EconomySpecTarifValuta": 0, //Экономия на специальном-тарифе
"EconomyIndTarifValuta": 0, //Экономия на индивидуальном тарифе
"EconomyGlobalDiscountValuta": 0, //Экономия на глобальной скидке
"EconomyDiscountCardValuta": 0, //Экономия на дисконтной карте
"EconomyAktsiyaValuta": 0, //Экономия на акционной скидке от направления/склада
"EconomyOnAmountSkidka":0, //Экономия на размере скидки
"EconomySummary": 16.5, //Суммарная экономия на квитанции
"ComissionGM":100, //Комиссия наложенного платежа наличными: % от суммы + 10 грн
"culture":"ru-RU", //Культура
"calcType":null,
"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
"individualTarifId":null, //Id индивидуального тарифа
"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
    "cargoCategoryId":null, //Категория отправляемого груза
    "categoryIdName":"Груз", //Наименование категории груза
    "cargoCategoryIdName":"", //Название отправляемого груза
    "classification":0, //Код категории
    "countPlace":1, //Количество мест
    "help":1, //Вес
    "size":1, //Объем
    "height":0, //Высота
    "lenght":0, //Длинна
    "width":0, //Ширина
    "helpTarif":0.98, //Тариф за вес
    "egTarif":230, //Тариф за единицу груза
    "ofornlenie":1.5, //Стоимость оформления
    "ofornlenieCost":1.5, //Общая стоимость оформления
    "deliveryCost":230, //Стоимость перевозки
    "documentCost":230,
    "comment":"Расчет по общему тарифу за объем\r\n
Стоимость перевозки составляет 230,00(грн./м3)*1,0000(м3) = 230,00 грн.\r\n
При скидке в 0% сумма перевозки со скидкой составляет 230,00 грн.\r\n"
    "isEconom":false, //Экономная но более долгая доставка
    "isExpress":false, //Экспресс доставка?
    "isIndividual":null, //Индивидуальный тариф?
    "PartnerNumber":null, //Номер декларации партнера
    "weightSummary":1, //Общий вес
    "volumeSummary":1 //Общий объём
}],
"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. слуги
        "name":"Доупаковка МАХI", //Наименование доп. услуги
        "cost":9.00000000, //Стоимость доп. услуги

```

```

        "count":1, //Количество услуг
        "classification":100000005, //Код категории
        "minWidth":null, //Минимальный вес
        "maxWidth":null, //Максимальный вес
        "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 в ТЗ 1
Стоимость перевозки составляет 645,0000(грн./м3)*1,0000(м3) = 645,00 грн.
При скидке в 0,00% сумма перевозки со скидкой составляет 645,00 грн.
В стоимость квитанции включена доп.услуга 'Доупаковка МАХI-Прозрачная
(стрейч 3м, скотч 4м, гофрокартон 1м)' количеством 1 ед.
Общая сумма заказанной доп.услуги = 22,00 грн.
В стоимость квитанции включена доп.услуга 'Доупаковка MIDI-Прозрачная
(стрейч 2м, скотч 2м, гофрокартон 0,5м)' количеством 5 ед.
Общая сумма заказанной доп.услуги = 90,00 грн.
В стоимость квитанции включена доп.услуга
'Доставка груза от 101 кг до 300 кг' количеством 1 ед.
Общая сумма заказанной доп.услуги = 130,00 грн.
В стоимость квитанции включена доп.услуга
'Оформление багажа' количеством 1 ед.
Общая сумма заказанной доп.услуги = 3,00 грн.
В стоимость квитанции включена доп.услуга
'Услуга наложенного платежа' количеством 1 ед.
Общая сумма заказанной доп.услуги = 10,00 грн.
Стоимость перевозки склад-склад 645,00 грн.
Стоимость перевозки склад-склад со скидкой 645,00 грн.
Стоимость доп.услуг: 255,00 грн.
Стоимость страхования: 4000,00 грн.
Общая стоимость квитанции: 900,00 грн.
Стоимость транспортно-экспедиционных услуг округлена до 1,00 грн.
Стоимость услуги страхования округлена до 1,00 грн. ",
"viewType":0,
"currency":100000000 //Код валюты

```

```
}  
}
```

3.7 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	Тип данных	Значение по умолчанию	Описание
CitySendId	Guid?	null	Id города отправления.
CityReceiveId	Guid?	null	Id города получения.
WarehouseSendId	Guid	*	Id склада отправления.
WarehouseReceiveId	Guid	*	Id склада получения.
InsuranceValue	Double	*	Заявленная стоимость груза.
InsuranceCurrency	Integer	100000000	Валюта оплаты страхования.

Выходные параметры

Формат выходных параметров

application/json, text/json

Представляет собой json. Коллекция объектов {id,name }

Value – Стоимость страхования

MinValue – минимальная стоимость заявленного груза для данного направления

Пример:

В формате json.

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

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

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	Тип данных	Значение по умолчанию	Описание
culture	String	uk-UA	Культура; Допустимые значения (en-US, ru-RU, uk-UA).

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 оценки работы компании - метод *PostServiceRate*.

POST api/v4/Public/PostServiceRate

Входные параметры

Наименование	Тип данных	Описание
input	RateServicesModel	Модель описывающая входные и выходные параметры

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": ""
}

```

4.5 Отправка заказа автомобиля - метод PostPickUpCargo.

POST api/v4/Public/PostPickUpCargo

Входные параметры

Наименование	Тип данных	Описание
input	PickUpCargoModel	Модель описывающая входные и выходные параметры

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 – метод PostLogin

POST api/v4/Public/PostLogin

Входные параметры

Наименование	Тип данных	Описание
model	LoginModel	Модель описывающая входные и выходные параметры

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 – метод PostLogoff

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 – метод *GetUserInfo*.

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

Метод требует авторизации

Входные параметры

Наименование	Тип данных	Значение по умолчанию	Описание
culture	String	uk-UA	Культура; Допустимые значения (en-US, ru-RU, uk-UA).

Выходные параметры

Представляет собой json. Объект {Id, AccessLevel, UserName, SmsPhoneNumber, ClientType, ClientNumber, PhoneNumber, CrmUserId, Email, showHelpHide, Photo, RoleName, IsLoyaltyProgram, AvailablePoints, CurrentPoints, City, ConfirmPhone}.

Id – Id клиента в базе сайта

AccessLevel – Уровень доступа

UserName – Имя пользователя

SmsPhoneNumber – Номер для СМС пользователя

ClientType – Тип клиента (Физ(false)/Юр(true) лицо)

ClientNumber – Штрих-код клиента

PhoneNumber – Номер телефона пользователя

CrmUserId – Id клиента в CRM

Email – Адрес электронной почты

RoleName – Роль пользователя

IsLoyaltyProgram – Зарегистрирован в программе лояльности?

AvailablePoints – Доступные баллы программы лояльности

CurrentPoints – Текущие баллы программы лояльности

City – Город (Месторасположение клиента)

ConfirmPhone – Подтвержденный номер телефона пользователя (для входа на сайт)

Формат выходных параметров

application/json, text/json

Пример:

```
{
  "status": true,
  "message": "",
  "data": {
    "Id": "45023",
```

```

    "AccessLevel": "Полный доступ",
    "UserName": "!! Тестовый клиент для сайта",
    "SmsPhoneNumber": 662332658,
    "ClientType": false,
    "ClientNumber": "00022558",
    "PhoneNumber": 662332658,
    "CrmUserId": "abcdefab-0123-4567-89ab-0123456789ab",
    "Email": "test@test.ts",
    "showHelpHide": null,
    "Photo": null,
    "RoleName": "PowerUser",
    "IsLoyaltyProgram": false,
    "AvailablePoints": 0,
    "CurrentPoints": 0,
    "City": "Киев",
    "ConfirmedPhone": null
  }
}

```

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	Тип данных	Значение по умолчанию	Описание
page	Integer	1	Страница квитанций
rows	Integer	10	Количество выводимых строк
type	Integer	0	Тип квитанций (0-отправки, 1-получения)
culture	String	uk-UA	Культура; Допустимые значения (en-US, ru-RU, uk-UA).
detail	Boolean	true	Надо ли возвращать информацию по доп. услугам и возможным получателям.

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,
ReceiverWarehouseName – Склад получения квитанции,
Status – Статус квитанции,
StatusesDecoding – Текстовый статус квитанции,
TotalCost – Общая стоимость,
PartnerNumber – Номер декларации партнёра,
Type – тип квитанции (см. справочник 8.4),
Weight – Вес,
Volume – Объем,
PaymentStatus – Статус оплаты,
Currency – Валюта,
CanChangeReceiver – 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,
Receiver – 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,
PushStateCode – Статус отправки
codCost – заявленная стоимость груза,
codCurrency – валюта наложенного платежа,
SenderPhone – Телефон отправителя (null если не экспресс доставка)
ReceiverPhone – Телефон получателя (null если не экспресс доставка)
AddressPickup – Адрес приема экспресс доставки (null если не экспресс доставка)
AddressDelivery – Адрес забора экспресс доставки
DateArrivalExpress – Дата и время экспресс доставки
CitySendName – Город отправления
DeliveryType – Схема доставки
codName – имя получателя средств наложенного платежа,
codWarehouse – склад получения средств наложенного платежа,
isGiveMoney – выданы ли деньги получателю наложенного платежа
codGiveMoneyDate – дата выдачи наложенного платежа отправителю
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": "!! Іванков Іван Тест",
      "Receipient": "!! НОВЫЙ КЛИЕНТ КИЕВ",
      "Payer": "!! Іванков Іван 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.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 в 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 – метод GetClientCards.

GET api/v4/Public/GetClientCards

Метод требует авторизации через API ключ

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-a2f2-5ddb34a07ce",
      "name": "5211*****9950"
    },
    {
      "id": "42b137f9-eb86-424b-b2ed-60b5b4659299",
      "name": "5168*****8737"
    }
  ],
  "status": true,
  "message": ""
}
```

В формате xml.

```
<ApiResult>
  <status>true</status>
  <message/>
  <data>
    <DirectoryItem>
      <id>3dafcedb-904b-4210-ae46-2af2acd385ef</id>
      <name>5104*****8490</name>
    </DirectoryItem>
    <DirectoryItem>
      <id>c05e1eae-80e2-4380-a2f2-5ddb34a07ce</id>
      <name>5211*****9950</name>
    </DirectoryItem>
    <DirectoryItem>
      <id>42b137f9-eb86-424b-b2ed-60b5b4659299</id>
      <name>5168*****8737</name>
    </DirectoryItem>
  </data>
</ApiResult>
```

6.3 Получение списка расчетных счетов клиента – метод *GetClientInvoices*.

GET api/v4/Public/GetClientInvoices

Метод требует авторизации через API ключ

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 Оформление квитанции - метод PostCreateReceipts.

POST api/v4/Public/PostCreateReceipts

Метод требует авторизации через API ключ

Входные параметры представляют собой массив моделей RegistrationReceiptsModel со следующими полями

Наименование	Тип данных	Обязательное? (Да/Нет)	Описание
deliveryScheme	Integer	По умолчанию: 0	Схема доставки 0: Склад-склад, 1: Двери-двери, 2: Склад-двери, 3: Двери-склад

Наименование	Тип данных	Обязательное? (Да/Нет)	Описание
areasSendId	Guid	Да	Guid города отправки
warehouseSendId	Guid	Да, при deliveryScheme = 0 или 2	Guid склада в городе отправки
pickUpContactName	String	Да, при deliveryScheme = 1 или 3	Контактное лицо для забора
pickUpContactPhone	String	Да, при deliveryScheme = 1 или 3	Телефон для забора (10 цифр. Н-р.: "0501112233")
pickUpAddressId	Guid	Одно из данных полей обязательно к заполнению при deliveryScheme = 1 или 3	Guid адреса забора
pickUpAddress	String		Адрес забора (через запяты: Индекс, улица, дом, квартира)
pickUpDate	String	Всегда принимает значение даты создания квитанции. Используется при deliveryScheme = 1 или 3	Дата забора
areasResiveId	Guid	Да	Guid города получения
warehouseResiveId	Guid	Да, при deliveryScheme = 0 или 3	Guid склада в городе получения
deliveryContactName	String	Да, при deliveryScheme = 1 или 2	Контактное лицо для доставки
deliveryContactPhone	String	Да, при deliveryScheme = 1 или 2	Телефон для доставки (10 цифр. Н-р.: "0501112233")
DeliveryComment	String	Нет	Комментарий при доставке

Наименование	Тип данных	Обязательное? (Да/Нет)	Описание
deiveryAddresId	Guid		Guid адреса доставки
deliveryAddress	String	Одно из данных полей обязательно к заполнению при deliveryScheme = 1 или 2	Адрес доставки (через запятые: Индекс, улица, дом, квартира)
SenderId	Guid	По умолчанию: Guid владельца API ключа (публичный ключ)	Guid отправителя. Может принять значение вашего Id или Id вашей родительской или дочерней организации.
possibleResiverReceipt_1	Guid	Да, либо создать нового с помощью полей (1)	Guid существующего получателя.
1) receiverName		Да, при создании получателя	Ф.И.О. получателя (физ. лица)
receiverType		По умолчанию: false	false: физ.лицо, true: юр.лицо
receiverPhone		Нет для создания физ.лиц Да для создания юр.лиц	Телефон получателя (10 цифр. Н-р: "0501112233")
receiverEgrpо		Нет для создания физ.лиц Да для создания юр.лиц	Egrpо
possibleResiverReceipt_2 possibleResiverReceipt_3 possibleResiverReceipt_4	Guid	Нет	Guid возможного получателя.
dateSend	String	Да	Дата отправки
Currency	Integer	По умолчанию: 100000000	Код валюты
payerType	Integer	По умолчанию: 0	Тип плательщика: 0: отправитель, 1: получатель

Наименование	Тип данных	Обязательное? (Да/Нет)	Описание
paymentType	Integer	По умолчанию: 0	Тип оплаты: 0: нал., 1: безнал.
payerId	Guid	По умолчанию: 0	Если плательщиком надо передать третье лицо, то передать Guid третьего лица.
paymentTypeInsurance	Integer	По умолчанию: 0	Тип оплаты страховки: 0: нал., 1: безнал.
payerInsuranceId	Integer	При paymentType = 0 – по умолчанию senderId. При paymentType = 1 – генерирует, если null	Плательщик страховки
InsuranceValue	Integer	По умолчанию: 0	Застрахованная стоимость груза
CashOnDeliveryPayerAccountId	Guid	Нет	Плательщик наложенного платежа
cashOnDeliveryValue	Integer	Нет	Сумма наложенного платежа
cashOnDeliveryValuta	Integer	По умолчанию: 100000000	Валюта наложенного платежа
cashOnDeliveryType	Integer	По умолчанию: 0	Тип наложенного платежа 0: платёжная карта, 1: расчётный счет, 2: наличными, 3: безопасная сделка
CashOnDeliverySafetyDeal	Boolean	Необязательное, использ. при cashOnDeliveryType = 3	Флаг: безопасная сделка?
CashOnDeliveryWarehouseId	Guid	Обязательное при cashOnDeliveryValue > 0	Guid склада выплат наложенного платежа

Наименование	Тип данных	Обязательное? (Да/Нет)	Описание
CashOnDeliveryRasschSchetId	Guid	Обязательное при cashOnDeliveryValue > 0 и cashOnDeliveryType = 1 или 3	Id расчетного счета
CashOnDeliveryCardId	Guid	Обязательное при cashOnDeliveryValue > 0 и cashOnDeliveryType = 0	Id платежной карты
CashOnDeliverySenderPhone	String	Обязательное при cashOnDeliveryValue > 0 и cashOnDeliveryType = 2	Телефон получателя наложенного платежа
CashOnDeliverySenderFullName	String	Обязательное при cashOnDeliveryValue > 0 и cashOnDeliveryType = 2 и Если получатель юр.лицо	Ф.И.О. получателя наложенного платежа
CashOnDeliveryReceiverPhone	String	Да при cashOnDeliveryValue>0 и cashOnDeliveryType = 2 и Если плательщик юр.лицо	Телефон плательщика наложенного платежа
CashOnDeliveryReceiverFullName	String	Да при cashOnDeliveryValue>0 и cashOnDeliveryType = 2 и Если плательщик юр.лицо	Ф.И.О. плательщика наложенного платежа
ReturnDocuments	Boolean	Флаг возврат документов при доставке	Нет
descentFromFloor	Integer	Спуск с этажа	Нет
climbingToFloor	Integer	Подъем на этаж	Нет
IsOverSize	Boolean	Негабарит, для Доставки груза	Нет
IsGidrobot	Boolean	Гидроборт, для Доставки груза	Нет

Наименование	Тип данных	Обязательное? (Да/Нет)	Описание
EconomDelivery	Boolean	Эконом доставка	Нет
EconomPickUp	Boolean	Эконом забор, для Забора груза	Нет
ExpressPickUp	Boolean	Экспресс забор, для Забора груза	Нет
parentNumber	String	Номер декларации партнера	Нет
DeliveryComment	String	Комментарий при доставке	Нет
category	Category Model[]	Массив грузов	Да

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

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 **pickUpAddress** и **deliveryAddress** соответственно. Или передав их **Id** в поля **pickUpAddressId** и **deliveryAddressId**.

Если указаны поля **possibleResiverReceipt_1**, **pickUpAddressId** или **deliveryAddressId**, то поля для создания новых **получателя**, **адреса забора** или **адреса доставки** (соответственно) не учитываются.

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

Пример входных данных:

```
{
  "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", //
    "DeliveryComment": "Комментарий при доставке", //Комментарий при доставке
    "ReturnDocuments": true, //Флаг возврат документов 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
    "cashOnDeliveryType": 0, //Тип наложенного платежа 0-платежная карта, 1-
расчетный 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": [
      {

```

отправляемого груза

```
        "categoryId": "00000000-0000-0000-0000-000000000000", //Категория тарифа
        "cargoCategoryId": "0307d03b-9e36-e311-8b0d-00155d037960", //Категория
    },
    {
        "countPlace": 10, //Количество мест
        "help": 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,
        "help": 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
            "countPlace": 1, //Количество мест
            "help": 20.0, //Вес
            "size": 1.0, //Объем
            "isEconom": true,
            "PartnerNumber": "1234567"
        }
    ]
}
]
}
```

отправляемого груза

Выходные параметры

Представляет собой json. Коллекция объектов {Id, Number, TotalCost, InsuranceCost, ComissionGM, Comment, egs}.

Id – Guid квитанции

Number – Номер квитанции

TotalCost – Стоимость перевозки

InsuranceCost – Стоимость страхования

ComissionGM – Комиссия наложенного платежа наличными(% от суммы +10грн за услугу нал. плат.)

Comment – Комментарий

egs – Массив единиц груза

Формат выходных параметров

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

Пример:

```
{
  "status": true,
  "message": [],
  "receipts": [
    {
      "Id": "f5a947f6-adcf-49e8-be46-a49d69621ae2",
      "Number": "9900000000",
      "TotalCost": 97.0,
      "InsuranceCost": 7.0,
      "ComissionGM": 30.0,
      "Comment": "",
      "egs": [
        {
          "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>
```

6.5 Деактивация единиц груза - метод PostDeactivateEg.

POST api/v4/Public/PostDeactivateEg

Метод требует авторизации через API ключ

Входные параметры

Наименование	Тип данных	Описание
input	ReceiptsOutputModel	Модель описывающая входные и выходные параметры

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.

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

6.6 Получение документов в PDF - метод *GetPdfDocument*.

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

Метод требует авторизации через API ключ

Входные параметры

Наименование	Тип данных	Значение по умолчанию	Описание
number	String	*	Номера квитанций (номера указываются через точку с запятой без пробелов, н-р, number=9900112233;9900223344)
type	Integer	*	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.

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

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

GET api/v4/Public/GetSenderList

Метод требует авторизации через API ключ

Входные параметры

Отсутствуют.

Выходные параметры

Представляет собой json. Коллекция объектов {id, name, cityId, cityName}.

id – id отправителя

name – Имя отправителя

cityId – Guid города отправителя

cityName – Название города отправителя

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>
```

6.8 Получение доступных валют - метод *GetCurrency*.

GET

[api/v4/Public/GetCurrency?CitySendId={CitySendId}&CityReceiveId={CityReceiveId}&PayerType={PayerType}&PayerId={PayerId}&culture={culture}](#)

Входные параметры

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

Выходные параметры

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

id – id валюты

name – Наименование валюты

Формат выходных параметров

application/json, text/json

Пример:

В формате json.

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

В формате xml.

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

```

        <id>1000000000</id>
        <name>Гривня</name>
    </DirectoryItem>
</data>
</ApiResponse>

```

6.9 Получение списка плательщиков - метод *GetAvailableServices*.

GET

api/v4/Public/GetAvailableServices?GetAvailableServices?scheme={scheme}&receiveWarehouseId={receiveWarehouseId}&CodValue={CodValue}

Входные параметры

Наименование	Тип данных	Значение по умолчанию	Описание
scheme	Integer	*	Схема доставки (0 - Склад-Склад, 1 - Двери-Двери, 2 - Склад-Двери, 3 - Двери-Склад)
receiveWarehouseId	Guid	*	Id склада получения
CodValue	Decimal?	null	Сумма наложенного платежа

Выходные параметры

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

- isInfo – доступна ДУ Информационное обслуживание
- isReturnDocs – доступна ДУ Возврат выдачи груза
- isDenyIssue – доступна ДУ запрет выдачи груза
- isDescent – доступна ДУ Спуск с этажа
- isLifting – доступна ДУ Подъем на этаж
- isAutoReturn – доступна ДУ Услуга автовозврата

Формат выходных параметров

application/json, text/json

Пример:

В формате json.

```

{
  "data": [
    {
      "isInfo": "true",
      "isReturnDocs": true, //доступна ДУ Возврат документов
      "isDenyIssue": true, //доступна ДУ Запрет выдачи груза
      "isDescent": true, //доступна ДУ Спуск с этажа
      "isLifting": true, //доступна ДУ Подъем на этаж
      "isAutoReturn": true //доступна ДУ Услуга автовозврата
    }
  ],

```

```
"data2": null,  
"status": true,  
"message": "",  
"code": 0  
}
```

6.10 Получение списка плательщиков - метод *GetPayer*.

GET

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

Метод требует авторизации через API ключ

Входные параметры

Наименование	Тип данных	Значение по умолчанию	Описание
CitySendId	Guid	*	Id города отправления.
CityReceiveId	Guid	*	Id города получения.
ClientSenderId	Guid	*	Id клиента отправителя
ClientReceiverId	Guid?	null	Id клиента получателя.
PayerType	Integer?	null	Тип плательщика 0 – отправитель, 1 – получатель, 2 – третье лицо.

Выходные параметры

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

id – id плательщика

name – Имя плательщика

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>
```

6.11 Получение адресов клиента - метод *GetClientAddress*.

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

Входные параметры

Наименование	Тип данных	Значение по умолчанию	Описание
CityId	Guid	*	Id города.
ClientId	Guid	*	Id клиента.

Выходные параметры

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

id – Guid адреса

name – адрес

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.

```
<ApiResult>
  <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>
</ApiResult>
```

6.12 Получение возможных получателей клиента- метод *GetPosibleReciver.*

GET

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

Метод требует авторизации через API ключ

Входные параметры

Наименование	Тип данных	Значение по умолчанию	Описание
CityReceiveId	Guid	*	Id города получения.
ClientSenderId	Guid	*	Id клиента отправителя.

Выходные параметры

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

id – Guid получателя

name – Имя получателя

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, ФОП</name>
    </DirectoryItem>
    <DirectoryItem>
      <id>ce30c74a-7253-400e-b98f-f7a19a811731</id>
      <name> Скаско Олена Миколаївна</name>
    </DirectoryItem>
  </data>
</ApiResult>
```

6.13 Получение вида оплаты клиента - метод *GetClientPaymentType*.

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

Входные параметры

Наименование	Тип данных	Значение по умолчанию	Описание
ClientId	Guid	*	Id клиента.

Выходные параметры

Представляет собой json.

data – вид оплаты клиента (bool). True = наличный, False = безналичный

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>
```

6.14 Получение полной информации по квитанции - метод *GetFullReceiptInformation*.

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

Метод требует авторизации логином и паролем

Входные параметры

Наименование	Тип данных	Значение по умолчанию	Описание
culture	String	uk-UA	Культура; Допустимые значения (en-US, ru-RU, uk-UA).
number	String	*	Номер квитанции

Выходные параметры

Представляет собой json. Коллекция объектов {areasSendId, areasSend, areasReceiveId, areasResive, warehouseSendId, warhouseSend, warehouseReceiveId, warhouseReceive, deliveryScheme, number, sender, senderId, receiver, receiverId, payer, payerId, paymentType, dateSend, dateReceive, state, Currency, partnerNumber, paymentStatus, paymentDate, lockShipping, totalCountPlace, totalWeight, totalSize, warehouseWarehouseAmount, discountAmount, lossesDescountAmount, totalAmount, insuranceValue, SafetyDealMoneyStatus, duArray, possibleReceiverArray, receiptsArray[number, state, receiptType, paymentType, paymentStatus, paymentDate, currency, payerId, payer, totalAmount, clientCardId, clientCard, codSender, codSenderPhone, isGiveMoney, codWarehouse, codCity], egArray}.

areasSendId – Guid города отправления

areasSend – город отправления

areasReceiveId – Guid города прибытия

areasResive – город прибытия

warehouseSendId – Guid склада отправления

warhouseSend – склад отправления

warehouseReceiveId – Guid склада прибытия

warhouseReceive – склад прибытия

deliveryScheme – схема доставки: 0 - склад-склад, 1 - двери-двери, 2 - склад-двери, 3 - двери-склад

number – номер квитанции

sender – имя отправителя

senderId – Guid отправителя

receiver – имя получателя

receiverId – Guid получателя

payer – имя плательщика

payerId – Guid плательщика

paymentType – тип оплаты: 0 - нал., 1 - безнал.

dateSend – дата отправления

dateReceive – дата прибытия

state – статус квитанции (см. справочник 8.1)

Currency – код валюты

partnerNumber – номер декларации партнера

paymentStatus – статус оплаты

paymentDate – дата оплаты

lockShipping – запрет на выдачу

totalCountPlace – общее количество посылок

totalWeight – общий вес

totalSize - общий размер

warehouseWarehouseAmount – цена перевозки без скидки и дополнительных услуг

discountAmount – сумма скидки

lossesDescountAmount – сумма потерь на скидках

totalAmount – общая стоимость по квитанции

insuranceValue – страховая стоимость груза

SafetyDealMoneyStatus – статус денежных средств по безопасной сделке

duArray – список дополнительных услуг

possibleReceiverArray – список возможных получателей

egArray – список единиц груза

receiptsArray – список связанных квитанций

```
[  
    Number – номер квитанции,  
    state – статус квитанции,  
    receiptType – тип квитанции,  
    paymentType – Тип оплаты 1-безнал., 0-нал,  
    paymentStatus – Флаг, статус оплаты,  
    paymentDate – дата оплаты,  
    currency – код валюты,  
    payerId – id плательщика,  
    payer – имя плательщика,  
    totalAmount – общая стоимость по квитанции,  
    clientCardId – id карты клиента,  
    clientCard – сокращенный номер карты клиента,  
    codSender – имя отправителя,  
    codSenderPhone – телефон отправителя,  
    isGiveMoney – выданы ли деньги получателю наложенного платежа,  
    codWarehouse – склад отправления/получения наложенного платежа,  
    codCity – город отправления/получения наложенного платежа,  
    codGiveMoneyDate – дата выдачи наложенного платежа отправителю  
]
```

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": "Наложенный 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": [
    {
        "cargoCregoryId": "0656bab4-e62c-e411-bd10-000d3a200936",
        "cargoCregory": "Автоаксесуары",
        "count": 1,
        "weight": 1.000,
        "size": 1.000000,
        "isEconomy": false,
        "cost": 2952.000
    },
    {
        "cargoCregoryId": "0f07d03b-9e36-e311-8b0d-00155d037960",
        "cargoCregory": "Документы",
        "count": 1,
        "weight": 0.010,
        "size": 0.010000,
        "isEconomy": false,
        "cost": 30.000
    }
]
}
}
}

```

6.15 Создание адреса или получателя - метод *PostCreateAddressOrClient*.

POST api/v4/Public/PostCreateAddressOrClient

Метод требует авторизации через API ключ

Входные параметры

Наименование	Тип данных	Описание
input	ClientModel	Модель описывающая входные параметры

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

```

```

    "Egrpо": "", //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
  "Appartament": "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",
      "Appartament": "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
    }
  }
}

```

```
}
```

6.16 Получение информации по стикеру квитанции - метод *GetStickers*.

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

Метод требует авторизации через API ключ

Входные параметры

Наименование	Тип данных	Значение по умолчанию	Описание
number	String	*	Номер квитанции

Выходные параметры

Представляет собой json. Коллекция объектов {barcode, categoryName, receiptNumber, receiver, dateSend, dateReceive, warehouseSend, warehouseReceive, totalPlaces, rang, econom, delivery}.

- Barcode – штрихкод
- categoryName – категория товара
- receiptNumber – номер квитанции
- receiver – имя получателя
- dateSend – дата отправления
- dateReceive – дата прибытия
- warehouseSend – склад отправления
- warehouseReceive – склад прибытия
- totalPlaces – общее количество посылок
- rang – ранг
- econom – экономная, но более долгая доставка
- delivery – есть ли адресная доставка

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.

```

<ApiResultStickers>
  <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>
      <econom>>false</econom>
      <delivery>true</delivery>
      <postomat>>false</postomat>
    </Sticker>
  </data>
</ApiResultStickers>

```

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

POST api/v4/Public/PostAddReceiptIntoPickUpRequest

Метод требует авторизации через API ключ

Входные параметры

Наименование	Тип данных	Описание
input	ExptactReceiptModel	Модель описывающая входные и выходные параметры квитанции

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>
```

6.18 Получение реестра отправки - метод *SendingRegister*.

GET

uk-UK/SharedForms/SendingRegister?id={id}

Входные параметры

Наименование	Тип данных	Значение по умолчанию	Описание
id	String	*	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 – метод *GetUnidersalLogsByReceiptNumber*.

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

Метод требует авторизации

Входные параметры

Наименование	Тип данных	Значение по умолчанию	Описание
number	String	*	Номер квитанции
culture	String	uk-UA	Культура; Допустимые значения (en-US, ru-RU, uk-UA).

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
    }
  ]
}
```

8. Дополнительные справочники

8.1 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	Переадресована на другой склад
10	На выгрузке на складе
11	Груз прибыл на транзитный склад
12	Готовится к доставке курьером
13	Доставляется курьером

8.2 Справочник валют

Код	Значение
100000000	Гривна
100000001	Рубль

8.3 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

8.4 Справочник типов квитанций

Код	Значение
2	Обычная квитанция
4	Переадресация
5	Доставка
6	Страхование
7	Забор груза
8	Продажа услуг
10	Наложенный платеж картой
11	Курьерская доставка
13	Наложенный платёж
14	Возвратный платёж