API for online stores to work with Delivery.

Version 3.4.1

Contents

Introduction

Json format for data exchenge.
Web service adress http://www.delivery-auto.com/api/

Previous version API is availabe by link http://www.delivery-auto.com/api/v1/ or http://www.delivery-auto.com/api/

The output parameters have the format:
```
{
  "status": true,
  "message": "",
  "data": Data
}
  On a successful execution of the command status == true.
  In case of an exception status == false, variable message contains a message of
error.
```

Representations

1.1 Getting a list of regions - method GetRegionList.

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

Input parameters

Name	Data type	Default value	Description
culture	String	uk-UA	Culture; Vaild values (en-US, uk-UA).
country	Integer?	null	Country id (1-Ukraine, null - all)

Output parameters

Represents as json. Collection of objects {id, name, externalId}.

id – region id
name – Region name
externalId – region Id

Output parameters format
application/json, text/json

Example:

```
{
  "status": true,
  "message": "",
  "data": [
    {
      "id": -1,
      "name": "BCE"
```

```
        "externalId": "00000000-0000-0000-0000-000000000000"
      },
      {
        "id": 3898,
        "name": "Винницкая область",
        "externalId": "c8ad84fe-cf49-e211-9515-00155d012d0d"
      }
    ]
  }
```

1.2 Getting a list of cities – method GetAreasList.

GET api/v4/Public/GetAreasList?culture={culture}&fl_all={fl_all}&regionId={regionId}&country={country}&cityName={cityName}

Input parameters

Name	Data type	Default value	Description
culture	String	uk-UA	Culture; Valid values (en-US, uk-UA).
fl_all	Boolean	false	A flag that allows you to show all the cities where it is possible to provide company services
regionId	Integer?	null	Region Id
country	Integer?	null	Country Id (1-Ukraine, null - all)
cityName	String	null	A city name on selected in parameter “culture” language

Output parameters

Represents as json. Collection of objects {id, name, RegionId, IsWarehouse, ExtracityPickup, ExtracityShipping, RAP, RAS, regionName, regionId, country, districtName}.

id – city id
name – City name
RegionId – Region id
IsWarehouse – Warehouse flag (1-there is a warehouse in a city)
ExtracityPickup – true = Executed out of city pick up
ExtracityShipping – true = Executed out of city delivery
RAP – true = Regional pick up
RAS – true = Regional delivery
regionName – regional name
regionId – region id
country – country code (1 –Ukraine)
districtName – district name

Output parameters format
application/json, text/json

Example:

```
{
  "status": true,
  "message": "",
  "data": [
    {
      "id": "2d481888-1429-e311-8b0d-00155d037960",
      "name": "Бар",
      "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 Getting a list of representatives - method GetWarehousesList.

GET api/v4/Public/GetWarehousesList?culture={culture}&includeRegionalCenters={includeRegionalCenters}&CityId={CityId}&RegionId={RegionId}&country={country}

Input parameters

Name	Data type	Default values	Description
culture	String	uk-UA	Culture; Valid values (en-US, uk-UA)
includeRegionalCenters	Boolean	false	To display offices?

Name	Data type	Default values	Description
CityId	Guid?	null	city Id
RegionId	Guid?	null	region Id.
country	Integer?	null	country Id (1-Ukraine, null - all)

Output parameters

Represents as json. Collection of objects {id, name, address, Latitude, Longitude, CityId, LatitudeCorrect, LongitudeCorrect, IsCashOnDelivery, CenterPickUpDelivery}.

id – warehouse id
name – warehouse name
address – warehouse address
Latitude – latitude (incorrect, because it’s mixed up in places with the longtitude)
Longitude – longtitude (incorrect, because it’s mixed up in places with the latitude)
CityId – city Id
LatitudeCorrect – Correct latitude
LongitudeCorrect – Correct longtitude
IsCashOnDelivery – is there a cash on delivery service at the warehouse
CenterPickUpDelivery – is there a delivery pick up center at the warehouse

Output parameters format

application/json, text/json

Example:

```
{
  "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 *Getting a detailed information about the representatives - method GetWarehousesInfo.*

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

Input parameters

Name	Data type	Default value	Description
culture	String	uk-UA	Culture; Valid values (en-US, uk-UA).
WarehousesId	Guid	*	representative Id.

Output parameters

Represents as json. Collection of objects {id, name, address, operatingTime, Phone, EmailStorage, Latitude, Longitude, LatitudeCorrect, LongitudeCorrect, Office, CityId, CityName, IsWarehouse, RcPhoneSecurity, RcPhoneManagers, RcPhone, RcName, WarehouseForDeliveryId, IsCashOnDelivery, WarehouseType, CenterPickUpDelivery}.

id – Representative id
name – Representative name
address – Representative address
operatingTime – Representative worktime
Phone – Representative phone numbers
EmailStorage – Representative email
Latitude – Latitude (incorrect, because it’s mixed up in places with the longtitude)
Longitude – Longtitude (incorrect, because it’s mixed up in places with the latitude)
LatitudeCorrect – Correct latitude
LongitudeCorrect – Correct longtitude
Office – Sign of office
CityId – city id
CityName – city name
IsWarehouse – is a warehouse
RcPhoneSecurity – security phone number
RcPhoneManagers – managers phone number
RcPhone – regional center phone number
RcName – regional center name
WarehouseForDeliveryId – warehouse for delivery
IsCashOnDelivery – is there a cash on delivery service at the warehouse
WarehouseType – warehouse type
CenterPickUpDelivery – is there a delivery pick up center at the warehouse

Output parameters format

application/json, text/json

Example:

```
{
  "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 Getting a list of the warehouses - method GetWarehousesListByCity.

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

Input parameters

Name	Data type	Default value	Descriptpion
CityId	Guid	*	City Id
DirectionType	Integer	*	Direction type: 0 – departure warehouses, 1 – receiving warehouses.
culture	String	uk-UA	Culture; Valide values (en-US, uk-UA).

Output parameters

Represents as json. Collection of objects {id, name, address, operatingTime, Phone, EmailStorage, Latitude, Longitude, LatitudeCorrect, LongitudeCorrect, Office, CityId, CityName, IsWarehouse, RcPhoneSecurity, RcPhoneManagers, RcPhone, RcName, WarehouseForDeliveryId, IsCashOnDelivery, WarehouseType, CenterPickUpDelivery}.

id – Represntative id
name – Representative name
address – Representative address
operatingTime – Representative work time
Phone – Representative phone numbers
EmailStorage – Representative email
Latitude – Latitude (incorrect because it’s messed up in places with the longtitude)
Longitude – Longtitude (incorrect because it’s messed up in places with the latitude)
LatitudeCorrect – Correct latitude
LongitudeCorrect – Correct longtitude
Office – Sign of office
CityId – city id
CityName – city name
IsWarehouse – is a warehouse
RcPhoneSecurity – security phone number
RcPhoneManagers – managers phone number
RcPhone – regional center phone number
RcName – regional center name
WarehouseForDeliveryId – delivery warehouse
IsCashOnDelivery – is there a cash on delivery service at the warehouse
WarehouseType – warehouse type
CenterPickUpDelivery – is there a delivery pick up center at the warehouse

Output parameters format
application/json, text/json

Example:

```
{
  "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 Search for the nearest representatives - method GetFindWarehouses.

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

Input parameters

Name	Data type	Default type	Description
culture	String	uk-UA	Culture; Valid values (en-US, uk-UA).
Longitude	Double	*	Point longtitude, from which the search is executed
Latitude	Double	*	Point latitude, from which the search is executed
count	Integer	1	The number of representatives returned in order of increasing distance from the specified point.
includeRegionalCenters	Boolean	false	To display offices.
CityId	Integer?	null	City Id
type	Integer?	null	Warehouse type (0-warehouse, 3-cash on delivery warehouse)
country	Integer?	null	Country Id (1-Ukraine, null - all)

Output parameters

Represents as json. Collection of objects {id, name, distance, latitude, longitude, LatitudeCorrect, LongitudeCorrect, cityName, address, IsWarehouse, Phone, working_time, WarehouseType, IsRegionalCentre}.

id – Representative id
name – Representative name
distance – Distance from specified point
latitude – Latitude (incorrect because it’s mixed up in places with the longtitude)

longitude – Logtitude (incorrect becouse it’s mixed up in places with the latitude)
latitudeCorrect – Correct latitude
longitudeCorrect – Correct longtitude
cityName – City name
address – Warehouse address
IsWarehouse – Is a warehouse
phone – Warehouse phone number
working_time – work time
WarehouseType – 0-warehouse, 3- cash on delivery warehouse
IsRegionalCentre – Is a regional center

Output parameters format

application/json, text/json

Example:

```
{
  "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 Getting a list of the representatives with detailed information about a city - method GetWarehousesListInDetail.

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

Input parameters

Name	Data type	Default value	Description
culture	String	uk-UA	Culture; Valid values (en-US, uk-UA).
CityId	Guid?	null	City Id.
onlyWarehouses	Boolean	false	Flag – warehouses only
country	Integer?	null	Country Id (1-Ukraine, null - all)

Output parameters

Represents as json. Collection of objects {id, name, address, operatingTime, Phone, EmailStorage, latitude, longitude, latitudeCorrect, longitudeCorrect, Office, CityId, CityName, IsWarehouse, RcPhoneSecurity, RcPhoneManagers, RcPhone, RcName, WarehouseForDeliveryId, IsCashOnDelivery, WarehouseType, CenterPickUpDelivery}.

id – Representative id
name – Representative name
address – Representative address
operatingTime – Representative work time
Phone – Representative phone numbers
EmailStorage – Representative Email
latitude – Latitude (incorrect because it's mixed up in places with longtitude)
longitude – Longtitude (incorrect because it's mixed up in playses with latitude)
latitudeCorrect – Correct latitude
longitudeCorrect – Correct longtitude
Office – Sign of office
CityId – City id
CityName – City name
IsWarehouse – Is a warehouse
RcPhoneSecurity – Security phone number
RcPhoneManagers – Managers phone number
RcPhone – Regional center phone number
RcName – Regional center phone number
WarehouseForDeliveryId – Delivery warehouse Id
IsCashOnDelivery – Is there a cash on delivery service at the warehouse

WarehouseType – Warehouse type

CenterPickUpDelivery – Is there a pick up center at the warehouse

Output parameters format

application/json, text/json

Example:

```
{
  "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. Receipts

2.1 Search for a receipt - method GetReceiptDetails.

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

Input parameters

Name	Data type	Default value	Description
culture	String	uk-UA	Culture; Valid values (en-US, uk-UA).
number	String	*	Receipt number

Output parameters

Represnts json. Collection of objects {id, number, SendDate, ReceiveDate, CreatedData, SenderWarehouseName, RecepientWarehouseName, Discount, TotalCost, Status, Weight, Volume, Sites, PaymentStatus, Currency, InsuranceCost, InsuranceValue, InsuranceCurrency, PushStateCode, CodCost, CodCurrency, Type, SenderPhone, ReceiverPhone, CitySendName, CityReceiveName, DeliveryType, StatusesDecoding, SafetyDealMoneyStatus, InsuranceInfo}.

id – Receipt id
number – Receipt number
SendDate – Dispatch date
ReceiveDate – Receiving date
CreatedData – Creation date
SenderWarehouseName – Dispatch warehouse
RecepientWarehouseName – Receipt warehouse
Discount – Discount amount
TotalCost – Receipt total cost
Status – Receipt current status
Weight – Cargo total weight
Volume – Cargo volume
Sites – Amount of sites
PaymentStatus – Payment status
Currency – Receipt currency
InsuranceCost – Insurance cost
InsuranceValue – Declared value
InsuranceCurrency – Insurance currency
PushStateCode – Dispatch status
CodCost – Cash on delivery amount
CodCurrency – Cash on delivery currency
Type – Receipt type (see directory 8.4)
SenderPhone – Sender phone number
ReceiverPhone – Receiver phone number
CitySendName – Sender city
CityReceiveName – Receiver city
DeliveryType – Delivery scheme (see paragraph 3.4)

StatusesDecoding – Receipt status

SafetyDealMoneyStatus – Safe deal cash status

Output paramters format

application/json, text/json

Example:

```
 {
  "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",
     "RecepientWarehouseName": "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 Delivery time calculation - method GetDateArrival.

GET api/v4/Public/GetDateArrival?areasSendId={areasSendId}&areasResiveId={areasResiveId}&dateSend={dateSend}&currency={currency}&warehouseSendId={warehouseSendId}&warehouseResiveId={warehouseResiveId}

Input parameters

Name	Data type	Default value	Description
areasSendId	Guid	*	Sender city Id.
areasResiveId	Guid	*	Arrival city Id.
dateSend	String	*	Dispatch date.
currency	Integer	100000000 (гривна)	Currency code.
warehouseSendId	Guid?	null	Dispatch warehouse id.
warehouseResiveId	Guid?	null	Arrival warehouse id.

Output parameters

Represents as json. Collection of objects {arrivalDate}.

arrivalDate – Arrival date in format YYYY-MM-DDThh:mm:ss±hh
sendDate – Dispatch date in format YYYY-MM-DDThh:mm:ss±hh
arrivalDateStr – Arrival date in format DD.MM.YYYY
sendDateStr – Dispatch date in format DD.MM.YYYY

Output parameters format
application/json, text/json

Example:

```
{
  "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. Transportation cost calculation

3.1 Data models for information exchange:

```
    class CalculatorModel //Main calculator model
    {
        string culture; //Culture
        string areasSendId; //Dispatch city id
        string areasResiveId; //Receiving city id
        string warehouseSendId; //Dispatch warehouse id
        string warehouseResiveId; //Receipt warehouse id
        string areasSendIdName; //Dispatch city name
        string areasResiveIdName; //Receiving city name
        string warehouseSendIdName; //Dispatch warehouse name
        string warehouseResiveIdName; //Receiving warehouse name
        double CashOnDeliveryValue; //Cash on delivery amount
        int CashOnDeliveryValuta; //Cash on delivery currency
        double InsuranceValue; //Cargo insurance value
        decimal InsuranceCost; //Insurance cost
        DateTime? dateSend; //Dispatch date
        DateTime? dateResive; //Receiving date
        int climbingToFloor; //Delivery to the floor
        int descentFromFloor; //Descent from the floor
        int deliveryScheme; //Delivery scheme
        List<CategoryModel> category; //Enumerations of cargo categories
        List<DopUslugaClassificationModel> dopUslugaClassificator; //Enumarations of add.
services
        decimal? categorySumma;
        decimal? allSumma; //Total shipping cost
        bool status; //Settlement status
        bool denyIssue; //Prohibition of issuance
        bool EconomDelivery; //Economy delivery, flag
        bool EconomPickUp; //Economy pick up, flag
        bool IsGidrobort; //Tail lift, flag
        bool IsOverSize; //Oversized, flag
        bool isPostomat; //Prohibition of issuance, flag
        string comment; //Settlement description
        decimal? SummaryTransportCost; //Warehouse-warehouse shippment cost
        decimal? SummaryDuCost; //Add. Services cost
        decimal? SummaryOformlenieCost; //Formalization cost
        int currency; //Currency
        int viewType;
    }

    class CategoryModel //Cargo category model
    {
        string categoryId; //Cargo category id
        string categoryIdName; //Cargo category name
        int classification;
        int countPlace; //Amount of places
        double? helf; //Weight
        double? size; //Wolume
        double? height; //Height
        double? lenght; //Length
        double? width; //Width
        double? helfTarif; //Tariff per kg
        double? egTarif; //Tariff per unit of cargo
        double? oformlenie; //Formalization cost per place
        double? oformlenieCost; //The total cost of formalization
        double? deliveryCost; //Shipment cost
        double? documentCost;
        string comment; //Settlement progress
    }
```

```
public class DopUslugaClassificationModel //Add. Services category model
{
    int classification; //Category code
    string name; //Category name
    List<DopUslugaModel> dopUsluga; //Enumeration of add. services
}


public class DopUslugaModel //Add. service model
{
    string uslugaId; //Add. service id
    string name; //Add. service name
    decimal? cost; //Add. service name
    int count; //Amount of services
    int classification;
    decimal? minWidth; //Minimal weight
    decimal? maxWidth; //Maximal weight
    decimal summa; //Add. service total cost
    string comment;
    int currency; //Currency
 }
```

3.2 Add. services and their included add. services categories directory output - method GetDopUslugiClassification.

GET api/v4/Public/GetDopUslugiClassification?culture={culture}&currency={currency}&CitySendId={CitySendId}&CityReceiveId={CityReceiveId}&formalization={formalization}

Input parameters

Name	Data type	Default value	Description
culture	String	uk-UA	Culture; Valid values (en-US, uk-UA).
currency	Integer	100000000	Currency code
CitySendId	Guid	*	Dispatch city id
CityReceiveId	Guid	*	Receipt city id.
formalization	Boolean	false	To display add. services for a cost calculation(false) or for a registration (true).

Output parameters

Represents as json. Collection of objects DopUslugaClassificationModel

Output parameters format
application/json, text/json

Example:

```
{
  "status": true,
  "message": "",
  "data": [
    {
      "classification": 100000005,
      "name": "Упаковочные материалы",
      "dopUsluga": [
        {
          "uslugaId": "2b4247c9-be8c-e211-be60-00155d037919",
          "name": "Доупаковка MAXI",
          "cost": 9.00000000,
          "count": 0,
          "classification": 0,
          "minWidth": null,
          "maxWidth": null,
          "summa": 0.0,
          "comment": null
         "currency": 100000001
        },
        {
          "uslugaId": "3e9cde5d-bf8c-e211-be60-00155d037919",
          "name": "Доупаковка MIDI",
          "cost": 6.00000000,
          "count": 0,
          "classification": 0,
          "minWidth": null,
          "maxWidth": null,
          "summa": 0.0,
          "comment": null
           "currency": 100000001
        }
      ]
    }
  ]
}
```

3.3 Tariff categories directory output - method GetTariffCategory.

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

Input parameters

Name	Data type	Default value	Description
culture	String	uk-UA	Culture; Valid values (en-US, uk-UA).

Name	Data type	Default value	Description
CitySendId	Guid	*	Dispatch city.
CityReceiveId	Guid	*	Receipt city.
WarehouseReceiveId	Guid	*	Receipt warehouse.

Output parameters

Represents as json. Collection of objects {id, name, MaxWidth, MaxSize, MinSize, MinWidth, Length, Width, Height, RequiredWeight, RequiredSize}

Id – Tariff category id
name – Tariff category name
MinWidth – Minimal weight
MaxSize – Maximal size
MinSize – Minimal size
MaxWidth – Maximal weight
Length – Length
Width – Width
Height – Height
RequiredWeight – Required weight
RequiredSize – Required wight

Output parameters format
application/json, text/json

Example:

```
{
  "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 Getting cargo categories - method GetCargoCategory.

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

Input parameters

Name	Data type	Default value	Description
TariffCategoryId	Guid	null	Tariff category id.
culture	String	uk-UA	Culture; Valid values (en-US, uk-UA).

Output parameters

Output parameters format

application/json, text/json

Example:

In json format.

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

In xml format.

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

3.5 Delivery schemes directory output - method GetDeliveryScheme.

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

Input parameters

Name	Data type	Default Value	Description
culture	String	uk-UA	Culture; Valid values (en-US, uk-UA).
CitySendId	Guid	*	Dispatch city id.
CityReciveId	Guid	*	Receipt city id.
WarehouseReceiveId	Guid	*	Receipt warehouse id.

Output parameters

Represents as json. Collection of objects {id,name }

Id – Delivery scheme id

name – Delivery scheme name

Output parameters format

application/json, text/json

Example:

```
{
  "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 Cost of transportation calculation - method PostReceiptCalculate.

POST api/v4/Public/PostReceiptCalculate

Input parameters

Name	Data type	Description
input	CalculatorModel (see par. 4.1)	Model that describes input and output parameters of the calculator.

Input parameters format:

```
{
       "culture": "uk-UA", //Culture
       "areasSendId": "4fc948a7-3729-e311-8b0d-00155d037960", //Dispatch city
       "areasResiveId": "e3ac6f68-3529-e311-8b0d-00155d037960", //Arrival city
       "warehouseSendId": "1c828aa6-70c8-e211-9902-00155d037919", //Dispatch warehouse
       "warehouseResiveId": "d908c5e1-b36b-e211-81e9-00155d012a15", //Arrival warehouse
       "InsuranceValue": 1000000, //Cargo insurance cost
       "CashOnDeliveryValue": 5000, //Cash on delivery cost
       "dateSend": "06.06.2014", //Dispatch date
       "deliveryScheme": 2, //Delivery scheme
       "category": [ //Cargo categories array
       {
              "categoryId": "00000000-0000-0000-0000-000000000000", //Cargo category id
              "countPlace": 1, //Amount of places
              "helf": 1, //Cargo weight
              "size": 1 //Cargo size
       }],
       "dopUslugaClassificator": [
       {
              "dopUsluga": [ //Add. services array
              {
                     "uslugaId": "2b4247c9-be8c-e211-be60-00155d037919", //Add. service id
                      "count": 1 //Amount of add. services
              },
              {
                     "uslugaId": "3e9cde5d-bf8c-e211-be60-00155d037919",
                      "count": 5
              }]
       }]
}
```

Output parameters

Output parameters format
application/json, text/json

Example:

```
  {
     "status": true,
     "message": "",
     "data":
     {
       "SummaryTransportCost":230, //Warehouse-warehouse transportation cost
       "SummaryDuCost":99, //Add. services cost
       "SummaryOformlenieCost":1.5, //Formalization cost
       "EconomyVesovojTarifValuta: 16.5, //Saving on weight tariff
       "EconomyEconomTarifValuta": 0, //Saving on econom-tariff
```

```
 "EconomySpecTarifValuta": 0, //Saving on special tariff
 "EconomyIndTarifValuta": 0, //Saving on individual tariff
 "EconomyGlobalDiscountValuta": 0, //Saving on global discount
 "EconomyDiscountCardValuta": 0, //Saving on discount card
 "EconomyAktsiyaValuta": 0, //Saving on direction/warehouse discount
 "EconomyOnAmountSkidka":0, //Saving on discount amount
 "EconomySummary": 16.5, //Total savings on receipt
 "ComissionGM":100, ,//Cash on delivery commission: % from amount + 10 hrn
 "culture":"uk-UA", //Culture
 "calcType":null,
 "areasSendId":"4fc948a7-3729-e311-8b0d-00155d037960", //Dispatch city id
 "areasResiveId":"e3ac6f68-3529-e311-8b0d-00155d037960", //Arrival city id
 "warehouseSendId":"1c828aa6-70c8-e211-9902-00155d037919", //Dispatch warehouse id
 "individualTarifId":null, //Individual tariff id
 "warehouseResiveId":"d908c5e1-b36b-e211-81e9-00155d012a15", //Arrival warehouse id
 "areasSendIdName":"Авдеевка", //Dispatch city name
 "areasResiveIdName":"Артемовск", //Arrival city name
 "warehouseSendIdName":"АВДЕЕВКА", //Dispatch warehouse name
 "warehouseResiveIdName":"АРТЕМОВСК", //Arrival warehouse name
 "InsuranceValue":1000000, //Cargo insurance value
 "InsuranceCost":4000, //Insurance cost
 "dateSend":"01.06.2014", //Dispatch date
 "dateResive":"04.06.2014", //Receiving date
 "deliveryScheme":2, //Delivery scheme
 "category":[ //Categories array
 {
        "categoryId":"00000000-0000-0000-0000-000000000000", //Cargo category id
        "cargoCategoryId":null, //Dispatch cargo category
        "categoryIdName":"Груз", //Cargo category name
        "cargoCategoryIdName":"", //Dispatch cargo name
        "classification":0, //Category code
        "countPlace":1, //Amount of places
        "helf":1, //Weight
        "size":1, //Size
        "height":0, //Height
        "lenght":0, //Length
        "width":0, //Width
        "helfTarif":0.98, //Weight tariff
        "egTarif":230, //Tariff per unit of cargo
        "oformlenie":1.5, //Formalization cost
        "oformlenieCost":1.5, //Formalization total cost
        "deliveryCost":230, //Delivery cost
        "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, //Economy but longer delivery
        "isExpress":false, //Express delivery?
        "isIndividual":null, //Individual tariff?
        "PartnerNumber":null, //Partner’s declaration number
        "weightSummary":1, //Total weight
        "volumeSummary":1 //Total size
}],
"dopUslugaClassificator":[ //Add. services categories array
  {
       "classification":100000005, //Catrgory code
       "name":"Упаковочные материалы", //Category name
       "dopUsluga":[ //Add. services array inside category
       {
              "uslugaId":"2b4247c9-be8c-e211-be60-00155d037919", //Add service id
              "name":"Доупаковка MAXI", //Add. service name
              "cost":9.00000000, //Add. service cost
              "count":1, //Amount of add. services
              "classification":100000005, //Category code
              "minWidth":null, //Minimal weight
```

```
                "maxWidth":null, //Maximal weight
                "summa":9, //Add. service total cost
                "comment":null
            },
            {
                "uslugaId":"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":[
            {
                "uslugaId":"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, //Total delivey cost
        "status":true, //Execution status
        "comment":"Рассчет 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 MAXI-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 //Currency code
    }
}
```

3.7 Getting the cost of insurance - method GetInsuranceCost.

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

Input parameters

Name	Data type	Default value	Description
CitySendId	Guid?	null	Dispatch city id.
CityReceiveId	Guid?	null	Receipt city id.
WarehouseSendId	Guid	*	Dispatch warehouse id.
WarehouseReceiveId	Guid	*	Receipt warehouse id.
InsuranceValue	Double	*	Declared cargo cost.
InsuranceCurrency	Integer	100000000	Insurance payment currency.

Output parameters

Output parameters format
application/json, text/json

Represents as json. Collection of objects {id,name }

Value – Insurance value

MinValue – Declared cargo minimal cost for the direction

Example:

```
In json format.
{
  "Value": 40.0,
  "MinValue": 10000.0,
  "status": true,
  "message": null
}
```

4. Communication with the user

4.1 Data models for the information exchange:

```
public class RateServicesModel
{
        Guid OfficeId; //Reresentative id
        int WarehosePlacing; //Warehouse location
        int CargoReceiveSpeed; //Cargo receiving speed
        int CargoOutputSpeed; //Cargo output speed
        int DocumentsIssuanceSpeed; //Documents formalization speed
        int DeliverySpeed; //Cargo delivery speed
        int TarrifsRate; //Transportation tariff (warehouse-warehouse)
        int CargoLoadTarrifs; //Cargo loading-delivery tariffs
        int WorkersCulture; //Warehouse service workers culture
        int QualityInGeneral; //Branch services quality in general
        string YourRecomendations; //Your wishes and recommendations
        string ClientNumber; //Client’s barcode
        string Name; //Client’s name
        string LastName; //Client’s last name
        string SecondName; //Client’s middle name
        string Phone; //Client’s phone number
        string Email; //Client’s email
        string CompanyName; //Company name
 }


 class PickUpCargoModel
 {
        string ContactName; //The contact person
        string Name; //Organization name/Full name
        string PhoneNumber; //Phone number
        string Email; //Email
        string Area; //Area
        string City; //City
        string Address; //Address
        string AccessMode; //The presence of an access mode
        int? Weight; //Cargo weight
        int? Size; //Cargo size
        int? Quantity; //Amount of places
        string Date; //Loading date
        string Time; //Desired time
        string Note; //Note
        bool? IsFloor; //Descent from the floor?
        string Floor; //Floor
        sting ToCity; //Receipt city
 }

 class ContactsMessageModel
 {
        string ReceiptNumber; //Receipt number
        string Name; //Full name
        string Phone; //Phone number
        string Email; //Email
        string Subject; //Subject of message
        Guid? Warehouse //Warehouse id
        string Message; //Message
        string CategoryName; //Message category name
 }
```

4.2 Getting company news - method GetNews.

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

Input parameters

Name	Data type	Default value	Description
culture	String	uk-UA	Culture; Valid values (en-US, uk-UA).
count	Integer	1	Amount of returning records on page
page	Integer	1	News page number.

Output parameters

Represents as json. Collection of objects {NewsItemId, Title, ShortContent, Content, PublishDate, ImageName, ImageUrl, ImageContent, WarehousesId}.

NewsItemId – News id
Title – News title
ShortContent – Brief content of the news
Content – Content of the news
PublishDate – Publication date
ImageName – Image name
ImageUrl – Image link
ImageContent – Image file content
WarehousesId – Warehouse id

Output parameters format
application/json, text/json

Example:

```
{
  "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;будет сообщено 
дополнительно.</span></p>\r\n",
      "PublishDate": "2014-06-27T00:00:00",
      "ImageName": null,
      "ImageUrl": null,
      "ImageContent": null
      "WarehousesId": null
    },
  ]
}
```

4.3 Getting a message subject - method GetMessagesTheme.

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

Input parameters

Name	Data type	Default value	Description
culture	String	uk-UA	Culture; Valid values (en-US, uk-UA).

Output parameters

Represents as json. Collection of objects {Id, Name}.

Id – Subject id

Name – Subject name

Output parameters format

application/json, text/json

Example:

```
{
  "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 Submitting an assessment of the company's performance - method PostServiceRate.

POST api/v4/Public/PostServiceRate

Input parameters

Name	Data type	Description
input	RateServicesModel	Model that describes input and output parameters

Example of input parameters:

```
{
  "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.com",
  "CompanyName": "test"
}
```

Output parameters

Output parameters format

application/json, text/json

Example:

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

4.5 Sending a vehicle order - method PostPickUpCargo.

POST api/v4/Public/PostPickUpCargo

Input parameters

Name	Data type	Description
input	PickUpCargoModel	Model that describes input and output parameters

Example of input parameters:

```
{
  "ContactName": "contact name",
  "Name": "name",
  "PhoneNumber": "123456",
  "Email": "name@name.com",
  "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"
}
```

Output parameters

Output parameters format
application/json, text/json

Example:

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

5. User area, register

5.1 Login – method PostLogin

POST api/v4/Public/PostLogin

Input parameters

Name	Data type	Description
model	LoginModel	Model that describes input and output parameters

Example of input parameters:

```
{
     “UserName”: "iv.iv.ivankov@gmail.com", //User’s login
     “Password”: "123456", //User’s password
     “RememberMe”: true //Remember me
};
```

Output parameters

Output parameters format

application/json, text/json

Example:

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

5.2 Logout – method PostLogoff

POST api/v4/Public/PostLogoff

Method requires authorization

Input parameters

```
Missing
```

Output parameters

Output parameters format
application/json, text/json

Example:

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

5.3 Getting am information about user – method GetUserInfo.

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

Method requires authorization

Input parameters

Name	Data type	Default value	Description
culture	String	uk-UA	Culture; Valid values (en-US, uk-UA).

Output parameters

Represents as json. Object {Id, AccessLevel, UserName, SmsPhoneNumber, ClientTipe, ClientNumber, PhoneNumber, CrmUserId, Email, showHelpHide, Photo, RoleName, IsLoyaltyProgram, AvailablePoints, CurrentPoints, City, ConfirmPhine}.

Id – Client id in website database
AccessLevel – Level of access
UserName – User name
SmsPhoneNumber – User SMS phone number
ClientTipe – Client type (Natural person(false)/Legal entity(true))
ClientNumber – Client barcode
PhoneNumber – User phone number
CrmUserId – Client id in CRM
Email – Email address
RoleName – User role
IsLoyaltyProgram – Is registrated in loyality program?
AvailablePoints – Loyality program availabe points
CurrentPoints – Loyality program current points
City – City (Client location)
ConfirmPhine – Confirmed user phone number (to enter the website)

Output parameters format
application/json, text/json

Example:

```
{
  "status": true,
  "message": "",
  "data": {
    "Id": "45023",
    "AccessLevel": "Полный доступ",
    "UserName": "!! Тестовый клієнт для сайту",
    "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 Getting user receipts – method GetUserReceipt.

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

Method requires authorization

Input parameters

Name	Data type	Default value	Description
page	Integer	1	Receipt page
rows	Integer	10	Number of displayed lines
type	Integer	0	Receipt type (0-sending, 1-receiving)
culture	String	uk-UA	Culture; Valid values (en-US, uk-UA).
detail	Boolean	true	Is need to return information about add. services and potential recipients.

Output parameters

Represents as json. Object { id, number, SendDate, ReceiveDate, SenderWarehouseName, RecepientWarehouseName, TotalCost, Status, StatusesDecoding, Weight, Volume, PaymentStatus, Currency, CanChangeRecepient, LockShipping, IsPrivate, IsAllowDeny, Sender, Recepient, Payer, StatedValue, Sites, PriceWarehouseWarehouse, AuxServicesList, InsuranceCost, InsuranceValue, PossibleReceivers, codCost, codCurrency, codName, codWarehouse, isGiveMoney, SafetyDealMoneyStatus }.

id – Receipt id,
number – Receipt number,
total – Amount of pages with receipts,
SendDate – Date the receipt was sent,
ReceiveDate – Date the receipt was received,
SenderWarehouseName – Warehouse sending receipt,
RecepientWarehouseName – Warehouse receiving receipt,
Status – Receipt status,
StatusesDecoding – Receipt text status,
TotalCost – Total cost,
PartnerNumber – Partner declaration number,
Type – Receipt type (see. directory 8.4),
Weight – Weight,
Volume – Volume,
PaymentStatus – Payment status,
Currency – Currency,
CanChangeRecepient – Is it possible to change sender,
LockShipping – Prohibition on issuance,
IsPrivate – Is closed for view,
IsAllowDeny – Is allowed for removing or establishing on issuance prohibition,
Sender – Sender name,
Recepient – Recepient name,
Payer – Payer name,
StatedValue – Declared value,
Sites – Amount of sites,
PriceWarehouseWarehouse – Shipping cost without discount and add. services,
AuxServicesList – List of add. services,
InsuranceCost – Insurance cost,
InsuranceCurrency – Insurance currency,
PossibleReceivers – List of possible recepients,
PushStateCode – Sanding state
codCost – Declared cargo cost,
codCurrency – Cash on delivery currency,
SenderPhone – Sender phone number (null if not an express delivery)
ReceiverPhone – Recepient phone number (null if not an express delivery)
AddressPickup – Express delivery shipping address (null if not an express delivery)
AddressDelivery – Express delivery load address
DateArrivalExpress – Express delivery date and time
CitySendName – Dispatch city
DeliveryType – Delivery scheme
codName – Cash on delivery recepient name,
codWarehouse – Cash on delivery warehouse name,
isGiveMoney – Are cash on delivery funds issued to recepient
codGiveMoneyDate – Date of issuing cash on delivery to sender
SafetyDealMoneyStatus – Safe deal funds status

Output parameters format

application/json, text/json

Example:

```
  {
    "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",
        "RecepientWarehouseName": "КИЕВ-1",
        "Status": "8",
        "StatusesDecoding": "Зарезервирована",
        "TotalCost": 278.000,
        "PartnerNumber": "",
        "Weight": 100.000,
        "Volume": 1.000000,
        "PaymentStatus": false,
        "Currency": 100000000,
        "CanChangeRecepient": false,
        "LockShipping": false,
        "IsPrivate": 0,
        "IsAllowDeny": true,
        "Sender": "!! Іванков Іван Тест",
        "Recepient": "!! 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. Making a receipt

6.1 Getting access via API key and data format selection.

To use these methods that require authorization through the API, you must additionally pass the api key. The user is given a pair of public and secret api keys. For authorization, the user needs to pass in the request header in the “HMACAuthorization” parameter the public key, the current server time and the hash code generated by encryption using the HmacSHA1 algorithm. Usage examples are given below.

Javascript example.

```
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
 });
```

C# example.

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

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

```
            return (sbinary);
        }

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

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

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

By default, data is output in json format. To change the type of input and output data to xml, you must pass the type=xml parameter to the address bar. Example //api/v3/Public/GetClientCards?type=xml.

6.2 Getting a list of client payment cards – method GetClientCards.

GET api/v4/Public/GetClientCards

Method requires authorization via API key

Input parameters

Missing

Output parameters

Represents as list of objects. Object { id, name }.

id – card Id

name – card name

Output parameters format

application/json, text/json

Example:

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

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

6.3 Getting a list of client settlement accounts – method GetClientInvoices.

GET api/v4/Public/GetClientInvoices

Method requires authorization via API key

Input parameters

Missing

Output parameters

Represents as list of objects. Object { id, name }.

id – Card id

name – Card number

Output parameters format
application/json, text/json

Example:

```
In json format:
{
  "data": [
    {
      "id": "f38964d0-f2a5-e411-b119-000d3a200160",
      "name": "7777777777777777777"
    }
  ],
  "status": true,
  "message": ""
}

In xml format:
<ApiResult>
        <status>true</status>
        <message/>
        <data>
                <DirectoryItem>
                        <id>f38964d0-f2a5-e411-b119-000d3a200160</id>
                        <name>7777777777777777777</name>
                </DirectoryItem>
        </data>
</ApiResult>
```

6.4 Making a receipt- method PostCreateReceipts.

POST api/v4/Public/PostCreateReceipts

Method requires authorization via API key

The input parameters are an array of RegistrationReceiptsModel models with the following fields

Name	Data type	Is required? (Yes/No)	Description
deliveryScheme	Integer	By default: 0	Delivery scheme 0: Warehouse-warehouse, 1: Address-Address, 2: Warehouse-doors, 3: Address-Warehouse
areasSendId	Guid	Yes	Dispatch city Guid
warehouseSendId	Guid	Yes, if deliveryScheme = 0 or 2	Guid of the warehouse in the dispatch city

Name	Data type	Is required? (Yes/No)	Description
pickUpContactName	String	Yes, if deliveryScheme = 1 or 3	Contact person for pick up
pickUpContactPhone	String	Yes, if deliveryScheme = 1 or 3	Phone number for pick up (10 digits. E.g.: “0501112233”)
pickUpAddressId	Guid	One of these fields must be filled in if deliveryScheme = 1 or 3	Pick up address Guid
pickUpAddress	String		Pick up address (separated by commas: Index, street, house, apartment)
pickUpDate	String	Always takes the value of the receipt creation date. Used when deliveryScheme = 1 or 3	Pick up date
areasResiveId	Guid	Yes	Guid of the receipt city
warehouseResiveId	Guid	Yes, if deliveryScheme = 0 or 3	Guid of the warehouse in the receipt city
deliveryContactName	String	Yes, if deliveryScheme = 1 or 2	Contact person for delivery
deliveryContactPhone	String	Yes, if deliveryScheme = 1 or 2	Phone number for delivery (10 digits. E.g.: “0501112233”)
DeliveryComment	String	No	Delivery comment
deiveryAddresId	Guid	One of these fields must be filled in if deliveryScheme = 1 or 2	Delivery address Guid
deliveryAddress	String		Delivery address (separated by commas: Index, street, house, apartment)

Name		Data type	Is required? (Yes/No)	Description
SenderId		Guid	By default: Guid of API key owner (public key)	Sender's Guid. Can take the value of your Id or the Id of your parent or subsidiary organization.
posibleResiverReceipt_1		Guid	Yes, or create a new one using the fields (1)	Guid of an existing recipient.
1)	receiverName		Yes, when creating a recipient	Recipient’s Full name (individual)
	receiverType		By default: **false**	false: individual, true: entity
	receiverPhone		No, for creating individuals Yes, for creating legal entities	Recipient's phone number (10 digits. E.g.: “0501112233”)
	receiverEgrpo		No, for creating individuals Yes, for creating legal entities	USREO
posibleResiverReceipt_2 posibleResiverReceipt_3 posibleResiverReceipt_4		Guid	No	Guid of a possible recipient.
dateSend		String	Yes	Dispatch date
Currency		Integer	By default: 100000000	Currency code
payerType		Integer	By default: 0	Payer type: 0: sender, 1: recipient
paymentType		Integer	By default: 0	Payment type: 0: cash, 1: non-cash.

Name	Data type	Is required? (Yes/No)	Description
payerId	Guid	By default: 0	If the payer needs to transfer a third party, then transfer the Guid of the third party.
paymentTypeInsuranse	Integer	By default: 0	Type of insurance payment: 0: cash, 1: non-cash
payerInsuranceId	Integer	If payerType = 0 – By default senderId. If payerType = 1 – generates, if **null**	Insurance payer
InsuranceValue	Integer	By default: 0	Insured value of cargo
CashOnDeliveryPayerAccountId	Guid	No	Cash on delivery payer
cashOnDeliveryValue	Ineger	No	Cash on delivery amount
cashOnDeliveryValuta	Integer	By default: 100000000	Cash on delivery currency
cashOnDeliveryType	Integer	By default: 0	Cash on delivery type 0: payment card, 1: settlement account, 2: cash, 3: safe transaction
CashOnDeliverySafetyDeal	Boolean	Optional, use with cashOnDeliveryType = 3	Flag: safe deal?
CashOnDeliveryWarehouseId	Guid	Required if cashOnDeliveryValue > 0	Cash on delivery payment warehouse Guid
CashOnDeliveryRasschSchetId	Guid	Required if cashOnDeliveryValue > 0 and cashOnDeliveryType = 1 or 3	Settlement account id

Name	Data type	Is required? (Yes/No)	Description
CashOnDeliveryCardId	Guid	Required if cashOnDeliveryValue > 0 и cashOnDeliveryType = 0	Payment card id
CashOnDeliverySenderPhone	String	Required if cashOnDeliveryValue > 0 и cashOnDeliveryType = 2	Cash on delivery recepient’s phone number
CashOnDeliverySenderFullName	String	Required if cashOnDeliveryValue > 0 and cashOnDeliveryType = 2 and if the recipient is a legal entity	Cash on delivery recepient’s full name
CashOnDeliveryReceiverPhone	String	Yes, if cashOnDeliveryValue>0 and cashOnDeliveryType = 2 and If the payer is a legal entity	Cash on delivery payer's phone number
CashOnDeliveryReceiverFullName	String	Yes, if cashOnDeliveryValue>0 and cashOnDeliveryType = 2 and If the payer is a legal entity	Cash on delivery payer's full name
ReturnDocuments	Boolean	Flag return of documents on delivery	No
descentFromFloor	Integer	Descent from the floor	No
climbingToFloor	Integer	Rise to the floor	No
IsOverSize	Boolean	Oversized, for cargo delivery	No
IsGidrobort	Boolean	Tail lift, for cargo delivery	No
EconomDelivery	Boolean	Economy delivery	No
EconomPickUp	Boolean	Economy pick up, for cargo pick up	No

Name	Data type	Is required? (Yes/No)	Description
ExpressPickUp	Boolean	Express pick up, for cargo pick up	No
parentNumber	String	Partner declaration number	No
DeliveryComment	String	Delivery comment	No
category	Category Model[]	Cargo array	Yes

The sender is the client that authorized before calling the method. SenderId can take the value of your Id (public key) or the Id of your parent or subsidiary organization.

If some of the required fields are not filled in, the method will return a warning and the receipt will not be created.

When debugging modules for creating receipts, it is recommended to use **debugMode.**

To determine the receipt recipient, you must either specify its Id or create a new one.
If an already existing one is specified, then its Id is passed in the parameter **posibleResiverReceipt_1.**
Three parameters are used to create a new recipient: ***receiverName, receiverType*** and ***receiverPhone***.

The situation is similar with the fields of the delivery address and the pick up address. You can create a new pick up and delivery address by passing them in text format to the fields ***pickUpAdress*** and ***deliveryAddress*** accordingly. Or by passing their ***Id*** to the fields ***pickUpAdressId*** and ***deliveryAddressId***.

If fields **posibleResiverReceipt_1,** ***pickUpAdressId*** or ***deliveryAddressId*** are specified, then the fields for creating a new **recepient**, ***pick up address*** or ***delivery address*** (respectively) are not taken into account.

As an array, you can send receipts combined with one pick up order, that is, with the same dispatch city, address, and dispatch date.

Input data example:

```
{
      "culture": "uk-UK", //Culture
      "flSave": "true", //Save flag
      "debugMode": "false", //Debug mode flag
      "receiptsList": [
        {
            "areasSendId": "f6ee49fa-3e29-e311-8b0d-00155d037960", //Dispatch city
            "areasResiveId": "ebc7639a-db2a-e311-8b0d-00155d037960", //Receipt city
            "warehouseSendId": "6b3b6d45-b249-e211-ab75-00155d012d0d", //Dispatch
warehouse
            "warehouseResiveId": "ab3b6d45-b249-e211-ab75-00155d012d0d", // Receipt
warehouse
```

```
                         "dateSend": "2015-06-24T00:00:00", //Dispatch date
                         "deliveryScheme": 1, //Delivery scheme 0- Warehouse-Warehouse, 1- Address-
adress, 2- Warehouse-address, 3- Address-warehouse
                         "receiverName": "Иванков Иван Иванович", // Recipient; for an individual -
full name
                         "receiverPhone": "0500000000", //Recipient phone number
                         "receiverType": false, //false – individual, true – legal entity
                         "currency": 100000000, //Receipt currency
                         "InsuranceValue": 10000.0, // Insured value of cargo
                          "senderId": "cdbfe2d5-bf02-4c0d-b7d6-5cf277761c50", //Sender Id
                          "payerInsuranceId": "1aa70d22-1209-e511-b3b5-000d3a200160", // insurance payer
                          "payerId": "1aa70d22-1209-e511-b3b5-000d3a200160", // if the payer needs to
                    transfer a third

                         "payerType": 1, // Payer type 0-sender, 1-recipient
                         "paymentType": 0, //Payment type 1- non-cash, 0- cash
                         "paymentTypeInsuranse": 0, //Insurance payment type 1- non-cash, 0- cash

                         "deliveryAddress": "Науки, 5", //Delivery address
                         "deliveryContactName": "Дмитрий", //Contact person for delivery
                         "deliveryContactPhone": "0500000000", //Phone number for delivery
                         "DeliveryComment": "Комментарий при доставке", //Comment on delivery
                         "ReturnDocuments": true, //Flag return of documents on delivery
                         "climbingToFloor": 4, //Rise to the floor
                          "EconomDelivery": false, //Economy delivery
                          "IsOverSize": false, //Oversized, for cargo delivery
                          "IsGidrobort": false, //Tail lift, for cargo delivery
                          "EconomPickUp": false, //Economy pick up, for cargo pick up
                           "ExpressPickUp": false, //Express pick up, for cargo pick up
                         "cashOnDeliveryType": 0, //Cash no delivety type 0-payment card, 1-settlement
account, 2-cash, 3-Safe deal
                         "CashOnDeliveryValuta": 100000000, //Cash on delivery currency
                         "CashOnDeliveryValue": 1000.0, // Cash on delivery amount
                         "CashOnDeliveryCardId": "B08ACA89-B6C8-4014-ABF8-3EA61B18E5DA", //Payment card
Id
                          "CashOnDeliveryPayerAccountId": "1aa70d22-1209-e511-b3b5-000d3a200160" //Cash
                    on delivery payer
                         "CashOnDeliveryRasschSchetId": "00000000-0000-0000-0000-000000000000", //
Settlement account ID (this field is mandatory for Safe transaction)
"CashOnDeliveryDescription": "описание", // Description of the payment
                        "CashOnDeliveryWarehouseId": "6b3b6d45-b249-e211-ab75-00155d012d0d", // Cash on
                    delivery payment warehouse
                        "CashOnDeliverySenderFullName": "Иванов Иван Иванович",//Cash on delivery
                    recipient’s full name
                        "CashOnDeliverySenderPhone": "0501234567",//Cash on delivery recipient’s phone
                    number
                        "CashOnDeliveryReceiverFullName": "Петров Петр Петрович",//Cash on delivery
                    sender’s full name
                        "CashOnDeliveryReceiverPhone": "0671234567",//Cash on delivery sender’s phone
                    number
                         "pickUpDate": "2015-06-30T00:00:00", //Pick up date
                         "pickUpContactName": "Василий", //Contact person for pick up
                         "pickUpContactPhone": "0500000000", //Pick up contact phone number
                         "pickUpAddressId": "a5eaf714-fb60-e411-b421-000d3a200936", //Pick up address
                         "descentFromFloor": 4, //Descent from the floor
                         "category": [
                           {
                               "categoryId": "00000000-0000-0000-0000-000000000000", //Tariff Category
                               "cargoCategoryId": "0307d03b-9e36-e311-8b0d-00155d037960", // Category
of the shipped cargo
                               "countPlace": 10, //Number of places
                               "helf": 100.0, //Weight
                               "size": 2.0, //Size
                               "isEconom": true, //Economical but longer delivery
                               "PartnerNumber": "123456" //Partner declaration number
                           },
                           {
                               "categoryId": "00000000-0000-0000-0000-000000000000",
                               "cargoCategoryId": "0f07d03b-9e36-e311-8b0d-00155d037960",
                               "countPlace": 10,
                               "helf": null,
                               "size": null,
                               "isEconom": true,
                               "PartnerNumber": "123457"
                           }
                         ]
                     },
                     {
                         "areasSendId": "f6ee49fa-3e29-e311-8b0d-00155d037960", //Dispatch city
                         "areasResiveId": "4577d856-322b-e311-8b0d-00155d037960", //Recipient city
                         "warehouseSendId": "6b3b6d45-b249-e211-ab75-00155d012d0d", //Dispatch
warehouse
                         "warehouseResiveId": "efbecb4b-da49-e211-9515-00155d012d0d", //Recipient
warehouse
                         "dateSend": "2015-06-24T00:00:00", //Dispatch date
                         "deliveryScheme": 3, //Delivery scheme 0- Warehouse-Warehouse, 1- Address-
            Adddress, 2- Warehouse-address, 3- Address-warehouse
                         "posibleResiverReceipt_1": "07b98959-52ab-40a0-9ce7-ab7ee678d809",//Recipient
       Id
                     "posibleResiverReceipt_2": "07b98959-52ab-40a0-9ce7-ab7ee678d809",//Possible
                     recipient Id
                     "posibleResiverReceipt_3": "07b98959-52ab-40a0-9ce7-ab7ee678d809",//Possible
                     recipient Id
                     "posibleResiverReceipt_4": "07b98959-52ab-40a0-9ce7-ab7ee678d809",//Possible
                     recipient Id

                         "currency": 100000000, //Receipt currency
                         "InsuranceValue": 10000.0, //Insured value of cargo

                         "payerType": 0, //Payer type 0-sender, 1-recipient
                         "paymentType": 0, // Payment type 1- non-cash, 0- cash
                         "paymentTypeInsuranse": 0, //Insurance payment type 1- non-cash, 0- cash

                         "pickUpDate": "2015-06-30T00:00:00", //Pick up date
                         "pickUpContactName": "Василий", //Contact person for pick up
                         "pickUpContactPhone": "0500000000", // Pick up contact phone number
                         "pickUpAddressId": "a5eaf714-fb60-e411-b421-000d3a200936", //Pick up address
                         "descentFromFloor": 4, //Descent from the floor
                         "category": [
                           {
                               "categoryId": "00000000-0000-0000-0000-000000000000", //Tariff Category
                               "cargoCategoryId": "0307d03b-9e36-e311-8b0d-00155d037960", //Category of
       the shipped cargo
                               "countPlace": 1, //Amount of places
                               "helf": 20.0, //Weight
                               "size": 1.0, //Size
                               "isEconom": true,
                               "PartnerNumber": "1234567"
                           }
                         ]
                     }
                   ]
               }
```

Output parameters

Represents as json. Collection of objects {Id, Number, TotallCost, InsuranceCost, ComissionGM, Comment, egs}.

Id – Receipt Guid

Number – Receipt number

TotallCost – Shipping cost

InsuranceCost – Cost of insurance

ComissionGM – Cash on delivery commission (% from the sum +10hrn for cash on delivery service)

Comment – Comment

egs – Array of cargo units

Output parameters fomat

application/json, text/json

Example:

```
{
  "status": true,
  "message": [],
  "receipts": [
    {
      "Id": "f5a947f6-adcf-49e8-be46-a49d69621ae2",
      "Number": "9900000000",
      "TotallCost": 97.0,
      "InsuranceCost": 7.0,
      "ComissionGM": 30.0,
      "Comment": "",
      "egs": [
        {
          "Id": "3fed9940-b094-4236-a3b0-728a83123eca",
          "PartnerNumber": null,
          "Number": "9900000000002002151"
        }
      ]
    }
  ]
}
```

In xml format.

```
<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 *Deactivation of cargo units - method PostDeactivateEg.*

POST api/v4/Public/PostDeactivateEg

Method requires authorization via API key

Input parameters

Name	Data type	Description
input	ReceiptsOutputModel	Model that describes input and output parameters

Input data example:

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

Output parameters

Output parameters format
application/json, text/json

Example:

In json format.

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

In xml format.

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

6.6 Getting documents in PDF - method GetPdfDocument.

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

Method requires authorization via API key

Input parameters

Name	Data type	Default value	Description
number	String	*	Receipt numbers (numbers are indicated separated by semicolons without spaces, e.g., number=9900112233;9900223344)
type	Integer	*	Document type: 0 - Receipt printing, 1 - Printing stickers of units of cargo on the godex, 2 - Printing stickers of cargo units on one sheet, 4 - Printing on one sheet 95x95 of several receipts.

Output parameters

Output parameters format
application/json, text/json

Example:

```
In json format.
{
  "status": true,
  "message": "",
  "file": "EKJRLJFSHDWEKNLVSJDIFJS"
}

In xml format.
<ApiResultFile>
        <status>true</status>
        <message/>
        <file>EKJRLJFSHDWEKNLVSJDIFJS</file>
</ApiResultFile>
* Значение поля file возвращает код зашифрованный в base64
```

6.7 Getting a list of senders (client subsidiary or parent organizations) - method GetSenderList.

GET api/v4/Public/GetSenderList

Method requires authorization via API key

Input parameters
Missing.

Output parameters

Represent as json. Collection of objects {id, name, cityId, cityName}.

Id – Sender Id

name – Sender name

cityId – Sender city Guid

cityName – Sender city name

Output parameters format

application/json, text/json

Example:

In json format.

```
{
  "data":[
    {
      "id":"cdbfe2d5-bf02-4c0d-b7d6-5cf277761c50",
      "name":"!! Іванков Іван 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":""
}
```

In xml format.

```
<ApiResultFile>
        <status>true</status>
        <message/>
        data>
                <DirectoryItem>
                        <id>cdbfe2d5-bf02-4c0d-b7d6-5cf277761c50</id>
                        <name>!! Іванков Іван 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>
</ApiResultFile>
```

6.8 Getting available currencies - method GetCurrency.

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

Input parameter

Name	Data type	Default value	Description
CitySendId	Guid	*	Dispatch city Id.
CityReceiveId	Guid	*	Recipient city Id.
PayerType	Integer	0	Payer type 0 - sender, 1 - recipient, 2 - third party.
PayerId	Guid?	null	Payer Id.
culture	String	uk-UA	Culture; Valid values (en-US, uk-UA).

Output parameters

Represent as json. Collection of objects {id, name}.

id – Currency Id

name – Currency Id

Output parameters format

application/json, text/json

Example:

In json format.

```
{
  "data": [
    {
      "id": "100000000",
      "name": "Гривня"
    }
  ],
  "status": true,
  "message": ""
}
```

In xml format.

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

6.9 Getting a list of the payers - method GetAvailableServices.

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

Input parameters

Name	Data type	Default values	Description
scheme	Integer	*	Delivery scheme (0 - Warehouse-Warehouse, 1 - Address-Address, 2 - Warehouse-Address, 3 -Address-Warehouse)
receiveWarehouseId	Guid	*	Recipient warehouse Id
CodValue	Decimal?	null	Cash on delivery amount

Output parameters

Represents as json. Collection of objects {id, name}.

isInfo –AS Information service is available
isReturnDocs – AS Return issuance of cargo is available
isDenyIssue – AS Prohibition of issuance of cargo is available
isDescent – AS Descent from the floor is available
isLifting – AS Rise to the floor is available
isAutoReturn – AS Autoreturn service is available

Output parameters format
application/json, text/json

Example:

```
  In json format.
  {
    "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 Подъём на этаж
        "isAutoReturn": true //доступна ДУ Услуга автовозврата

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

6.10 *Getting a list of the payers - method GetPayer.*

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

Method requires authorization via API key

Input parameters

Name	Data type	Default value	Description
CitySendId	Guid	*	Dispatch city Id.
CityReceiveId	Guid	*	Recipient city Id.
ClientSenderId	Guid	*	Client sender Id.
ClientReceiverId	Guid?	null	Client recipient Id.
PayerType	Integer?	null	Payer type 0 - sender, 1 - recipient, 2 - third party.

Output parameters

Represents as json. Collection of objects {id, name}.

id – Payer Id

name – Payer name

Output parameters format

application/json, text/json

Example:

In json format:
```
{
  "data":[
    {
      "id":"c11d0fff-b75d-e411-b4c0-000d3a200936",
      "name":"!! Иванков Получатель 99 Авдеевка"
    },
    {
      "id":"cdbfe2d5-bf02-4c0d-b7d6-5cf277761c50",
      "name":"!! Іванков Іван Тест"
    }
  ],
  "status":true,
  "message":""
}
```

In xml format.
```
<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>!! Іванков Іван Тест</name>
          </DirectoryItem>
    </data>
</ApiResult>
```

6.11 Getting client addresses - method GetClientAddress.

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

Input parameters

Name	Data type	Default value	Description
CityId	Guid	*	City Id.
ClientId	Guid	*	Client Id.

Output parameters

Represents as json. Collection of objects {id, name}.

id – Address Guid

name – Address

Output parameters format

application/json, text/json

Example:

```
In json format:
{
  "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": ""
}

In xml format:
<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 *Getting possible client recipients - method GetPosibleReciver.*

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

Method requires authorization via API key

Input parameters

Name	Data type	Default value	Description
CityReceiveId	Guid	*	Receipt city Id.
ClientSenderId	Guid	*	Sender client Id.

Output parameters

Represents as json. Collection of objects {id, name}.

id – Recipient Guid

name – Recipient name

Output parameters format
application/json, text/json

Example:

In json format:
```
{
  "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":""
}
```

In xml format:
```
<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 Getting client payment type - method GetClientPaymentType.

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

Input parameters

Name	Data type	Default value	Description
ClientId	Guid	*	Client Id.

Output parameters

Represents as json.

data – client payment type (bool). True = cash, False = non-cash

Output parameters format
application/json, text/json

Example:

```
In json format:
{
  "data": true,
  "status": true,
  "message": ""
}

In xml format:
<ApiResultBool>
    <status>true</status>
    <message/>
    <data>true</data>
</ApiResultBool>
```

6.14 Getting full information about the receipt - method GetFullReceiptInformation.

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

The method requires authorization with a username and password

Input parameters

Name	Data type	Default value	Description
culture	String	uk-UA	Culture; Valid values (en-US, uk-UA).
number	String	*	Receip number

Output parameters

Represents as json. Collection of objects {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 – Dispatch city Guid
areasSend – Dispatch city
areasReceiveId – Recipiet city Guid
areasResive – Recipient city
warehouseSendId – Dispatch warehouse Guid
warhouseSend – Dispatch warehouse
warehouseReceiveId – Recipient warehouse Guid
warhouseReceive – Recipient warehouse
deliveryScheme – delivery scheme: 0 - warehouse-warehouse, 1 - address-address, 2 - warehouse-address, 3 - address-warehouse
number – Receipt number
sender – Sender name
senderId – Sender Guid
receiver – Recipient name
receiverId – Recipient Guid
payer – Payer name
payerId – Payer Guid
paymentType – payment type: 0 - cash, 1 – non-cash
dateSend – Dispatch date
dateReceive – Arrival date
state – receit status (see directory 8.1)
Currency – Currency code
partnerNumber – partner declaration number
paymentStatus – payment status
paymentDate – payment date
lockShipping – ban on issuance
totalCountPlace – total number of parcels
totalWeight – total weight
totalSize – total size
warehouseWarehouseAmount – transportation price without discounts and additional services
discountAmount – discount amount
lossesDescountAmount – discount loss amount
totalAmount – total receipt value
insuranceValue – insurance value of the cargo
SafetyDealMoneyStatus – Safety Deal funds status
duArray – list of additional services
possibleReceiverArray – list of possible recipients
egArray – list of cargo units

receiptsArray – list of related receipts

[

Number – receipt number,

state – receipt status,

receiptType – receipt type,

paymentType – Payment type 1- non-cash, 0- cash,

paymentStatus – Flag, payment status,

paymentDate – payment date,

currency – currency code,

payerId – payer id,

payer – payer name,

totalAmount – total receipt value,

clientCardId – client card id,

clientCard – abbreviated customer card number,

codSender – Sender name,

codSenderPhone – Sender phone number,

isGiveMoney – whether the money was issued to the recipient of the cash on delivery,

codWarehouse – warehouse of dispatch/receipt of cash on delivery,

codCity – City of dispatch/receipt of cash on delivery,

codGiveMoneyDate – date of issue of cash on delivery to the sender

]

Output parameters format

application/json, text/json

Example:

In json format:

```
{
  "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": "!! Іванков Іван 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": "Таможеннные 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": "!! Іванков Іван Тест",
        "totalAmount": 40.000,
        "clientCardId": null,
        "clientCard": "",
        "codSender": "",
        "codSenderPhone": "",
        "isGiveMoney": false,
        "codWarehouse": "",
        "codCity": ""
      }
    ],
    "egArray": [
      {
        "cargoCregoryId": "0656bab4-e62c-e411-bd10-000d3a200936",
        "cargoCregory": "Автоаксессуары",
        "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 Creating an address or a recipient - method PostCreateAddressOrClient.

POST api/v4/Public/PostCreateAddressOrClient

Method requires authorization via API key

Input parameters

Name	Data type	Description
input	ClientModel	Model that describes input parameters

Input data example:

```
{ //Model for creating a possible recipient
    "AccountId": "",
    "ClientType": "false", //Client type false-physical. person true– legal entity
    "Name":"", //Name of organization (legal entity)
    "SecondName": "Тестовый", //Last name
    "FirstName": "Клиент", //First name
    "LastName": "ДляСайта", //Middle name
    "CityId": "16617df3-a42a-e311-8b0d-00155d037960", //City Id
    "Egrpo":"", //Client USREO
    "PhoneNumber":"0509996665", //Client phone number
    "Street":"ул. Щорса", //Street
    "House":"17", //House
    "Appartament":"3", // Apartment
    "senderId": "c11d0fff-b75d-e411-b4c0-000d3a200936" //Sender Id
}

{ //Model for creating an address
    "AccountId": "1541a45b-1a56-e511-89e5-000d3a200160", //Client Id
    "CityId": "16617df3-a42a-e311-8b0d-00155d037960", //City Id
    "Street": "ул. Васильковская", //Street
    "House": "17", //House
    "Appartament": "3", //Apartment
    "senderId": "c11d0fff-b75d-e411-b4c0-000d3a200936" //Sender Id
}
```

Output parameters

Output parameters format

application/json, text/json

Example:

In json format:

```
{
  "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",
      "Egrpo": "",
      "Inn": "",
      "Kpp": "",
      "OwnershipCode": 100000066,
      "PhoneNumber": "0509996665",
      "SmsPhoneNumber": "0509996665",
      "ParentAccountId": null,
      "ParentAccountName": "",
      "StateCode": 0,
      "CountryCode": "38",
      "MasterId": null
    }
  }
}
```

6.16 Getting information from receipt sticker - method GetStickers.

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

Method requires authorization via API key

Input parameters

Name	Data type	Default value	Description
number	String	*	Receipt number

Output parameters

Represents as json. Collection of objects {barcode, categoryName, receiptNumber, receiver, dateSend, dateReceive, warehouseSend, warehouseReceive, totalPlaces, rang, econom, delivery}.

Barcode – Barcode
categoryName – product category
receiptNumber – receipt number
receiver – recipient's name
dateSend – dispatch date
dateReceive – arrival date
warehouseSend – dispatch warehouse
warehouseReceive – arrival warehouse
totalPlaces – total amount of parcels
rang – rank
econom – economical but longer delivery
delivery – is there a targeted delivery

Output parameters format
application/json, text/json

Example:

```
In json format:
{
  "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": "МАРІУПОЛЬ-1",
      "warehouseReceive": "ОЛЕКСАНДРІЯ",
      "totalPlaces": "11",
      "rang": 1,
      "econom": false,
      "delivery": true,
      "postomat": false
    },
    …
  ],
  "status": true,
  "message": ""
}
```

In xml format:

```
<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>МАРІУПОЛЬ-1</warehouseSend>
         <warehouseReceive>ОЛЕКСАНДРІЯ</warehouseReceive>
         <totalPlaces>11</totalPlaces>
         <rang>1</rang>
         <econom>false</econom>
         <delivery>true</delivery>
         <postomat>false</postomat>
      </Sticker>
   </data>
</ApiResultStickers>
```

6.17 Consolidation of receipts into one pick up request - method PostAddReceiptIntoPickUpRequest.

POST api/v4/Public/PostAddReceiptIntoPickUpRequest

Method requires authorization via API key

Input parameters

Name	Data type	Description
input	ExptactReceiptModel	A model that describes the input and output parameters of the receipt

Input data example:

```
var data = {
    "pickUpContactName": "Василий", //contact person for pick up
    "pickUpContactPhone": "0500000000", // Pick up contact phone number
    "pickUpAddress": "ул. Науки, 50", // Pick up address if not created
    "descentFromFloor": 4, //Descent from the floor
    "pickUpAddressId": "a5eaf714-fb60-e411-b421-000d3a200936", //Pick up address if it was
    created earlier
    "receiptNumberList": ["9900070818", "9900070822", "9900070823"] //Array of receipt
    numbers
}
```

As an array, you can send receipts united by one city of dispatch, date of dispatch.

Output parameters

Output parameters format
application/json, text/json

Example:

```
In json format:
{
  "status": true,
  "message": "",
  "data": []
}
In xml format:
<ApiResult>
        <status>true</status>
        <message/>
        <data />
</ApiResult>
```

6.18 Getting the dispatch register - method SendingRegister.

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

Input parameters

Name	Data type	Default value	Description
id	String	*	Pickup request number

Output parameters

Output parameters format
A file with html extension

Link Example:

http://www.delivery-auto.com.ua/uk-UK/SharedForms/SendingRegister?id=35334

Output file:

```
Register_35334.html
```

7. Operations with receipt logs

7.1 Getting receipt logs – method GetUnidersalLogsByReceiptNumber.

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

Method requires authorization

Input parameters

Name	Data type	Default value	Description
number	String	*	Receipt number
culture	String	uk-UA	Culture; Valid values (en-US, uk-UA).

Output parameters

Represents a list of json objects. Object {id, CreatedOn, WarehouseId, WarehouseName, OperationCode, OperationName }.

CreatedOn – Creation data
WarehouseId – Warehouse Id
WarehouseName – Warehouse name
OperationCode – Operation code
OperationName – Operation name

Output parameters format
application/json, text/json

Example:

```
{
  "status": true,
  "message": "",
  "calculatorModel": [
    {
      "CreatedOn": "2014-12-22T13:12:39.393",
      "WarehouseId": "bdff546c-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. Additional directories

8.1 Receipt status directory

Code	Meaning
0	Issued. The cargo has been delivered to the recipient in full size.
1	Partially issued. The cargo was not delivered to the recipient in full size.
2	Formalized. The cargo has arrived at the warehouse of dispatch.
3	Disposed
4	Sold
5	Canceled
6	On the way. The cargo is on the way.
7	Available for issue. The cargo has arrived at the receiving warehouse.
8	Reserved
9	Forwarded to another warehouse
10	Unloading in the warehouse
11	The cargo has arrived at the transit warehouse
12	Preparing for delivery by courier
13	Is being delivered by courier

8.2 Currency directory

Code	Meaning
100000000	Hryvnia

8.3 Operation codes dirctory

Operation code	Description
100000002	Formalization of a receipt in the warehouse
100000013	Change receipt arrival date
100000016	Forwarding a receipt between warehouses of the same city
100000018	Loading cargo into a car
100000026	Change the date of recieving the receipt on delivery
100000059	Loading cargo into a car
100000060	Unloading cargo from the car
100000061	Unloading cargo from the car
100000062	Loading cargo into a car
100000070	Cancellation of the issuance of a receipt
100000072	Issuance of cargo to the client
100000079	Loading cargo into a car
100000082	Forwarding a receipt between cities
100000111	Formalization of a receipt in the warehouse
100000115	Formalization of the return receipt in the warehouse
100000122	Cancellation of the issuance of a receipt
100000125	Issuance of cargo to the client
100000132	Unloading cargo from the car

8.4 Receipts types directory

Code	Meaning
2	Regular receipt
4	Forwarding
5	Delivery
6	Insurance
7	Cargo pick up
8	Service sales
10	Cash on delivery by card
11	Courier delivery
13	Cash on delivery
14	Refundable payment