Automatic order forwarding via API call.

Direction of communication:

Pepita → Partner store (push), i.e.:
When an order arrives at Pepita, the order is automatically forwarded to the API endpoint provided by the partner store.

It might be useful to agree on a common key (a random string of sufficient length) for the API, and this key would always be added to the code when calling the URL used for forwarding the order. (For example,

https://api.partneraruhaz.hu/order/store?apikey=qwerty

In this way, we can avoid needless communication through the URL.

Example of JSON data structure to be used for order forwarding, and the definition of properties:

"origin":This is where the order originates from. It might be useful to save it for later for purposes of statistics (or for other reasons) so that the origin of the order is known.

"id": Unique order ID, which can also be found in the partner's administration

interface. "date": Date of the order (when the customer submitted the order).

"payment_mode": Payment mode, possible values:

- cod: cash on delivery
- transfer: money transfer - creditcard: credit card

"customer_message": Comment by the customer that can be optionally entered during the ordering process

"courier_message": Customer's comment about delivery. E.g.: The courier should call me before arrival on 06-20-1234567.

"status": By default, this is not forwarded, but we can forward any status that triggers an event at the partner store, if required. E.g.: "new_order" "payment_status": Payment status - it may take either of two values:

- paid: paid – this status is normally assigned to payment by credit card -
- unpaid: unpaid

"total_shipping_price": The total gross value of the shipping price to be paid "total_shipping_price_currency": Currency of the shipping price

"voucher": The total amount paid by voucher "delivery_mod":
Possible values:

- shipping: The default value for an order
- gls: Delivery via GLS courier
- gls_parcellocker: GLS parcel box delivery
- gls_xxl: Delivery via GLS XXL courier
- mpl: Delivery via MPL courier
- email: Delivery by e-mail

"package_label": Base64 encoded content of the pdf package label

"customer": Data structure containing customer information

- "last_name": Customer's last name
- "first_name": Customer's first name
- "phone": Customer's phone number

"email": Customer’s e-mail address

"billing_name": Billing information – billing name,

"billing_country": Billing information – country code,
"billing_city": Billing information – city, town or village,

"billing_street": Billing information – street, house, floor, door number

"billing_street_address": Billing information – street name

"billing_house_number": Billing information – house, floor, door number "billing_postal_code": Billing information – postal code,

"shipping_country": Shipping information – country,

"shipping_county": Shipping information – county, only for Romanian orders,

"shipping_city": Shipping information – city, town or village,

"shipping_street": Shipping information – street, house, floor, door number,

"shipping_postal_code": Shipping information – postal code,

"shipping_street_address": Shipping information – street name,

"shipping_house_number": Shipping information – house, floor, door number, "tax_number": Customer’s tax number (if they made a company purchase) "products": Array data structure containing information of ordered products

"id": pepita ID of the product

"sku": stock-keeping unit code of the product (given by the partner)

"currency": currency

"quantity": ordered quantity

"price": gross sales price of the product given in HUF "vat":
VAT rate given as a percentage (e.g. 27%) Example:

{

"origin":"pepita.hu",

"id":118924,

"date":"2018-05-21 10:23:41", "payment_mode":cod,

"customer_message":"Tisztelt Partner\u00fcnk! A vev\u0151 t\u00f6bb helyr\u0151l is v\u00e1s\u00e1rolt, elk\u00e9pzelhet\u0151, hogy megjegyz\u00e9se nem \u00d6nre vonatkozik.

Ebben az esetben hagyja figyelmen k\u00edv\u00fcl! Vev\u0151 megjegyz\u00e9se:

K\u00e9zbes\u00edt\u00e9s el\u0151tt f\u00e9l \u00f3r\u00e1val,h\u00edvjon a

06-30-111- 11111as telefonsz\u00e1mon!",

"courier_message": "Keremhivjanakelotte",

"status":"new_order",

"payment_status":"unpaid",

"total_shipping_price":1200,

“total_shipping_price_currency":"HUF""voucher": "",

"delivery_mod": "mpl",

"package_label": " Base64 encoded content of the pdf package label"

"customer":{

"last_name":"Teszt",

"first_name":"Péter",

"phone":"06201111111",

```
"email":"teszt.peter@pepita.hu",
"billing_name":"Teszt Péter",
"billing_country":"HU",
"billing_city":"Miskolc",
"billing_street":"Teszt u.14",
"billing_street_address":"Teszt u."
"billing_house_number":"14"
"billing_postal_code":"3534",
"shipping_country":"HU",
"shipping_city":"Miskolc","shipping_street":"Teszt u.14",
"shipping_street_address":"Teszt u",
"shipping_house_number":"14",
"shipping_postal_code":"3534”
}, "products":[
{
"id":"139955",
"sku":"ozq123",
"currency":"HUF",
"quantity":2,
"price":3192,
"vat":27, },
{
"id":"139956",
"sku":"ozq124",
"currency":"HUF",
"quantity":2,
"price":3192,
"vat":27,
},
{
"id":"139957",
"sku":"ozq125",
"currency":"HUF",
"quantity":2,
"price":3192,"vat":27,
},
],
}
```


The example JSON data structure below is recorded in response to the request, to make sure the communication has been successful and that the submitted data has been received.

"isError": Indicates whether the partner’s store successfully received the order

information. Possible values:

- false: The order has been received successfully
- true: There was a problem saving the order

"responseCode": ResponseCode returned by the partner store's system
"message": Comment that can be provided by the partner's store; any information related to the communication can be provided, typically in the case of a processing error.

Example 1:

{
"isError":false,

"responseCode:200,"messages":[

] }

{

Example 2:

"isError":true,

"responseCode:500

“message":"Invalid

}