

Tangram REST API (PPG customized)

1	Technical details.....	2
1.1	Error handling.....	2
2	Customers.....	3
2.1	Customer Detail (GET).....	4
2.2	Customer List (GET).....	4
2.3	Customer Create (POST).....	5
2.4	Customer Update (POST).....	6
3	Price.....	7
3.1	Price for single product (GET).....	7
4	Stock.....	9
4.1	Stock for single product (GET).....	9
5	Delivery options.....	10
5.1	Delivery options for <i>single</i> product (GET).....	10
5.2	Delivery options for cart.....	10
6	Orders.....	11
6.1	Create/Simulate order (POST).....	11
6.1.1	Delivery options / shipping cost calculation.....	11
6.1.2	Request.....	11
6.1.3	Response.....	13
6.2	Order history (GET).....	16
6.3	Order details (GET).....	17
6.4	Coupon validate (GET).....	18
7	Quotes (price offer).....	19
7.1	Quote list (GET).....	19
7.2	Quote detail (GET).....	20
7.3	Quote accept (POST).....	21
7.4	Quote reject (POST).....	21
8	Settings/Options.....	22
8.1	Delivery options (GET).....	22
8.2	Payment options (GET).....	22
8.3	Store list (GET).....	23
9	Client Link Interface.....	24
9.1.1	Spúšťanie skriptov.....	24
9.1.2	Nastavenie databázy/schémy.....	24
9.2	Export PRODUCTS.....	24
9.3	Export CUSTOMERS.....	24
9.4	Export ORDERS.....	24

1 Technical details

Protocol = HTTP

Request/Response body = JSON

Character set = UTF-8

Authentication = none

GET commands has input parameters in Request URL, return values in Response body.

POST commands has all required data in Request body, return value(s) in Response body.

1.1 Error handling

Response header contains HTTP status with standard HTTP error codes:

Status 200 = OK, response body contains JSON formatted result of operation

anything else = system error (error in database, syntax error in script etc.), response body contains PLAIN text error description.

Returned status „200“ does not automatically mean, that operation was successful. It only means, that there was no SYSTEM error. Return value (JSON formatted) can still contain Warning or Error text intended for end-user, usually result of scripted validation rules. Return values depend on specific script implementation and are explained at every specific request.

2 Customers

CUSTOMER fields description (R = read only property, ignored in Create/Update functions)

Id	R	Customer unique ID (String 17) assigned by Tangram at Create (firmy.xx)
Customer_type		1=B2B customer, 0 or anything else=B2C customer (firmy.f)
Valid	R	1=Valid, 0 or anything else=Inactive (waiting for validation or deactivated) (firmy.neaktivna)
Company_name		B2B: company name (read/write) B2C: person first+last name (read-only, auto generated in CustomerCreate)
Customer_name		Customer name (firmy.eshop_name)
Siret		Company register id (IČO for SK&CZ), B2B only (firmy.ico)
Vat		VAT register id, B2B only (firmy.icdph)
Email		E-mail address (firmy.email)
Phone		Phone number (firmy.telefon)
Invoice_addr1		Address line 1 (firmy.nazov2), taken from 2. line of company name
Invoice_addr2		Address line 2 (firmy.ulica), taken from „Street“ field
Invoice_post		Postal ZIP code (firmy.psc)
Invoice_city		City (firmy.mesto)
Invoice_country		Country code, 2 character ISO code „SK“, „CZ“ ... (firmy.stat)
Delivery_company		Delivery address - company name
Delivery_firstname		Customer first name (firmy.delivery_firstname)
Delivery_lastname		Customer last name (firmy.delivery_lastname)
Delivery_addr1		Delivery address line 1 (firmy.delivery_addr1)
Delivery_addr2		Delivery address line 2 (firmy.delivery_addr2)
Delivery_post		Delivery postal code (ZIP) (firmy.delivery_post)
Delivery_city		Delivery city (firmy.delivery_city)
Delivery_country		Delivery country code, 2 character ISO code (SK, CZ, ...) (firmy.delivery_country)
Customer_group	R	Customer group code (firmy.odbskup)
Price_group	R	Price group code (firmy.censkup)
Discount	R	Customer's discount (%)
Default_shop_id		Preferred retail shop code for „click & collect“ delivery mode (firmy.sklad)
Default_shop_name	R	Preferred retail shop name for „click & collect“ delivery mode
Payment_methods	R	Allowed payment methods P=invoice (if allowed), H=cash on pickup, D=cash on delivery, V=paid in advance (credit card), X=other
Delivery_options	R	Allowed delivery options: A = pickup at default shop (no delivery) B = send to Delivery address defined in Customers database C = send to one-time address, specified in Order Value is string combined from allowed options. Example: „ABC“ means all options active
Sales_rep_id		Sales representative id (firmy.obchzast)
Sales_rep_phone	R	Sales representative phone (not implemented in Tangram database)
Loyalty_program	R	Member of Loyalty program 1=yes, 0=no („Talos yes/no“)
Loyalty_points	R	Loyalty program points collected („Talos points“)
Timestamp	R	Date and time of creation or last modification (yyyy-mm-ddThh:mm:ss)

2.1 Customer Detail (GET)

Get customer info for one customer specified by unique ID

URL = **ppg/customer?id=xxxx**

Response body example

```
{
  "customer":
  {
    "id": "ESHOP.000157",
    "customer_type": null,
    "valid": 1,
    "company_name": "John Scalzi, end user",
    . . .
  }
}
```

2.2 Customer List (GET)

Loading new or updated customers from Tangram. For new customers registered in Tangram, who supplied e-mail address used as web login. Password should be generated by web server. Invitation mail should be sent to customer, informing about login credentials to e-commerce site.

Returns list of customers, created or updated in last x days or last y hours. Provide either „days“ or „hours“, not both. **If no parameter is specified, all customers are exported.** Value days=0 means today, days=1 means today and yesterday, etc. Value hours=1 means last hour, hours=24 means last 24 hours, etc.

URL = **ppg/customerlist ? days=x & hours=y**

Response body example (array of customers)

```
{
  "rows":
  [
    {
      "id": "ESHOP.000157",
      "customer_type": null,
      ...
    },
    ...
    {
      "id": "1234567890",
      "customer_type": "B2B",
      ...
    }
  ]
}
```

2.3 Customer Create (POST)

Create new customer. Request body can contain any field, except of „read-only“ ones (see list of fields). Read-only or misspelled fields dont throw error, they are silently ignored. Omitted fields (except of mandatory ones) will get their default values.

Company and person name rules:

B2B: "company_name" is mandatory, "delivery_first/last" names are optional.

B2C: person name should be specified in "delivery_firstname" and "delivery_lastname". Don't specify "company_name" as it is auto-generated from first+last name.

Return value is unique CUSTOMER-ID. Specific implementation for PPG uses Siret (IČO in SK/CZ) for B2B customers, and „ESHOP.nnnnnn“ with autoincremented value for B2C customers.

Mandatory fields:

Siret for B2B customers

Email for all customers

Validation rules:

1. Siret must be unique for B2B customers
2. E-mail must be unique for all customers

URL = **ppg/customercreate**

Request body example

```
{
  "client_type": 1,
  "company_name": "Future Minds Ltd.",
  "siret": "9876543210",
  "vat": "CZ9876543210",
  "email": "sales@futureminds.cz",
  "phone": "+420555123456789",
  "invoice_addr1": "adresa 1. riadok",
  . . .
}
```

Success return value (generated uique ID for created customer) in Response body

```
{
  "result": "OK",
  "id": "ESHOP.000157"
}
```

Error return value in Response body

```
{
  "result": "ERROR",
  "id": "9876543210",
  "errid": "EMAIL",
  "errtext": "localized error message"
}
```

Error list :

EMAILEXIST = e-mail address already exists in Customer's database, ID contains conflicting customer ID

EMAILNULL = e-mail cannot be empty

EMAILBAD = e-mail bad format (must contain @, cannot contain spaces or commas)

EXIST = customer already exists in Customer's database (only for B2B, errtext contains description and name of the conflicting customer)

SIRETNNULL = Siret code cannot be empty for B2B customer

SIRETBAD = Siret must be Integer value for B2B customer

2.4 Customer Update (POST)

Update selected fields for existing customer, all fields are optional, only specified fields are updated.

Misspelled or not-allowed fields throw ERROR result.

Updatable fields for all types of customers:

email, phone,
delivery_company, delivery_firstname, delivery_lastname,
delivery_addr1, delivery_addr2,
delivery_post, delivery_city, delivery_country,
default_shop_id

Additional fields for B2C customers only:

invoice_addr1, invoice_addr2, invoice_post, invoice_city, invoice_country

Company name rules:

B2B: "company_name" is not allowed to be updated

B2C: "company_name" will be auto-updated if delivery_firstname or delivery_lastname is specified

Validation rules:

1. customer ID must be specified
2. customer ID must be found in database
3. field names are checked

URL = **ppg/customerupdate?id=xxxx**

Request body example

```
{
  "delivery_firstname": "John",
  "delivery_lastname": "Horowitz",
  "email": "sales@futureminds.cz",
  "phone": "+420555123456789",
  "invoice_addr1": "adresa 1. riadok",
  . . .
}
```

Success return value (generated unique ID for created customer) in Response body

```
{
  "result": "OK",
  "id": "ESHOP.000157"
}
```

Error return value in Response body

```
{
  "result": "ERROR",
  "id": "9876543210",
  "errid": "xxx",
  "errtext": "xxx"
}
```

Error list :

IDNULL = customer ID not specified

NOTFOUND = customer ID not found in database

DENIED = list of rejected fields (access denied or wrong field name)

3 Price

3.1 Price for single product (GET)

Get price for one item (standard product. Ready-mix or tinted). API gets Price_group and Customer_discount (%) from Customer's database. Then calculates price specified item. If item is tinted color, then tinting formula must be specified in „components“ array.

URL = **ppg/price**

Request body example – standard products and ready-mix

```
{
  "customer_id": "1234567890",
  "item_id": "00192051",
}
```

Request body example – tinted products

```
{
  "customer_id": "1234567890",
  "item_id": "00192051",           # base material, identified by SKU
  "colorants":
  [
    { "color_id": "URY", "quantity": 0.014 }, # colorant 1, code from colorbank
    { "color_id": "ABB", "quantity": 0.020 } # colorant 2, code from colorbank
  ]
}
```

Request fields

Customer_id	Customer unique ID
Item_id	Product SKU code (standard products or base of tinted color)
Colorants	List of colorants (formula) for tinted color
Color_id	Colorant code as returned from ColorBank
Quantity	Colorant quantity for 1 unit of color base

Response body example

```
{
  .."result":"OK",
  "customer_name":"Demo odberateľ DPH",
  "price_group":"001",
  "discount":13,
  "item_id":"00192051",
  "priceb_bv":1.29167,
  "priceb_iv":1.55,
  "priced_bv":1.12375,
  "priced_iv":1.34850,
  "vat":20,
  "promotion": "",
  "price_discount":13,
  "price_type":"A",
  "valid_from":"2017-09-28",
  "valid_to":"2999-12-31",
  "comment":"Cena za balenie pevne dana"
}
```

Response fields

customer_name	Customer's name or null if customer_id not found in database
price_group	Price_group for customer (default „basic“ price group, if customer not found)
discount	Customer's discount (%) for all rows in order
item_id	Product SKU code (for tinted colors it's code of color base)
priceb_bv	Price Basic Before VAT
priceb_iv	Price Basic Including VAT

priced_bv	Price Discounted Before VAT
priced_iv	Price Discounted Including VAT
vat	VAT (%)
promotion	Promotion code or null if price is not in promotion
price_discount	Final discount (%), combination of customer's discount and product promo discount
price_type	A=basic price, B=promotion price, E or F=special customer's price
valid_from	Date when price starts (important for promotion prices)
valid_to	Date when price ends (including this date)
comment	Any comment

Error return value in Response body (id = item ID)

```
{
  "result": "ERROR",
  "id": "9876543210",
  "errid": "xxx",
  "errtext": "xxx"
}
```

Error list (value of „errid“):

IDNULL = item ID not specified

PRICENULL = price of „item_id“ not found in pricelist

COLIDNULL = colorant code „color_id“ not found in conversion table (cannot find colorant SKU code)

COLPRICENULL = price of colorant „color_id“ not found in pricelist

4 Stock

4.1 Stock for single product (GET)

Current stock quantity for single item, in selected store and predefined nearby stores. Returned as array of rows.

URL = **ppg/stock?id=xxx&store=xxx**

[**id**] product SKU code (mandatory)

[**store**] store code (mandatory)

Table of predefined nearby stores is saved in Tangram database and is editable by Tangram users.

Response body example

```
{
  "store": "001",
  "stock": 6,
  "store_name": "Brno, Maříkova 23"
  "nearby":
  [
    {
      "store": "012",
      "stock": 675.48,
      "store_name": "Blansko, Sedlčanská 1452/3"
    },
    {
      "store": "018",
      "stock": null,
      "store_name": "Slavkov u Brna, Masarykova 4"
    },
    . . .
  ]
}
```

Comments

Store	Retail store code
Stock	Stock quantity (precision 0,001)
Store_name	Retail store name
Nearby	Array of stock quantity in nearby stores

5 Delivery options

5.1 Delivery options for single product (GET)

Delivery options for single item. Used on Product page to display delivery options.

If (optional) quantity is specified, then availability is also checked against total weight of products.

If (optional) customer ID is specified then availability is also checked against customer's configuration.

URL = **ppg/delivery?id=xxx&quantity=nnn&customer=xxx**

[**id**] product SKU code (mandatory)

[**quantity**] product quantity (optional)

[**customer**] customer ID (optional)

Response body example (array of objects):

```
[
  {
    "delivery": "STORE",
    "available": true
  },
  {
    "delivery": "PPL",
    "available": false
  },
  . . .
]
```

Comments

Delivery	Delivery service code
Available	true or false

5.2 Delivery options for cart

Delivery options for each row in cart, and cart as a whole, are calculated in OrderCreate (Simulation mode).

6 Orders

6.1 Create/Simulate order (POST)

Simulate price calculation for cart or save customer order to Tangram database.

6.1.1 Delivery options / shipping cost calculation

Simulation mode (SIM=1). Delivery options are ALWAYS calculated, as eshop needs ROW delivery options to display in cart page. There are no additional fields in **request** body. Delivery options are returned in **response** body for each row and for cart as a whole.

Save mode (SIM=0). There are new "shipping_xxx" fields in request body, which should contain selected service and price code. These fields are mandatory only if customer selected "Home delivery" or "Parcelshop delivery". For "Pickup at store" these fields could be omitted and "delivery_shop" should be specified.

6.1.2 Request

URL = ppg/ordercreate

Request body example. Save mode (sim=0). Order contains 3 standard products (P), one tinted color (T) and one coupon code. Tinted color is two 5kg buckets with two tints. Quantities of tints are calculated for 1 piece of base material (5kg in this example).

```
{
  "header":
  {
    "sim": 0,                                # 1 = simulation, 0 = save (default)
    "customer_id": "123",                    # customer unique ID
    "extid": "123456",                       # order number assigned by e-shop
    "date": "2019-09-09",                   # date of order, default=today
    "spedition": "GEIS",                   # delivery option (renamed to "shipping_service")
    "payment": "V",
    "payinfo": "765213284",
    "comment": "poznámka ...",               # comment from customer (max 2000 chars)
    "delivery_shop": "002",                  # delivery shop code (for pickup at specified shop)
    "delivery_company": "Tilbury",           # company name (optional)
    "delivery_name": "Jacques Mayol",        # person name
    "delivery_addr1": "20 rue marx dormoy",  # address details
    "delivery_addr2": "Reims",
    "delivery_post": "75018",
    "delivery_city": "Paris",
    "delivery_country": "FR",
    "delivery_email": "pso@tilbury.com",     # e-mail
    "delivery_phone": "+421555123456",       # phone number
    "shipping_service": "PPL",               # shipping service code (Save mode only)
    "shipping_pricecode": "DPPL30",          # shipping price code (Save mode only)
    "shipping_parcelshop": "KM123456"       # parcel shop code or null (Save mode only)
  },
  "rows":
  [
    { "type": "P", "item_id": "191", "quantity": 3.5 }, # standard product (SKU)
    { "type": "P", "item_id": "192", "quantity": 40 }, # standard product (SKU)
    { "type": "T", "item_id": "00192051", "quantity": 2, # base material, identified by SKU
      "comment": "Primalex interior 2 x 5kg, shade Light olive green",
      "shade": "15264",
      "colorants":
      [
        { "color_id": "URY", "quantity": 0.014 }, # colorant 1, code from colorbank
        { "color_id": "ABB", "quantity": 0.020 } # colorant 2, code from colorbank
      ]
    },
    { "type": "P", "item_id": "193", "quantity": 1 }, # standard product (SKU)
    { "type": "C", "item_id": "PRIMA1900024" }        # coupon code
  ]
}
```

Comments to header

Spedition		Delivery method from list provided by DeliveryOptions request
Payment		Payment method, fixed list of codes (default „H“): P=invoice (if allowed), H=cash on pickup or cash on delivery, V=paid in advance (credit card, bank transfer, ...), X=other
Payinfo		Payment info – transaction number for ComGate payments (type „V“)
Shipping info		Shipping info should be specified in Save mode (SIM=0) only. In case of "pickup at PPG store" it should be ommited. "Service" and "pricecode" are the values from "delivery_options" from previous Order Simulation, choosen by customer. "Parcelshop" is parcel shop code for PPL shipping service.

ORDER ROW fields description (**M** = Mandatory)

Type	M	P=Standard product, T=Tinted color header (informational row with color code and optional description), C=Coupon code. Mapped to OBPR.KP
Item_id	M	Product code: P-rows: product SKU code T-rows: SKU code for base material C-rows: Coupon code
Color_id		Colorant code, used in T-rows/component list, where SKU code is not available. Tangram will try to find colorant SKU code from Collection and Color_id in definition table (stored in Tangram database, editable by Tangram users).
Quantity		Product quantity (obpr.mn), mandatory for P and T rows.
Comment		Comment line for tinted color header, only for T rows (optional)
Shade		Color shade code (T-rows), mapped to OBPR.ZAK
Colorants		Array of components for tinted color, first row should contain base material, other rows are colorants (dyes). Number of components is not limited.

6.1.3 Response

Response body example, returns rows with calculated prices and coupon with calculated discount. Rows are returned in the same order as in request body. No row is added or skipped. P and T rows are extended with unit and total prices. C row is extended with Promotion comment and calculated discount.

If Result is „OK“, then no Errid and Errtext fields are generated. If Result is „ERROR“, then Errid and Errtext fields contain additional information.

```
{
  "result": "OK" or "ERROR",
  "errid": "xxx",
  "errtext": "xxx",
  "header":
  {
    "order_id": "25FC3BE90F7E6F5C",      # unique ID, generated by Tangram in save mode
    "order_visible_id": "E017/1523",      # order number formatted for customer
    "order_group": "E017",                # order group, generated by Tangram in save mode
    "order_no": "1523",                   # order number, generated by Tangram in save mode
    "... other data copied from request.header (customer_id, delivery_shop, ...)"
    "totalb_bv": 1000,                    # total basic before VAT
    "totalb_iv": 1200,                    # total basic including VAT
    "totald_bv": 900,                     # total discounted before VAT
    "totald_iv": 1080,                    # total discounted including VAT
    "weight": 26.7,                       # total weight for cart
    "delivery_options": [ ... array of cart delivery options, see below ... ]
  },
  "rows":
  [
    { "rownum": 1, "type": "P", "item_id": "191",
      "comment": "this is user comment",
      "name": "Brush 60 mm",
      "package": 1,
      "package_unit": "pc",
      "quantity": 3.5,
      "stock": 4,
      "nearby": 26,
      "priceb_bv": 100, "priceb_iv": 120, "priced_bv": 90, "priced_iv": 108,      # prices
      "totalb_bv": 350, "totalb_iv": 420, "totald_bv": 315, "totald_iv": 378,    # prices
      "discount": 10, "vat": 20,                                                 # prices
      "weight": 26.7,                                                           # total weight for row
      "delivery_options": [ ... array of row delivery options, see below ... ]
    },
    { "rownum": 2, "type": "P", "item_id": "192", ... },
    { "rownum": 3, "type": "T",
      "item_id": "00192051",
      "comment": "this is user comment (optional)",
      "name": "COLORMIX {15264} Primalex interior 5kg",
      "package": 5,
      "package_unit": "kg",
      "shade": "15264",
      "quantity": 2,
      "stock": null,
      "nearby": 30,
      "... prices ...",
      "colorants":
      [
        { "color_id": "URY", "item_id": "00654301", "quantity": 0.014 },
        { "color_id": "ABB", "item_id": "00654302", "quantity": 0.020 }
      ]
    },
    { "rownum": 4, "type": "P", "item_id": "193", "quantity": 1, ... },
    { "rownum": 5, "type": "C", "item_id": "PRIMA1900024",
      "comment": "Autumn 2019 sale for Primalex products 15%",
      "discount": 158.20
    }
  ]
}
```

Returned ORDER ROW fields description

Rownum	Row number 1..n
--------	-----------------

Type	P=Standard product, T=Tinted color, C=Coupon
Item_id	Product SKU code (obpr.kod) or Coupon code (all rows)
Name	P-row: „product name“ T-row: „COLORMIX {shade} product name“
Package	Package size and unit
Shade	Color code for tinted colors
Quantity	Ordered quantity
Stock	Available stock in selected shop
Nearby	Available stock in nearby shops (sum of all nearby shops)
Colorants	Array of colorants for tinted colors
Priceb_bv	Unit Price Basic Before VAT (for P+T rows)
Priceb_iv	Unit Price Basic Including VAT (for P+T rows)
Priced_bv	Unit Price Discounted Before VAT (for P+T rows)
Priced_iv	Unit Price Discounted Including VAT (for P+T rows)
Totalb_bv	Total Price Basic Before VAT (for P+T rows)
Totalb_iv	Total Price Basic Including VAT (for P+T rows)
Totald_bv	Total Price Discounted Before VAT (for P+T rows)
Totald_iv	Total Price Discounted Including VAT (for P+T rows)
Discount	Discount (% for P-rows and T-subrows, absolute value for C rows)
VAT	VAT rate % (P and T rows))
Comment	Comment for C-row: promotion name and other info Comment for P+T rows: optional comment entered by customer

ROW delivery options

Field	Description
Service	Delivery service code ("PPL")
Available	Delivery availability (true or false)
Errorcode	Error code if item delivery is not available (Available=false) "WEIGHT" total weight for item exceeds limit "UNSHIPPED" item is not allowed for delivery

CART delivery options

Field	Description
Service	Delivery service code ("PPL")
Available	Delivery availability (true or false)
Pricecode	Delivery price code (if Available=true)
Pricevalue	Delivery price value (if Available=true)
Errorcode	Error code if whole cart delivery is not available (Available=false) "WEIGHT" total weight for cart exceeds limit "UNSHIPPED" one or more items are not allowed for delivery

ERRID values in case of „ERROR“:

ERRID	ERRTEXT
JSON	EN: JSON data error - "header" block not found This is system error checked while parsing JSON data from request.

Currently there is only one possible error checked in OrderCreate. All missing values are either replaced by

defaults or processed as empty.

Future planned error codes (not implemented yet):

CUSTOMERNF = Customer ID not found in database

SHOPNF = Shop ID not found in database

ITEMNF = Item ID (product SKU) not found in database

LIMIT = Order value exceeded Customer's credit limit. Limit is checked only for Payment=Invoice.

6.2 Order history (GET)

Get list of all orders for specific customer.

URL = **ppg/orderlist?customer=xxx&days=nnn**

[customer] customer ID, mandatory parameter

[days] date filter, optional, default value is 365 (one year)

Response body example (array of orders)

```
{
  "orders":
  [
    {
      "order_id":"25C6CABF9B0BC2F0",          # unique ID
      "order_visible_id":"ESHOP/123456818",    # order number formatted for customer
      "status":"5",                           # order status (fixed list)
      "order_group":"ESHOP",                  # order group
      "order_number":"123456818",              # order number (sequential)
      "customer_id":"ESHOP.000001",            # customer unique ID
      "pricegroup":"001",
      "discount":3,                           # discount (%)
      "ext_id":"123481",                       # order number from e-shop
      "date":"2020-01-15",                     # order date
      "spedition":"A",                         # delivery method
      "payment":"D",                           # payment method
      "payinfo":"765213284",                   # payment transaction number
      "comment":"testovacia objednávka Tilbury",
      "delivery_shop":"63",                    # delivery details
      "delivery_company":"Tilbury",
      "delivery_name":"Jacques Mayol",
      "delivery_addr1":"20 rue marx dormoy",
      "delivery_addr2":"Reims",
      "delivery_post":"75018",
      "delivery_city":"Paris",
      "delivery_country":"FR",
      "delivery_email":"pso@tilbury.com",
      "delivery_phone":"+421555123456",
      "totalb_bv":134.46,                      # total basic before VAT
      "totalb_iv":161.26,                     # total basic including VAT
      "totald_bv":130.42,                     # total discounted before VAT
      "totald_iv":156.39                      # total discounted including VAT
    },
    . . .
  ]
}
```

Field names are identical with OrderCreate request

Fields STOCK and NEARBY are not calculated/exported

6.3 Order details (GET)

Get order details from Tangram database. Response contains header and array of rows. Format of header and rows is identical to OrderCreate request. Fields STOCK and NEARBY are not calculated/exported.

URL = **ppg/orderdetail?order_id=25C94639BFF26E91**

Response body example:

```
{
  "header":
  {
    # order header is identical to OrderList request
  },
  "rows":
  [
    {
      "rownum":1,                # Row number 1..n
      "type":"T",               # Row type P, T or C
      "item_id":"00332694",     # Item SKU code
      "comment":"this is optional comment from customer",
      "name":"product name",
      "package":5,              # package details
      "package_unit":"kg",
      "shade":"BLK-001",        # color code (tinted colors only)
      "quantity":3,             # item quantity
      "priceb_bv":27.51,        # Unit price, before discount, before VAT
      "priceb_iv":33.01,        # Unit price, before discount, incl. VAT
      "priced_bv":26.68,        # Unit price, discounted, before VAT
      "priced_iv":32.01,        # Unit price, discounted, incl. VAT
      "totalb_bv":82.53,        # Total price, before discount, before VAT
      "totalb_iv":99.03,        # Total price, before discount, incl. VAT
      "totald_bv":80.04,        # Total price, discounted, before VAT
      "totald_iv":96.03,        # Total price, discounted, incl. VAT
      "discount":3,             # discount (%)
      "vat":20                  # VAT (%)
    },
    . . .
  ]
}
```

6.4 Coupon validate (GET)

Validate coupon-code against Tangram database

Coupons can be allowed only for specified group of customers (filter in coupon definition)

Every coupon has validity period (date from/to)

Coupon usage methods:

- FREE = coupon usage not checked, can be used repeatedly
- CUSTOMER = coupon usage once per customer (for specified group of customers)
- UNIQUE = coupon code is unique and can be used only once

Response contains error code/text or coupon name

URL = **ppg/CouponValidate?customer=123&coupon=BALA2020015**

Response body example:

```
{
  "result": "OK" or "ERROR",
  "errid": "",
  "errtext": "",
  "customer_id": "123",
  "coupon_id": "BALA2020015",
  "coupon_name": "Balakryl 2020 zľava 15% na všetky výrobky Balakryl"
}
```

Result = „OK“ or „ERROR“

ERRID values in case of „ERROR“:

ERRID	ERRTEXT
NOTFOUND	EN: Coupon code [\$coupon_id] not found in database CZ: Neznámý kód kupónu [\$coupon_id] SK: Neznámý kód kupónu [\$coupon_id]
DATEBEFORE	EN: Coupon not valid before \$dateBeg CZ: Kupón ještě není platný SK: Kupón ešte nie je platný
DATEAFTER	EN: Coupon not valid after \$dateEnd CZ: Kupón už není platný SK: Kupón už nie je platný
CUSTOMER	EN: Coupon not valid for customer [\$customer_id] CZ: Kupón není povolený pro zákazníka [\$customer_id] SK: Kupón nie je povolený pre zákazníka [\$customer_id]
USEDUNIQUE	EN: Coupon already used CZ: Kupón už byl jednou použitý SK: Kupón už bol raz použitý
USEDCUSTOMER	EN: Coupon already used for customer [\$customer_id] CZ: Kupón už byl použitý pro zákazníka [\$customer_id] SK: Kupón už bol použitý pre zákazníka [\$customer_id]

ERRTEXT contains error description in english

COUPON_NAME contains coupon name, if coupon was found in database

7 Quotes (price offer)

Quote is a kind of not-confirmed Order. Quote is created in Tangram database, using Tangram's built-in function. Notification of new Quote is sent to customer by e-mail from Tangram (built-in mail client).

E-shop can list all pending Quotes, display detail of selected Quote, then "accept" or "reject" Quote.

7.1 Quote list (GET)

List of all pending Quotes for selected Customer. Quote data structure is similar to Orders, except of payment information. Customer has to be already registered in e-shop database.

URL = **ppg/QuoteList?customer=123**

Response body example:

```
{
  "quotes":
  [
    {
      "quote_id":"25C6CABF9B0BC2F0",          # unique ID
      "quote_visible_id":"Q01/123456818",      # quote number formatted for customer
      "quote_group":"Q01",                    # quote group
      "quote_number":"123456818",              # quote number (sequential)
      "customer_id":"123",                     # customer unique ID
      "pricegroup":"001",                      # price group code
      "discount":3,                            # discount (%)
      "date":"2020-01-15",                     # order date
      "comment":"testovacia cenová ponuka",
      "spedition": "GEIS",                     # delivery option code
      "delivery_shop":"63",                    # delivery details
      "delivery_company":"Tilbury",
      "delivery_name":"Jacques Mayol",
      "delivery_addr1":"20 rue marx dormoy",
      "delivery_addr2":"Reims",
      "delivery_post":"75018",
      "delivery_city":"Paris",
      "delivery_country":"FR",
      "delivery_email":"jacques.mayol@tilbury.com",
      "delivery_phone":"+421555123456",
      "totalb_bv":134.46,                      # total basic before VAT
      "totalb_iv":161.26,                     # total basic including VAT
      "totald_bv":130.42,                     # total discounted before VAT
      "totald_iv":156.39                      # total discounted including VAT
    },
    . . .
  ]
}
```

7.2 Quote detail (GET)

Detail of selected Quote (id = value of "Quote_ID" from QuoteList). Item data (name, package, ...) are retrieved from Tangram database, as item does not to be registered in e-shop.

URL = **ppg/QuoteDetail?id=xxxxx**

Response body example:

```
{
  "header":
  {
    # quote header is identical to QuoteList request
  },
  "rows":
  [
    {
      "rownum":1,                # Row number 1..n
      "item_id":"00332694",      # Item SKU code
      "comment":"this is optional comment from customer",
      "name":"product name",
      "package":5,               # package size
      "package_unit":"kg",
      "quantity":3,              # item quantity
      "priceb_bv":27.51,          # Unit price, before discount, before VAT
      "priceb_iv":33.01,          # Unit price, before discount, incl. VAT
      "priced_bv":26.68,          # Unit price, discounted, before VAT
      "priced_iv":32.01,          # Unit price, discounted, incl. VAT
      "totalb_bv":82.53,          # Total price, before discount, before VAT
      "totalb_iv":99.03,          # Total price, before discount, incl. VAT
      "totald_bv":80.04,          # Total price, discounted, before VAT
      "totald_iv":96.03,          # Total price, discounted, incl. VAT
      "discount":3,               # discount (%)
      "vat":20                    # VAT (%)
    },
    . . .
  ]
}
```

7.3 Quote accept (POST)

Acceptation of Quote, Tangram will copy Quote to standard Order.

Only "quote_id" field is mandatory, all other fields are optional.

Content of Quote is not changed. Prices are not re-calculated, they are copied from Quote unchanged. Provided values (like Delivery address) will be used as replacement for original data in created Order.

If customer pays in advance, E-shop should provide Payment="V" and Payinfo=Variable symbol.

URL = **ppg/QuoteAccept**

Request body example

```
{
  "quote_id": "25C6CABF9B0BC2F0",          # unique ID
  "payment": "D",                          # optional payment type (same as Orders)
  "payinfo": "765213284",                  # optional variable symbol if paid by card online
  "comment": "poznámka ...",               # optional comment from customer (max 2000 chars)
  "spedition": "GEIS",                    # delivery option code ("A" for pickup at the store)
  "delivery_shop": "63",                  # delivery details (same as Orders)
  "delivery_company": "Tilbury",
  "delivery_name": "Jacques Mayol",
  "delivery_addr1": "20 rue marx dormoy",
  "delivery_addr2": "Reims",
  "delivery_post": "75018",
  "delivery_city": "Paris",
  "delivery_country": "FR",
  "delivery_email": "jacques.mayol@tilbury.com",
  "delivery_phone": "+421555123456",
}
```

Success return value in Response body

```
{
  "result": "OK",
  "id": "25C6CABF9B0BC2F0",
  "text": "Quote accepted"
}
```

Error return value in Response body

```
{
  "result": "ERROR",
  "id": "25C6CABF9B0BC2F0",
  "errid": "XXXXX",
  "errtext": "localized error message"
}
```

7.4 Quote reject (POST)

Rejection of Quote, Tangram will mark Quote as Cancelled and remove it from list of pending Quotes.

URL = **ppg/QuoteReject**

Request body example

```
{
  "quote_id": "25C6CABF9B0BC2F0"          # unique ID
}
```

Success return value in Response body

```
{
  "result": "OK",
  "id": "25C6CABF9B0BC2F0",
  "text": "Quote rejected"
}
```

Error return value in Response body

```
{
  "result": "ERROR",
  "id": "25C6CABF9B0BC2F0",
  "errid": "XXXXX",
  "errtext": "localized error message"
}
```

8 Settings/Options

8.1 Delivery options (GET)

Get list of all delivery options. These codes are used in Order header. Delivery method should be choosed by customer on website.

URL = **ppg/deliverylist**

Response body example

```
{
  "items":
  [
    {
      "id": "SHOP",
      "name": "Pickup at retail shop"
    },
    {
      "id": "FEDEX",
      "name": "FedEx parcel service"
    }
  ]
}
```

8.2 Payment options (GET)

Get list of all payment options. These codes are used in Order header. Payment method should be choosed by customer on website from list of allowed payment.

URL = **ppg/paymentlist**

Response body example

```
{
  "items":
  [
    {
      "id": "D",
      "name": "Cash on delivery"
    },
    {
      "id": "F",
      "name": "Invoice"
    }
  ]
}
```

8.3 Store list (GET)

Get list of all retail shops (stores) for click&collect delivery, These codes are used for anonymous Order (not logged-in customer), where we don't have preferred shop. Also it can be used, when registered customer wants to set delivery to some other than preferred store.

URL = **ppg/storelist**

Response body example

```
{
  "items":
  [
    {
      "id":"63",
      "type":"PPG" or "FRANCHISE",
      "name":"PPG Deco Czech a.s.",
      "addr1":"Náchod, Na Hamrech 874, 547 01",
      "addr2":"Břasy 223",
      "city":"Břasy",
      "post":"338 24",
      "county":"R0",
      "region":"PL",
      "person":"Michal Váleček, vedoucí",
      "phone":"+420 555 123 456",
      "email":"nachod@dumbarev.cz",
      "register":"Obch. rejstřík Krajský soud v Plzni, oddíl B, vložka 1101"
    },
    . . .
  ]
}
```

Comments

Id	Shop ID
Type	Shop type "PPG" or "FRANCHISE"
Name	Shop name (company)
Addr1	Address line 1
Addr2	Address line 2
Post	Postal code (ZIP)
City	City
Country	County (smaller regional unit „okres“)
Region	Region (bigger regional unit „kraj“)
Person	Contact person name
Email	Contact e-mail
Phone	Contact phone number
Register	Company registration info

9 Client Link Interface

This section contains functions, that are exporting data to Client Link. Documentation is in Slovak language, as these functions are used by CZ/SK DevOps team.

ClientLink interface obsahuje funkcie na export:

- CUSTOMERS (zákazníci z tabuľky FIRMY)
- PRODUCTS (cenníkové položky z tabuľky TOVARY)
- ORDERS (skladové doklady kategórie Predaj)

Príslušné skripty sú súčasťou TangramAPI a sú uložené v podadresári TangramAPI/PPG.

Súbory CLxxxx.groovy obsahujú skripty na spúšťanie príslušných funkcií.

Súbory CLxxxx.ftl obsahujú šablóny (FreeMarker) na formátovanie SOAP správ pre ClientLink.

9.1.1 Spúšťanie skriptov

Skripty je možné spustiť manuálne z browsera alebo automatizovane programom CURL (alebo podobným).

Program CURL je open source, stiahnuteľný z <https://curl.se/windows/>. Program sa neinštaluje, len sa rozbalí ZIP súbor na disk. Spúšťa sa program curl.exe z adresára "bin".

Príklad použitia CURL v command line alebo BAT-súbore:

curl.exe http://localhost:4321/info

CURL odošle HTTP request na zadané URL. Vrátená odpoveď sa štandardne zobrazí na Standard Output, ale je možné ju presmerovať do súboru voľbou "--output".

Návod na použitie cez "curl --help" alebo "curl --manual", podrobná dokumentácia v pod-adresári "docs".

9.1.2 Nastavenie databázy/schémy

Databázová schéma, z ktorej sa dáta exportujú, sa nastavuje podľa kombinácie parametrov COUNTRY a TEST (viď ďalej). Preto stačí, ak budú skripty pre ClientLink nainštalované len v jednom zo spustených API. User, pod ktorým je spustené príslušné API musí mať samozrejme prístup do všetkých uvedených schém.

Nastavenie schém je napevno vnútri skriptov:

Country=CZ and Test=true -> TRIGA

Country=CZ and Test=false -> TESTCZ

Country=SK and Test=true -> PPGSVK

Country=SK and Test=true -> TESTSK

Export do Testovacej (QA) alebo Produkčnej (PROD) databázy sa prepína parametrom "test=true" alebo "test=false"

9.2 Export PRODUCTS

Spustenie exportu produktov. Dáta sa načítavajú len z tabuľky TOVARY, späť do tabuľky sa zapisuje potvrdenie o odoslaní. Vyžaduje sa prístup Select a Update.

Tabuľka TOVARY musí mať vytvorené tieto X-stĺpce:

X_CRM VARCHAR2(1 BYTE), not-null hodnota (napríklad "A") znamená "exportovať" (vložiť cez TangramEOS)

X_CRMID VARCHAR2(20 BYTE), výsledok: ID produktu pridelené CRM systémom

X_CRMSENT DATE, výsledok: dátum a čas posledného úspešného exportu

X_CRMRESULT VARCHAR2(250 BYTE), výsledok: popis chyby v prípade neúspešného exportu

URL = **ppg/clipproducts?country=CZ&test=false&max=1000**

country = CZ alebo SK (musí sa zadať)

test = true alebo false (nepovinný parameter, default je true)

max = maximálny počet záznamov (nepovinný parameter, default je 1000)

9.3 Export CUSTOMERS

Spustenie exportu zákazníkov z tabuľky FIRMY.

Tento skript ešte nebol prenesený do API, bude doplnený neskôr.

9.4 Export ORDERS