Front Office - Dokumentacja API

PAR Bakuła Sp. Jawna

Integracja poprzez API - dokumentacja techniczna

Spis treści

1. Użytkowanie oprogramowania Front Office - 3
2. Interfejs API do integracji z innymi programami - 4
3. Włączenie API w domenie oraz autoryzowanie użytkownika - 4
4. Oprogramowanie potrzebne do konstruowania zapytań - 5
5. Autoryzacja i sposób konstruowania zapytania - 5
6. API - Produkty (GET) - 6
7. API - Aktualne ceny i stany magazynowe (GET) - 9
8. API - Kategorie produktów (GET) - 10
9. API - Rezerwacje (GET) - 11
10. API - Adresy wysyłki (GET)- 12
11. API - Adresy faktury - 13
12. API - Składanie zamówienia (GET) - 14
13. API - Techniki zdobienia (GET) - 16
14. API - Wersje językowe (GET) - 17
15. API - Składanie zamówienia v.1 (POST) - 18
16. API - Składanie zamówienia v.2 (POST) - 20
17. API - Przesyłanie plików graficznych z wizualizacją zdobienia (POST) - 24
18. API - Rezerwacje (POST) - 24

Użytkowanie oprogramowania Front Office

Oprogramowanie FrontOffice 3 składa się z dwóch części:

- publicznej, dostępnej dla wszystkich użytkowników sieci Internet (z niektórymi funkcjami osiągalnymi tylko dla wybranych użytkowników)
- administracyjnej, dostępnej tylko dla użytkowników posiadających określone role w systemie

Publiczna część serwisu FrontOffice to klasyczna witryna WWW połączona z zaawansowanym sklepem internetowym on-line, dostosowanym w pełni do sprzedaży gadżetów reklamowych wraz ze zdobieniem. W odróżnieniu od klasycznych sklepów internetowych, w jednym spójnym oprogramowaniu mamy zintegrowane:

- tworzenie menu oraz artykułów na stronę WWW
- tworzenie bannerów i grafik reklamowych na stronę główną
- publikowanie katalogi produktów wraz z zaawansowanymi funkcjami filtrowania i definiowania promocji
- udostępnienie koszyka zakupowego, w którym można określić technikę zdobienia i kolorystykę (otrzymując od razu wycenę zdobienia)
- udostępnienie Studia Graficznego, w którym klient sam wykona wizualizację zdobienia na stronie WWW
- newsletter pozwalający gromadzić adresy użytkowników i wysyłać do nich informacje mailowe

Każdy z Partnerów korzystających z FrontOffice 3 może zdefiniować własną domenę, w której sam określi jakie chce mieć pozycje menu na stronie głównej, jakie artykuły i dane kontaktowe prezentować klientom, jakim logiem i kolorystyką witryny chce się posługiwać, w jakim zakresie udostępniać swoim klientom narzędzia do prezentacji katalogu oraz sprzedaży gadżetów reklamowych on-line. Te i inne funkcje konfiguracyjne dostępne są w części administracyjnej oprogramowania FrontOffice 3.

Administracyjna część serwisu FrontOffice 3 to zaawansowany system zarządzania kolekcjami produktów, technikami zdobienia, materiałami graficznymi i zamówieniami. Na tej stronie właściciel domeny (lub uprawnieni przez niego użytkownicy, np. pracownicy obsługujący klientów) ma możliwość wglądu w listę zamówień, nadawanie im statusów realizacji, prowadzenie korespondencji z klientem odnośnie realizacji danego zamówienia, załączanie do zamówienia dodatkowych materiałów oraz tworzenie z zebranych zamówień tzw. zleceń zakupowych – czyli zamówień dostawy towarów, które zamówili klienci końcowi.

Oprócz tego część administracyjna pozwala na dostęp do szeregu ustawień – cenników, newsletterów, profili użytkowników, tłumaczeń itd. Jeśli do tego dodamy możliwość dodawania własnych kolekcji produktów, możliwość udostępniania własnych kolekcji innym Partnerom, płatności elektroniczne i systemy promocji to uzyskujemy potężne narzędzie sprzedażowe.

Dodatkowo istnieje możliwość integracji systemu FrontOffice 3 z oprogramowaniem magazynowym lub finansowo-księgowym funkcjonującym w firmie Partnera. Dzięki temu wszystkie zamówienia mogą od razu wpadać w tryb realizacji, a po zakończeniu tego procesu, na stronie WWW w zakładce danych klienta pojawi się odpowiedni status oraz linki do pobrania faktury lub listu przewozowego.

Interfejs API do integracji z innymi programami

API jest częścią systemu który umożliwia udostępnianie danych innym systemom bądź sklepom internetowym. API przekazuje takie informacje, jak:

- lista produktów, zdjęć, opisów, technik zdobienia;
- zdefiniowanie ceny, poziomu usług;
- ilości stanów magazynowych i planowane dostawy;
- informacje i stawki dotyczące zdobień produktów;
- listę zamówień, składanie zamówień, adresy dostaw;
- tworzenie projektów zdobień w Studiu Graficznym;

Włączenie API w domenie oraz autoryzowanie użytkownika

Wszystkie akcje API posiadają trójstopniowy system aktywacji. Dzięki temu możliwe jest określenie w każdej domenie indywidualnych zasad korzystania z danych przez interfejs programistyczny. Można przykładowo:

- uruchomić API, ale tylko w zakresie udostępniania informacji o produktach dostępnych w danej domenie;
- nadać uprawnienia tylko niektórym użytkownikom do korzystania z tej możliwości;

W pierwszej kolejności administrator domeny musi włączyć interfejs programistyczny dla swojej domeny. Aby to zrobić należy wejść do panelu administracyjnego domeny partnerskiej, następnie w edycję domeny, która ma posiadać dane ustawienia, a następnie zaznaczyć odpowiednie opcje.

W drugiej kolejności sprawdzany jest użytkownik, który próbuje wywołać metodę API. Jeśli nie poda poprawnego loginu i hasła użytkownika który istnieje w systemie i ma aktywne konto, to API zwróci błąd protokołu HTTP AUTH (403).

W trzeciej kolejności sprawdzane są uprawnienia zalogowanego użytkownika. Na tym etapie użytkownik może uzyskać odmowę skorzystania z API, gdy nie jest uprawniony do tego typu metod. Przykładowo: pobranie listy produktów dostępnych w domenie powiedzie się, ale już złożenie zamówienia spotka się z odmową. Dla innego użytkownika możemy zdefiniować zupełnie odmienny wachlarz uprawnień – od możliwości pobierania informacji o cenach i dostępności produktów, aż do zakładania rezerwacji, zamówień czy korzystania z zewnętrznego Studia Graficznego. Należy pamiętać, że w dwóch różnych domenach mogą istnieć użytkownicy o tym samym loginie (czyli identycznym adresie e-mail) ale ich ustawienia są oddzielnymi parametrami.

Podsumowując

Każdorazowo, gdy API nie działa tak jak należy warto upewnić się:

- Czy w zapytaniu podajemy ustawienia właściwej domeny?
- Czy w zapytaniu do tej domeny używamy loginu i hasła użytkownika który istnieje w tej właśnie domenie?
- Czy w ustawieniach tej konkretnej domeny konkretny użytkownik ma nadane uprawnienia do korzystania z API? Uprawnienia nadawane są przez administratora domeny PAR.

Aby można było korzystać z interfejsu programistycznego domeny, wymagane jest:

- włączenie API po stronie domeny – warunek konieczny, domyślnie API nie jest włączone;
- posiadania aktywnego konta w danej domenie, której API chcemy wykorzystywać;
- użytkownik, którym będziemy się posługiwać do autoryzacji wywołań API, musi mieć włączoną możliwość korzystania z tych narzędzi.

Oprogramowanie potrzebne do konstruowania zapytań

Aby konstruować zapytanie do API należy skorzystać z dowolnego klienta protokołu HTTP:

- przeglądarka WWW
- klient HTTP
- specjalnie napisany program (moduł sklepu internetowego korzystający z danych udostępnianych przez API)
- aplikacja Postman

Wszystkie wywołania i metody API systemu FrontOffice 3 są zgodne ze standardem REST.

Najprostszym sposobem sprawdzenia możliwości korzystania z API jest wpisanie do przeglądarki internetowej jednego z zapytań służących do pobierania danych, np. akcji /products:
`https://{adres.domeny}/products`

Po którym nastąpi prośba o autoryzację akcji i jeśli podamy dane użytkownika, który w tej domenie ma uprawnienia do korzystania z API, to zobaczymy plik w formacje JSON lub XML zawierający dane produktów domeny.

Rozwojowym formatem zapytań do API będzie format standardu REST i format JSON. Jednakże w celu zapewnienia wstecznej kompatybilności z formatem XML używanym w systemie FrontOffice 2, dla niektórych metod zwracana jest domyślnie wartość w formacie XML.

Możliwe jest podanie na końcu zapytania rozszerzenia "xml" lub "json", dzięki któremu w odpowiedzi zamiast domyślnej postaci otrzymamy plik w formacie odpowiadającym rozszerzeniu w adresie zapytania.

Podsumowując

Najprościej sprawdzić API wywołując jedną z jego metod w przeglądarce. Do zaawansowanych testów interfejsu programistycznego należy zaopatrzyć się w odpowiednie oprogramowanie, jak DHC Client lub HTTP Requester, ewentualnie napisać własny program zgodny z niżej przedstawioną dokumentacją.

Autoryzacja i sposób konstruowania zapytania

Zapytanie konstruujemy w części interfejsu zatytułowanej "REQUEST". W polu wyboru protokołu pozostawiamy wybraną opcję "HTTP", w polu edycyjnym po prawej stronie wpisujemy adres URL z zapytaniem. Niezmienną część zapytania do API będzie stanowił adres serwera i ścieżka API:

`https://{adres.domeny}.pl/api`

Następnie należy dopisać nazwę metody API, z której chcemy skorzystać, np:

`https://{adres.domeny}/api/orders`

Takie zapytanie zazwyczaj zwróci nam kolekcję obiektów odpowiadających wywołanej metodzie (w tym przypadku - kolekcję zamówień). Jeśli chcemy odwołać się do konkretnego obiektu, np. do zamówienia o identyfikatorze 2, to adres URL trzeba uszczegółowić zgodnie z dostępnymi dla danej metody API parametrami (opisanych w następnych rozdziałach), np:

`http://{adres.domeny}/api/order/2`

Każde zapytanie musi być autoryzowane przez login i hasło użytkownika zarejestrowanego w bazie użytkowników. Aby zapewnić autoryzację zapytania, należy w części interfejsu zatytułowanej "HEADERS" dodać nagłówek "Authorization" podając nasz login i hasło (po naciśnięciu "set an authorization" należy wybrać typ autoryzacji "Basic").

API - Produkty

Endpoint GET: https://{adres_domeny}/api/products

> Uwaga: endpoint zwraca listę wszystkich produktów. Przesłanie wszystkich obiektów JSON zajmuje chwilę.

Pojedyńczy produkt

Endpoint GET: https://{adres_domeny}/api/products/16

Pojedyńczy produkt - odpowiedź


```
{
  "id": 16,
  "identyfikator_produktu": 16,
  "kod": "R01022",
  "nazwa": "Etui na długopis, czarny ",
  "opis": "Czarne welurowe etui na jeden długopis, pióro lub ołówek.",
  "opis_dodatkowy": "",
  "opis_cecha": "",
  "inne": "",
  "cechy_specyficzne": "pojedyncze",
  "kategorie": [
    {
      "id": 60,
      "nazwa": "Etui"
    }
  ],
  "cena_katalogowa": "0.44",
  "cena_po_rabacie": "0.44",
  "cena_promocyjna_dla_produktu": null,
  "stan_magazynowy": 5626,
  "ilosc_dostawy": 0,
  "ilosc_dostawy_niezatwierdzonej": 0,
  "data_dostawy": null,
  "rezerwacje": 0,
  "rezerwacja_do": null,
  "jednostka_miary": "szt.",
  "wymiary": "160.0x30.0x1.0",
  "material_wykonania": "welur",
  "material_dodatkowy": null,
  "identyfikator_tabeli_koloru": 3,
  "identyfikator_koloru": 16,
  "identyfikator_koloru_kod": "02",
  "kolor_podstawowy": "czarny",
  "kolor_dodatkowy": "",
```

```
    "dostepne_kolory": [
      {
        "id": 16,
        "nazwa": "czarny",
        "rgb": "#000000",
        "quality_id": 16,
        "quality_kod": "R01022"
      }
    ],
    "towar_nowosc": false,
    "wyprzedaz": false,
    "eco": false,
    "promocja": false,
    "baterie": "",
    "techniki_zdobienia": [
      {
        "id": 9323,
        "technika_zdobienia": "sitodruk B5",
        "miejsce_zdobienia": "na wierzchu",
        "termin_realizacji": 10,
        "ilosc_kolorow": 1,
        "wymiary_zdobienia": "100x20",
        "maksymalny_rozmiar_logo": "60x15"
      }
    ],
    "zdjecia": [
      {
        "zdjecie_full": "/shared/zdjecia_katalog/full/R01022_02.jpg",
        "zdjecie_thumb": "/shared/zdjecia_katalog/przegladowka/R01022_02.png",
        "zdjecie": "/shared/zdjecia_katalog/pelne/R01022_02.png"
      }
    ],
    "opakowania": {
      "rodzaj_opakowania": "zbiorcze",
      "material_opakowania": "folia",
      "kolor_opakowania": "transparentny",
      "opakowanie_jednostkowe": {
        "waga_brutto": null,
        "waga_netto": null,
        "dlugosc": null,
        "szerokosc": null,
        "wysokosc": null
      },
      "karton_wewnetrzny": {
        "waga_brutto": null,
        "waga_netto": null,
        "ilosc": null,
        "dlugosc": null,
        "szerokosc": null,
        "wysokosc": null
      },
      "karton_duzy": {
        "waga_brutto": "13.0",
        "waga_netto": "12.0",
```

```
        "ilosc": 5000,
        "dlugosc": "320.0",
        "szerokosc": "340.0",
        "wysokosc": "250.0"
      }
    },
    "identyfikator_kolekcji": 1
  }
```

gdzie:

- `id` - (obowiązkowy) identyfikator produktu
- `identyfikator_produktu` - identyfikator produktu
- `kod` - kod magazynowy produktu
- `nazwa` - nazwa produktu
- `opis, opis_dodatkowy, opis_cecha, inne, cechy_specyficzne` - ogólne oraz szczegółowe charakterystyki produktu
- `kategorie` - tablica zawierająca informacje `id` oraz `nazwa` w odniesieniu do kategorii
- `cena_katalogowa` - cena produktu
- `cena_po_rabacie` - cena rabatowa
- `cena_promocyjna_dla_produktu` - cena promocyjna
- `stan_magazynowy` - ilość produktu dostępna w magazynie
- `ilość_dostawy` - zatwierdzona ilość zamówionego produktu
- `ilość_dostawy_niezatwierdzonej` - niezatwierdzona ilość zamówionego produktu
- `data_dostawy` - termin przekazania produktu klientowi
- `rezerwacje` - rezerwacja produktu
- `rezerwacja_do` - czas zarezerwowania produktu
- `jednostka_miary` - zastosowana jednostka miary
- `wymiary` - rozmiar produktu (długość | szerokość | wysokość)
- `materiał_wykonania` - główny materiał zastosowany w produkcie
- `identyfikator_tabeli_koloru` - wartość liczbowa identyfikatora tabeli koloru
- `identyfikator_koloru` - wartość liczbowa identyfikatora koloru
- `identyfikator_koloru_kod` - kod identyfikatora koloru
- `kolor_podstawowy` - string zwracający nazwę koloru podstawowego
- `kolor_dodatkowy` - string zwracający wartość koloru dodatkowego
- `dostępne_kolory` - tablica zwracająca wartości koloru: `id`, `nazwa`, `rgb`, `quality_id`, `quality_kod`
- `towar_nowosc` - informacja zwracająca wartość prawda / fałsz w odniesieniu do stanu towaru
- `wyprzedaz` - informacja odnosząca się do promocji produktu w ramach wybrzedaży
- `eco` - status produktu ekologiczny
- `promocja` - produkt w promocji
- `baterie` - baterie w wyposażeniu produktu
- `techniki_zdobienia` - tablica zwracająca informacje `id`, `technika_zdobienia`, `miejsce_zdobienia`, `termin_realizacji`, `ilosc_kolorów`, `wymiary_zdobienia`, `maksymalny_rozmiar_logo`
- `zdjęcia` - tablica zwracająca url zdjęć produktu w trzech rozmiarach: `zdjecie_full`, `zdjecie_thumb`, `zdjecie`
- `opakowania` - tablica zawierająca parametry i tablice odnoszące się do parametrów opakowań: `rodzaj_opakowania`, `material_opakowania`, `kolor_opakowania`,

- `opakowanie_jednostkowe` - tablica zagnieżdżona w tablicy `opakowania` zawierająca informacje: `waga_brutto`, `waga_netto`, `dlugosc`, `szerokosc`, `wysokosc`
- `karton_wewnetrzny` - tablica zagnieżdżona w tablicy `opakowania` przechowująca informacje: `waga_brutto`, `waga_netto`, `ilosc`, `dlugosc`, `szerokosc`, `wysokosc`
- `karton_duzy` - tablica zagnieżdżona w tablicy `opakowania` zawierająca informacje: `waga_brutto`, `waga_netto`, `ilosc`, `dlugosc`, `szerokosc`, `wysokosc`
- `identyfikator_kolekcji` - wartość liczbowa przechowująca dane identyfikatora

API - Aktualne ceny i stany magazynowe

Endpoint GET: `https://{adres_domeny}/api/stocks`

Zapytanie zwraca obiekt XML przekazujący informacje dotyczące stanów magazynowych oraz cen. Poniżej widok dla pojedynczego produktu.

```
<produkty>
    <produkt>
        <id>16</id>
        <kod>R01022</kod>
        <stan_magazynowy>5626</stan_magazynowy>
        <ilosc_dostawy>0</ilosc_dostawy>
        <ilosc_dostawy>0</ilosc_dostawy>
        <data_dostawy/>
        <cena_katalogowa>0.44</cena_katalogowa>
        <cena_po_rabacie>0.44</cena_po_rabacie>
    </produkt>
</produkty>
```

Aby otrzymać plik w formacje JSON należy na końcu zapytania dodać element .json - na przykład.

Endpoint GET dla pliku JSON: `https://{adres_domeny}/api/stocks.json`

Zapytanie zwraca obiekt JSON przekazujący informacje dotyczące stanów magazynowych oraz cen. Poniżej widok dla pojedynczego produktu.

```
{
  "products": [
    {
      "product": {
        "id": "16",
        "kod": "R01022",
        "stan_magazynowy": "5626",
        "ilosc_dostawy": "0",
        "ilosc_dostawy_niezatwierdzonej": "0",
        "data_dostawy": "",
        "cena_katalogowa": "0.44",
        "cena_po_rabacie": "0.44",
        "cena_promocyjna": ""
      }
```

```
    }]
}
```

gdzie:

- `products` - tablica zawierająca wszystkie obiekty produktów
- `product` - obiekt zawierający dane odnoszące się do pojedynczego produktu
- `id` - identyfikator produktu
- `kod` - kod produktu
- `stan_magazynowy` - ilość produktu dostępna na magazynie
- `ilosc_dostawy` - ilość zatwierdzonej dostawy
- `ilosc_dostawy_niezatwierdzonej` - ilość niezatwierdzonej dostawy
- `data_dostawy` - data dostawy
- `cena_katalogowa` - cena produktu bez rabatów i promocji
- `cena_po_rabacie` - cena produktu po rabacie
- `cena_promocyjna` - cena produktu w promocji

API - Kategorie produktów

Informacje odnoszącą się do kategorii produktów znajdujących się w katalogu zawiera API Categories

Endpoint GET: `https://{adres_domeny}/api/categories`

Zapytanie zwraca obiekt XML

```
<?xml version="1.0" encoding="UTF-8"?>
<categories menuoff="1">
    <category id="15" name="Czapki, Parasole" parent="1">
        <node id="27" name="Czapki" parent="15"/>
        <node id="28" name="Parasole" parent="15"/>
    </category>
</categories>
```

Endpoint GET dla obiektu JSON: `https://{adres_domeny}/api/categories.json`

```
{
  "categories": [
    {
      "category": {
        "id": "15",
        "name": "Czapki, Parasole",
        "parent": "1",
        "nodes": [
          {
            "id": "27",
            "name": "Czapki",
            "parent": "15"
          },
```

```
            {
              "id": "28",
              "name": "Parasole",
              "parent": "15"
            }
          ]
        }
      }],
  }
```

gdzie:

- `categories` - tablica zwracająca wszystkie obiekty kategorii
- `category` - pojedynczy obiekt kategorii zawierający:
- `id` - identyfikator kategorii
- `name` - nazwa kategorii
- `parent` - numer kategorii (rodzica) do której należy bieżąca kategoria
- `nodes` - tablica zawierająca dane podkategorii: `id`, `name`, `parent`

API - Rezerwacje

System FrontOffice 3 umożliwia tworzenie obiektów rezerwacja, które łączą określoną ilość towaru dostępną w magazynie z użytkownikiem (o ile ma on uprawnienia do tworzenia rezerwacji). Dzięki temu użytkownicy mogą na krótki okres (72 godziny, np. do czasu potwierdzenia zamówienia od klienta końcowego lub do chwili zaksięgowania przelewu na koncie bankowym) zarezerwować sobie pewną ilość towaru bez konieczności składania zamówienia. Po upływie tego okresu rezerwacja staje się nieaktywna i towar wraca na stan magazynowy.

Endpoint GET: `http://{adres_domeny}/api/reservations` - lista wszystkich rezerwacji w formacie XML

Endpotnt GET `http://{adres_domeny}/api/reservations.json` - lista wszystkich rezerwacji w formacie JSON

Listę można ograniczyć podając w URL-u parametry `is_active` i `valid`.

Endpoint GET: `https://{adres.domeny}/api/reservations?is_active=true&valid=2020-02-22 13:30:00` zwróci listę wszystkich rezerwacji, które będą aktywne do 22.02.2020 go godziny 13:30

Aby otrzymać listę wszystkich nieaktywnych rezerwacji używamy zapytania:

Endpoint GET: `https://{adres.domeny}/api/reservations?is_active=false`

Uwagi:

Dopuszczalne wartości parametru `is_active` to `true` i `false`. Parametr ten określa status rezerwacji, według którego chcemy ograniczyć listę. Parametr `valid` określa datę i czas, dla których chcemy pozyskać informacje o przewidywanym statusie rezerwacji (zgodnie z 72-godzinnym okresem ważności). Datę należy podać w formacie: "%Y-%m-%d %H:%M:%S", np. "2016-02-16 12:00:00". Możliwe jest opuszczenie niektórych pozycji z formatu, np. "2016-02-16" będzie interpretowane jako "2016-02-16 00:00:00", a "12:00:00" będzie

interpretowane jako godzina dla dzisiejszej daty. Podawany czas jest przez aplikację interpretowany jako lokalny.

API - Adresy wysyłki

Endpoint GET: `https://{adres_domeny}/api/addresses`

W odpowiedzi otrzymamy plik XML z listą adresów

```
<?xml version="1.0" encoding="UTF-8"?>
<addresses>
    <address id="27906">
        <nazwa_firmy_imie>Api</nazwa_firmy_imie>
        <nazwa_firmy_cd_nazwisko>Tester</nazwa_firmy_cd_nazwisko>
        <ulica>Kwiatowa</ulica>
        <kod_pocztowy>12-345</kod_pocztowy>
        <miasto>Bydgoszcz</miasto>
        <kraj>PL</kraj>
        <telefon>123456789</telefon>
        <czy_domyslny>true</czy_domyslny>
    </address>
</addresses>
```

Endpoint GET dla pliku JSON: `https://{adres_domeny}/api/addresses.json`

```
  "addresses": [
    {
      "address": {
        "id": "27906",
        "adres_firmy_imie": "Api",
        "adres_firmy_cd_nazwisko": "Tester",
        "ulica": "Kwiatowa",
        "kod_pocztowy": "12-345",
        "miasto": "Bydgoszcz",
        "kraj": "PL",
        "telefon": "123456789",
      }
    },]
```

gdzie:

- `addresses` - tablica wszystkich adresów
- `address` - pojedyńczy obiekt zawierający informacje dotyczące indywidualnego adresu:
- `id` - identyfikator adresu
- `adres_firmy_imie`- imię adresata

- `adres_firmy_cd_nazwisko`- nazwisko adresata
- `ulica` - nazwa ulicy
- `kod_pocztowy` - kod pocztowy
- `miasto` - miasto
- `kraj` - kraj
- `telefon` - telefon

API - Adresy faktury

Endpoint GET: `https://{adres_domeny}/api/addresses/invoice_address`

W odpowiedzi otrzymamy plik XML z danymi adresów na fakturze:

```
<?xml version="1.0" encoding="UTF-8"?>
<invoice_address id="4659">
    <nazwa_firmy_imie>Api</nazwa_firmy_imie>
    <nazwa_firmy_cd_nazwisko>Tester</nazwa_firmy_cd_nazwisko>
    <ulica>Kwiatowa 2/3</ulica>
    <kod_pocztowy>12-345</kod_pocztowy>
    <miasto>Bydgoszcz</miasto>
    <kraj/>
    <nip>123-456-789</nip>
</invoice_address>
```

Zapytanie GET dla pliku JSON: `https://{adres_domeny}/api/addresses/invoice_address.json`

```
{
  "invoice_address": {
    "id": "4659",
    "adres_firmy_imie": "Api",
    "adres_firmy_cd_nazwisko": "Tester",
    "ulica": "Kwiatowa 2/3",
    "kod_pocztowy": "12-345",
    "miasto": "Bydgoszcz",
    "kraj": "",
    "nip": "123-456-789"
  }
}
```

gdzie:

- `invoice_address` - obiekt przechowujący adresy na fakturze
- `id` - identyfikator faktury
- `adres_firmy_imie` - imię na fakturze
- `adres_firmy_cd_nazwisko` - nazwisko na fakturze
- `ulica` - ulica wraz z numerem domu / mieszkania
- `kod_pocztowy` - kod pocztowy
- `miasto` - miasto

- `kraj` - kraj
- `nip` - numer NIP

API - Składanie zamówienia

Endpoint GET dla wszystkich zamówień: `https://{adres_domeny}/api/orders`

> Uwaga: endpoint istnieje także w wersji api v1, w przypadku braku nagłówka `application/vnd.parfo.v2` to zapytanie zostanie przetworzone w wersji 1.

Pojedyńcze zamówienie


```
{
  "order": {
    "id": "3",
    "order_status_id": "12212",
    "cart_id": "27",
    "invoice_address_id": "",
    "numer": "603/2015",
    "data_utworzenia": "2015-05-25",
    "status": "Zamówienie anulowane",
    "data_statusu": "2016-04-20",
    "historia_zamowienia": [
      {
        "status": "Zamówienie anulowane",
        "data_statusu": "2016-04-20 13:45",
        "email": "jan.kowalski@par.com.pl"
      },
      {
        "status": "Zamówienie złożone",
        "data_statusu": "2016-04-20 13:45",
        "email": "jan.kowalski@par.com.pl"
      }
    ],
    "wartosc": "855.0",
    "koszt_wysylki": "",
    "waluta": "",
    "komentarz_do_zamowienia": "",
    "warunki_akceptacji": "",
    "custom_ident_client": "",
    "custom_ident_order": "",
    "link_do_faktury_pdf": "https://demo.royaldesign.pl/api/orders/render_pdf/3",
    "zamowione_produkty": [
      {
        "id": "3",
        "cart_item_id": "36",
        "quality_id": "68",
        "pelna_nazwa": "Czapka z daszkiem Coimbra, granatowy ",
        "kod": "R08720.04",
        "ilosc": "100",
        "cena": "5.25",
        "wartosc": "855.0",
```

```
          "czy_jest_zdobienie": "true",
          "zdobienia": [
            {
              "id": "3",
              "przygotowalnia_cena": "50.0",
              "przygotowalnia_wartosc": "200.0",
              "zdobienie_cena": "1.3",
              "zdobienie_wartosc": "130.0",
              "wartosc": "330.0",
              "liczba_kolorow": "4",
              "technika_zdobienia": "sitodruk B5",
              "miejsce_zdobienia": "na bocznym panelu",
              "termin_realizacji": "10",
              "maksymalny_rozmiar_logo": "45x45",
              "projekty": []
            }
          ]
        }
      ],
      "realizacje": []
    }
  }
```


gdzie:

- `order` - obiekt zamówienia przechowujący parametry zamówienia:
- `id` - identyfikator
- `order_status_id` - numer statusu zamówienia
- `cart_id` - identyfikator karty
- `invoice_address_id` - identyfikator faktury
- `numer` - numer zamówienia w systemie FrontOffice3
- `data_utworzenia` - data utworzenia
- `status` - status zamówienia
- `historia_zamowienia` - tablica obiektów zawierających parametry `status`, `data_statusu`, `email`
- `wartosc` - wartość zamówienia
- `koszt_wysylki` - koszt wysyłki
- `waluta` - waluta (PLN lub EUR)
- `komentarz_do_zamowienia` - uwagi klienta
- `warunki_akceptacji` - warunki akceptacji
- `custom_ident_client` -(opcjonalny) identyfikator klienta końcowego (dowolny łańcuch znaków do 255 znaków).
- `custom_ident_order` - (opcjonalny) identyfikator zamówienia klienta końcowego (dowolny łańcuch znaków do 255 znaków)
- `link_do_faktury_pdf` - link do faktury
- `zamowione_produkty` - tablica zawierająca parametry zamówionych produktów:
    - `id` - identyfikator produktu
    - `cart_item_id` - identyfikator koszyka
    - `quality_id` - identyfikator jakości produktu
    - `pelna_nazwa` - pełna nazwa produktu

    - `kod` - kod produktu
    - `ilosc` - ilość zamówionego produktu
    - `cena` - cena jednostkowa zamówionego produktu
    - `wartosc`- wartość całego zamówienia
    - `czy_jest_zdobienie` - (`true` lub `false`) - informacja czy zamówiony produkt jest ze zdobieniem
    - `zdobienia` - tablica zawierająca parametry dotyczące zdobień:
        - `id` - identyfikator zdobienia
        - `przygotowalnia_cena` - cena przygotowalnia
        - `przygotowalnia_wartosc` - wartosc przygotowalnia
        - `zdobienie_cena` - cena zdobienia
        - `wartosc` - cena za przygotowanie produktu oraz cena zdobienia
        - `liczba_kolorow` - liczba kolorów zdobienia
        - `technika_zdobienia` - technika zdobienia
        - `miejsce_zdobienia` - miejsce zdobienia
        - `termin_realizacji` - czas zdobienia (w dniach roboczych)
        - `maksymalny_rozmiar_logo` - maksymalna wielkość logo dla zamówienia
        - `projekty` - tablica zwracająca projekty
- `realizacje` - tablica zwracająca realizacje

API - Techniki zdobienia

API technics pobiera listę produktów ze zdobieniem

Endpoint GET pobierający obiekt JSON: `https://{adres_domeny}/api/technics.json`

```
[
    {
        "id": 1,
        "nazwa": "grawer",
        "kategoria": "L1",
        "kod_przygotowania": "P L",
        "termin_realizacji": 5,
        "przygotowalnia_cena": "15.0",
        "przygotowalnia_cena_eur": "3.61",
        "cennik": [
            {
                "liczba_sztuk": 1,
                "cena_pln": "15.0",
                "cena_eur": "3.61",
                "ryczalt": 1,
                "kod_zdobienia": "L1    1-9"
            }]
    }
]
```

gdzie:

- `id` - identyfikator zdobienia

- `nazwa` - nazwa zdobienia
- `kategoria` - kategoria zdobienia
- `kod_przygotowania` - kod przygotowania
- `termin_realizacji` - czas realizacji (w dniach roboczych)
- `przygotowalnia_cena` - cena przygotowania
- `przygotowalnia_cena_eur` - cena przygotowania w euro
- `cennik` - tablica zawierająca parametry produktu ze zdobieniem która zawiera obiekty z następującymi danymi:
  - `liczba_sztuk` - liczba sztuk
  - `cena_pln` - cena PLN
  - `cena_eur` - cena Euro
  - `ryczalt` - zryczałtowana cena zdobienia
  - `kod_zdobienia` - kod zdobienia

API - Wersje językowe

Przy pobieraniu listy produktów i kategorii można określić wersję językową, w jakiej będą zwracane dane. Aby uzyskać wersje w języku polskim należy wykorzystać poniższe zapytania:

Endpoint GET produkty - polska wersja językowa: `https://{adres_domeny}/pl/api/products`

Endpoint GET kategorie - polska wersja językowa: `https://{adres_domeny}/pl/api/categories`

Aby uzyskać wersję w języku angielskim należy wykorzystać poniższe zapytania:

Endpoint GET produkty - angielska wersja językowa: `https://{adres_domeny}/en/api/products`

Endpoint GET kategorie - angielska wersja językowa: `https://{adres_domeny}/en/api/categories`

Pozostałe języki:

- rosyjski - przykład - `https://{adres_domeny}/ru/api/categories`
- słowacki - przykład - `https://{adres_domeny}/sl/api/categories`

> Uwagi: język polski oraz angielski są najlepiej opracowane

Składanie zamówienia wersja API 1

Endpoint POST: `https://{adres_domeny}/api/orders`

Dla prawidłowego złożenia zamówienia należy zadbać o autoryzację oraz nagłówki:

- `Accept`, wartość: `application/xml`
- `Content-Type`, wartość: `application/json`

Treść zamówienia należy wpisać w polu edycji "BODY".

Przykład:

```
{
  "order": {
```

```
    "address_id": 3447,
    "order_note":"Proszę o wysyłkę najpóźniej 20.02.2017",
    "accept_conditions":"true",
    "currency":"PLN",
    "shipping_method_id":"2",
    "payment_method_id":"2",
    "order_items": [
      {
        "id":"1109",
        "amount":"2"
      },
      {
        "id":"11",
        "amount":"5"
      }
    ]
  }
}
```

Powyżej widoczna jest podstawowa struktura zamówienia, składanego przez API Orders.

W polu „address_id" podajemy id adresu użytkownika (adres do wysyłki – adresy już zapisane w trakcie poprzednich zakupów można pobrać przy pomocy metody /api/addresses opisanej wcześniej), w polu „order_note" komentarz do zamówienia (parametr opcjonalny), pole „accept_conditions" musi być ustawione na „true", oznacza to, że użytkownik zapoznał się z treścią i zaakceptował „Zasady i warunki współpracy z PAR BAKUŁA SP.J.".

Pole „shipping_method_id" oznacza sposób realizacji dostawy, gdzie:

1. Odbiór osobisty
2. Przesyłka kurierska

Pole „payment_method_id" oznacza sposób realizacji płatności, gdzie:

1. Przedpłata z bonusem 3% wartości zamówienia (tylko dla użytkowników z kredytem kupieckim)
2. Przelew terminowy (tylko dla użytkowników z kredytem kupieckim)
3. Przedpłata (tylko dla użytkowników bez kredytu kupieckiego)

Podanie nieprawidłowego identyfikatora sposobu realizacji dostawy lub sposobu realizacji płatności powoduje odrzucenie zamówienia.

W tablicy `order_items` podajemy kolejne produkty, na które chcemy złożyć zamówienie, wyszczególniając id danego produktu („id") oraz ilość zamówionego towaru (pole „amount").

Opcjonalnie można też podać identyfikator projektu zdobienia przypisanego do produktu (pole „project_id"). Informację o „id" produktu można znaleźć na przykład na liście produktów zwracanej przez API Products. Natomiast „address_id" można uzyskać wykorzystując opisane wcześniej API Addresses. Obowiązkowo należy podać pole „currency" („PLN" lub „EUR"). Pole „address_id" można przy składaniu zamówienia pominąć, jeśli chcemy podać adres wysyłki wprost - poniżej przykład takiego zamówienia:

```
{
  "order": {
    "address_attributes": {
      "name1":"Krzysztof Nowak",
      "name2":"Agencja Reklamowa",
      "street":"Storczykowa 3",
      "city":"Gdańsk",
      "postal_code":"12-345",
      "country":"PL",
      "phone":"123456789",
      "is_default":"0"
    },
    "order_note":"Proszę o wysyłkę najpóźniej 20.02.2016",
    "accept_conditions":"true",
    "currency":"PLN",
    "shipping_method_id":"2",
    "payment_method_id":"2",
    "order_items": [
      {
        "id":"1109",
        "amount":"2"
      },
      {
        "id":"11",
        "amount":"5"
      }
    ]
  }
}
```

W przypadku poprawnego zamówienia, jako odpowiedź otrzymamy informację o złożonym zamówieniu. Informacje te zostaną podane w formacje XML.

Jeśli w polu URL-a wpiszemy:

`https://{adres.domeny}/api/orders.json`

a zamiast nagłówka: `Accept: application/xml`, podamy `Accept: application/json` w odpowiedzi otrzymamy informację o złożonym zamówieniu podane w formacje JSON.

Składanie zamówienia w wersji v.2 API

Wersja druga rozszerza możliwości wersji pierwszej o opcjonalne elementy np. dodawanie zdobień, czy wgrywanie projektów.

Endpoint POST: `https://{adres_domeny}/api/orders`

Prawidłowa konfiguracja nagłówka dla v.2 obejmuje:

- ustawienie nagłówka `Content-Type` z wartością `application/json`
- natomiast nagłówka `Accept` z wartością `application/vnd.parfo.v2`

> **Uwaga:** W przypadku braku nagłówka `Accept` z wartością `application/vnd.parfo.v2` mamy możliwość obsługiwania zamówień w wersji v.1 oraz nie mamy dostępu do `upload_projects` - która umożliwia wgranie projektu dla zdobionego produktu.

Przykład zamówienia:

```
{
  "order": {
    "address_attributes": {
      "name1":"Krzysztof Nowak",
      "name2":"Agencja Reklamowa",
      "street":"Storczykowa 3",
      "city":"Gdańsk",
      "postal_code":"12-345",
      "country":"PL",
      "phone":"123456789",
      "is_default":"0"
    },
    "order_note":"Dodatkowa informacja dotycząca zamówienia wypełniana przez
klienta (opcjonalnie)",
    "accept_conditions":"true",
    "currency":"PLN",
    "shipping_method_id":"2",
    "payment_method_id":"2",
    "order_items": [
      {
        "id":"608",
        "amount":"2",
        "decorations": [
          {
            "technic_id": "2734",
            "number_of_colors": "1"
          },
          {
            "technic_id": "6824",
            "number_of_colors": "2"
          }
        ]
      },
      {
        "id":"673",
        "amount":"2"
      }
    ]
  }
}
```

gdzie:

- `address_attributes`, `order_note`, `accept_conditions`, `currency`,`shipping_method_id`, `payment_method_id` - analogicznie jak w wersji 1

    - `custom_ident_client` - (opcjonalny) identyfikator klienta końcowego (dowolny string do 255 znaków)
    - `custom_ident_order` - (opcjonalny) identyfikator zamówienia klienta końcowego (dowolny string do 255 znaków)
    - `order_items` - (obowiązkowa) lista produktów w zamówieniu zawiera produkty zdobione i niezdobione

Produkt bez zdobienia:

- `id` - (obowiązkowy) identyfikator produktu
- `amount` - (obowiązkowy) liczba produktów
- `reservation_id` - (opcjonalny) identyfikator rezerwacji, należy podać jeśli produkt został zarezerwowany.

Produkt ze zdobieniem:

- parametry dla produktu bez zdobienia oraz dodatkowo:
- `decorations` - (obowiązkowy) lista informacji o zdobieniu, gdzie:
- `technic_id` - (obowiązkowy) identyfikator techniki zdobienia dla produktu. Dostępne techniki zdobienia produktu są zwracane np. dla endpointa `/api/products/:id` pola `techniki_zdobienia -> id`
- `number_of_colors` - (obowiązkowy) liczba kolorów zdobienia. *Uwaga*: liczba nie może być większa niż dopuszczalna liczba kolorów dla wybranej techniki.

> `custom_ident_client` oraz `custom_ident_order` nie są przetwarzane we Frontoffice3. Format danych jest dowolny (alfanumeryczny) a długość identyfikatora nie może przekroczyć 255 znaków. Dane są przekazywane do systemu księgowego i służą do łatwiejszej identyfikacji np. przesyłek.

Odpowiedź


```
{
  "order": {
    "id": "78255",
    "order_status_id": "195110",
    "cart_id": "117090",
    "invoice_address_id": "4659",
    "numer": "14/2020",
    "data_utworzenia": "2020-02-24",
    "status": "Zamówienie złożone",
    "data_statusu": "2020-02-24",
    "historia_zamowienia": [
      {
        "status": "Zamówienie złożone",
        "data_statusu": "2020-02-24 15:05",
        "email": "api_tester@abc.com"
      }
    ],
    "wartosc": "302.0",
    "koszt_wysylki": "21.8",
    "waluta": "PLN",
```

```
    "komentarz_do_zamowienia": "Dodatkowa informacja dotycząca zamówienia
wypełniana przez klienta (opcjonalnie)",
    "warunki_akceptacji": "true",
    "custom_ident_client": "",
    "custom_ident_order": "",
    "link_do_faktury_pdf":
"https://demo.royaldesign.pl/api/orders/render_pdf/78255",
    "zamowione_produkty": [
      {
        "id": "264271",
        "cart_item_id": "425162",
        "quality_id": "673",
        "pelna_nazwa": "Notes 90x140/64k kratka Zaragoza, czarny ",
        "kod": "R64230",
        "ilosc": "2",
        "cena": "7.9",
        "wartosc": "15.8",
        "czy_jest_zdobienie": "false",
        "zdobienia": []
      },
      {
        "id": "264272",
        "cart_item_id": "425161",
        "quality_id": "608",
        "pelna_nazwa": "Ołówek stolarski, srebrny ",
        "kod": "R73792.01",
        "ilosc": "2",
        "cena": "0.6",
        "wartosc": "286.2",
        "czy_jest_zdobienie": "true",
        "zdobienia": [
          {
            "id": "33055",
            "przygotowalnia_cena": "40.0",
            "przygotowalnia_wartosc": "40.0",
            "zdobienie_cena": "55.0",
            "zdobienie_wartosc": "55.0",
            "wartosc": "95.0",
            "liczba_kolorow": "1",
            "technika_zdobienia": "tampodruk A0",
            "miejsce_zdobienia": "na boku - 1 strona",
            "termin_realizacji": "8",
            "maksymalny_rozmiar_logo": "60x7",
            "projekty": []
          },
          {
            "id": "33056",
            "przygotowalnia_cena": "40.0",
            "przygotowalnia_wartosc": "80.0",
            "zdobienie_cena": "55.0",
            "zdobienie_wartosc": "110.0",
            "wartosc": "190.0",
            "liczba_kolorow": "2",
            "technika_zdobienia": "tampodruk A0",
```

```
            "miejsce_zdobienia": "na boku - 2 strona",
            "termin_realizacji": "8",
            "maksymalny_rozmiar_logo": "60x7",
            "projekty": []
          }
        ]
      }
    ],
    "realizacje": []
  }
}
```

- `id` - identyfikator produktu
- `cart_item_id` - identyfikator pozycji karty
- `quality_id` - identyfikator jakości
- `pelna_nazwa` - pełna nazwa produktu
- `kod` - kod produktu
- `ilosc` - ilość zamawianego produktu
- `cena` - cena produktu
- `wartosc` - wartość
- `czy_jest_zdobienie` - zdobienie produktu wartość prawda lub fałsz
- `zdobienia` - tablica przekazująca szczegółowe informacje dotyczące zdobienia produktu:
- `id` - identyfikator zdobienia;
- `przygotowalnia_cena` - cena przygotowania
- `przygotowalnia_wartosc` - wartość przygotowania (całe zamówienie)
- `zdobienie_cena` - cena zdobienia
- `zdobienie_wartosc` - wartosc zdobienia (całe zamówienie)
- `wartosc` - całkowita wartość (wartość przygotowania + wartość zdobienia)
- `liczna_kolorow` - liczna zastosowanych kolorów
- `technika_zdobienia` - technika zdobienia
- `miejsce_zdobienia` - lokalizacja zdobienia na produkcie
- `termin_realizacji` - czas realizacji zamówienia
- `maksymalny_rozmiar_logo` - maksymalny rozmiar logo
- `projekty` - tablica przekazująca projekty
- `realizacje` - tablica przekazująca realizacje

Kod HTTP: 200 OK

W przypadku błędu:

```
{
  "error": "Too many colors for technic 10699 (max. 1)"
}
```

Kod HTTP: 422 Unprocessable Entity

Wszystkie błędy zwracają kody z zakresu 4xx, dodatkowy opis błędu w atrybucie `error` obiektu json

Przesłanie plików graficznych z wizualizacją zdobienia

Funkcja dostępna tylko w wersji v.2 API

Endpoint POST: `https://{adres_domeny}/api/upload_projects`

Parametry:

- `filename` - (obowiązkowy) przesyłany plik zgodnie ze standardem HTTP
- `order_id` - (obowiązkowy) identyfikator zamówienia (parametr `order -> id`) otrzymany w odpowiedzi przy składaniu zamówienia
- `decoration_id` - (obowiązkowy) - identyfikator zdobienia (parametr `order -> zamowione_produkty -> zdobienia -> id`)

Odpowiedź:

- w przypadku poprawnego przesłania pliku dla zdobienia: status HTTP 201 Created
- w przypadku błędu struktura json:

```
{
    "error": "Couldn't find Order"
}
```

wraz z kodem HTTP: 4xx (w zależności od błędu)

Rezerwacje

API Rezerwacje umożliwia zarezerwowanie produktu na 72h. Zarezerwowana ilość będzie tymczasowo zdjęta z prezentowanych stanów magazynowych i widoczna jako rezerwacja o danym numerze na stronie WWW w zakładce *Moje konto*. Identycznie jak w przypadku utworzenia rezerwacji z poziomu koszyka na stronie WWW.

Endpoint POST: `https://{adres_domeny}/api/reservations`

Dla takiego zapytania oprócz zadbania o autoryzację należy także ustawić nagłówki:

- `Accept`, wartość: `application/xml`
- `Content-Type`, wartość: `application/json`

Składnia przykładowej rezerwacji:

```
{
  "reservation": {
    "quality_id":"5",
    "amount":"2"
  }
}
```

Korzystanie z komendy POST dla rezerwacji wymaga posiadania odpowiednich uprawnień. W przypadku braku uprawnień użytkownik otrzyma błąd HTTP: `403 Forbidden`:

```
<?xml version="1.0" encoding="UTF-8"?>
<hash>
    <status>error</status>
    <message>User not allowed to access to reservations</message>
</hash>
```