

Konfigurator Lanberg Lanberg Configurator



1) Informacje wstępne

1.1) Uzyskanie dostępu

W celu umieszczenia konfiguratora Lanberg na własnej stronie internetowej należy skontaktować się z przedstawicielami marki Lanberg, aby złożyć wniosek o wygenerowanie klucza dostępu, który autoryzuje dostęp do systemu.

Klucz dostępu przypisany jest ściśle do adresu strony internetowej (na daną domenę lub subdomenę). W przypadku błędnego klucza lub domeny konfigurator nie zostanie wyświetlony.

1.2) Kontakt

W celu uzyskania dostępu do konfiguratora prosimy o kontakt pod adresem e-mail: support@lanberg.pl

2) Wdrożenie konfiguratora na stronę

2.1) Uruchomienie konfiguratora

Cały system konfiguratora znajduje się na serwerach marki Lanberg. Osoby, które chcą go umieścić na swojej stronie nie potrzebują żadnych dodatkowych modułów, wtyczek etc.

W celu osadzenia konfiguratora, w dokumencie HTML w miejscu, gdzie konfigurator ma zostać wyświetlony, należy wkleić poniższy kod zawierający iframe z uwzględnieniem języka oraz uzyskanego wcześniej klucza autoryzacyjnego:

```
<iframe src="https://configurator.lanberg.pl?key=_KEY_&locale=_LANG_" height="1000px" width="100%" style="border: 0;"></iframe>
```

Gdzie:

KEY - uprzednio uzyskany klucz autoryzacyjny do uruchomienia konfiguratora. Każdy klucz jest unikalny na domenę lub subdomenę. Uruchomienie klucza na innej stronie niż przypisana do klucza spowoduje zablokowanie uruchomienia konfiguratora.

LANG - język, który ma zostać wyświetlony w interfejsie graficznym konfiguratora oraz przy generowaniu pliku podsumowującego projekt szafy. Aktualnie obsługiwane wartości to:

- pl
- en

2.2) Obsługa biblioteki do zmiany wysokości iframe

Ze względu na różną wysokość szaf (i powiązaną z tym różną wysokością zdjęć) należy obsłużyć dynamiczną zmianę wysokości menu bocznego przy użyciu dodatkowej biblioteki związanej z iframe. Wysokość iframe będzie modyfikowana wraz ze zmianą zawartości konfiguratora np. w momencie przełączania wielkości szaf.

W celu poprawnego obsłużenia wysokości iframe należy umieścić bibliotekę **iframeResizer** i wywołać ją w kodzie JavaScript.

```
<script src="_PATH_" type="text/javascript"></script>
```

Gdzie:

PATH - ścieżka do pliku z biblioteką **iframeResizer.js**.

INFO

Możesz dołączyć tę bibliotekę w prosty sposób zamieniając **_PATH_** na ścieżkę odnoszącą się do biblioteki **iframeResizer.js** umieszczonej już na naszym serwerze. Link do biblioteki jest następujący:

<https://configurator.lanberg.pl/iframeResizer.js>

Jeżeli biblioteka jest już poprawnie ładowana, należy jeszcze ją wywołać z odpowiednimi parametrami:

```
iframeResize({  
  log: false,  
  inPageLinks: true,  
  checkOrigin: false  
});
```

2.3) Odnośniki URL do produktów

Każdy z towarów dostępny w menu może posiadać odnośnik URL do Twojej strony internetowej w postaci małej niebieskiej ikony informacji w lewej górnej krawędzi zdjęcia. Po najejchaniu i kliknięciu nastąpi przekierowanie w nowej karcie.

W celu umożliwienia klientom końcowym bezpośredniego przekierowania do produktu w Twoim systemie prosimy w e-mailu z prośbą o przydzielenie klucza autoryzacyjnego podać bezpośredni link wywoławczy do wyszukiwarki lub produktu w którym wartością zmienną będzie EAN. Przykład:

[https://lanberg.pl/search?query=\[EAN\]](https://lanberg.pl/search?query=[EAN])

INFO

Zmienna wartość **[EAN]** musi być obsługiwana w postaci jawnej / czystego tekstu na Twojej stronie internetowej.

2.4) Integracja z systemem sprzedażowym (sklep + koszyk)

W celu zapewnienia jeszcze większego komfortu korzystania z konfiguratora przez końcowych klientów, wprowadzono możliwość jego integracji z systemem sprzedażowym. Dzięki temu klient końcowy korzystając z konfiguratora będzie mógł zobaczyć dedykowane, przygotowane wyłącznie dla niego ceny na produkty z (Twojego) sklepu w czasie rzeczywistym oraz jednym kliknięciem dodać wszystkie produkty ze skonfigurowanej szafy do koszyka w (Twoim) sklepie internetowym.

Całość komunikacji odbywa się między stroną internetową (systemem sprzedażowym), na której umieszczony jest konfigurator, a iframe'em (konfiguratorem), wykorzystując prosty skrypt napisany w języku JavaScript z użyciem **JSON**, jako formatu przekazywania danych. Całość opiera się na wywołaniu i obsłudze eventu. Poniżej prezentujemy gotowy kod pozwalający na integrację konfiguratora z stroną.

INFO

Dostęp do API działa tylko wtedy gdy Lanberg przyzna uprawnienia do korzystania z niego dla danego klucza autoryzacyjnego.

```
const iframe = document.getElementById('configurator');

function bindEvent(element, eventName, eventHandler)
{
    if (element.addEventListener) {
        element.addEventListener(eventName, eventHandler, false);
    }
    else if (element.attachEvent) {
        element.attachEvent('on' + eventName, eventHandler);
    }
}

function sendMessage(msg)
{
    iframe.contentWindow.postMessage(msg, '*')
}

bindEvent(window, 'message', function (e)
{
    try {
        let data = JSON.parse(e.data);
        switch (data.type) {
            case 'getPrice':
                sendMessage(getPrices(data.items));
                break;
            case 'addToCart':
                sendMessage(addToCart(data.items));
                break;
            default:
                return false;
        }
    }
    catch (e) {
        return false;
    }
});
```

```
function getPrices(items)
{
```

INFO

Tutaj należy umieścić (Twoją) implementację funkcji, której zadaniem jest sprawdzić ceny poszczególnych produktów, które są podane w zmiennej **items** w (Twoim) systemie sprzedażowym.

Następnie należy zwrócić odpowiedni komunikat, w odpowiednim formacie danych, co omówiono w punkcie 2.4.1.1.

Dane w zmiennej **items** są prezentowane w formacie **JSON**.

```
}
```

```
function addToCart(items)
{
```

INFO

Tutaj należy umieścić (Twoją) implementację funkcji, której zadaniem jest dodać w koszyku produkty, które są podane w zmiennej **items** w (Twoim) systemie sprzedażowym.

Następnie należy zwrócić odpowiedni komunikat, w odpowiednim formacie danych, co omówiono w punkcie 2.4.1.2.

Dane w zmiennej **items** są prezentowane w formacie **JSON**.

```
}
```

2.4.1) Dostępne metody wymiany danych

2.4.1.1) Pobieranie cen produktów

Dzięki implementacji tej metody, końcowi klienci korzystając z konfiguratora będą mogli na bieżąco uzyskiwać aktualną cenę wszystkich produktów po każdym dokonaniu czynności dodawania lub usuwania akcesoriów jak i samej szafy.

Przykład prawidłowego żądania wysłanego do (Twojego) systemu sprzedażowego:

```
{
  "timestamp": 1583144382,
  "type": "getPrice",
  "items":
  [
    {
      "count": 1,
      "ean": "0000000000001",
      "manufacturerCode": "ABCD001"
    },
    {
      "count": 3,
      "ean": "0000000000002",
      "manufacturerCode": "ABCD002"
    }
  ]
}
```

Przykład prawidłowej odpowiedzi wysłanej do konfiguratora:

```
{
  "timestamp": 1583144382,
  "currency": "pln",
  "net": true,
  "type": "price",
  "items":
  {
    "00000000000001":
    {
      "prices":
      [
        {
          "count": 1,
          "price": 100.99,
          "promoText": null,
          "promoUrl": null
        }
      ],
      "productUrl": "https://lanberg.pl",
      "photoUrl": "https://lanberg.pl"
    },
    "00000000000002":
    {
      "prices":
      [
        {
          "count": 2,
          "price": 400.50,
          "promoText": null,
          "promoUrl": null
        },
        {
          "count": 1,
          "price": 350.00,
          "promoText": "Kup 3, a ostatni otrzymasz taniej!",
          "promoUrl": "https://lanberg.pl"
        }
      ],
      "productUrl": "https://lanberg.pl",
      "photoUrl": "https://lanberg.pl"
    }
  }
}
```

Gdzie:

timestamp - unikalny identyfikator zapytania-odpowiedzi w postaci liczbowej.

type - typ wymiany danych. Aktualnie obsługiwane wartości to:

- getPrice - żądanie uzyskania cen produktów.
- price - odpowiedź na żądanie uzyskania cen produktów.

items - tablica zawierająca listę produktów.

count - liczba sztuk na produkt.

ean - 13 znakowy Europejski Kod Towarowy produktu.

manufacturerCode - kod producenta produktu.

currency - waluta cen dla wszystkich produktów. Aktualnie obsługiwane wartości to:

- pln
- usd
- eur

net - określa czy kwoty produktów są kwotami netto czy brutto. Aktualnie obsługiwane wartości to:

- true - kwota netto.
- false - kwota brutto.

000000000001 - identyfikator produktu w formie kodu EAN, który jest kluczem w odpowiedzi do konfiguratora.

prices - tablica zawierająca listę cen.

price - cena za jedną sztukę, gdzie separatorem jest kropka, reszta do 2 miejsc.

promoText - nazwa promocji (jeśli istnieje) produktu. Aktualnie obsługiwane wartości to:

- null - jeśli promocja nie dotyczy produktu.
- "Nazwa promocji" - nazwa promocji (w cudzysłowach) pod którą podlega produkt.

promoUrl - link do promocji (jeśli istnieje) produktu. Aktualnie obsługiwane wartości to:

- null - jeśli promocja nie dotyczy produktu.
- "Hiperłącze do promocji" - link www do promocji (w cudzysłowach) pod którą podlega produkt.

productUrl - link do produktu.

photoUrl - bezpośredni link do głównego zdjęcia produktu.

INFO

Żądania i odpowiedzi muszą być prezentowane w formacie **JSON**.

Konfigurator przesyła listę wszystkich akcesoriów, nawet tych, które należą do firm trzecich. W takim przypadku podczas wysyłania **żądania** może nastąpić sytuacja, w której wypełniony zostanie tylko **EAN** lub **kod producenta**. Uwarunkowane jest to informacjami, jakie przekaże klient końcowy dodając (zewnętrzny) produkt.

W **odpowiedzi**, konieczne jest zwrócenie tej samej wartości **timestamp** co zapytania, którego odpowiedź dotyczy. Należy również pamiętać, że dla tablicy **items** kluczem jest kod **EAN** - nie należy używać kodu producenta. Każdy produkt może mieć podanych wiele różnych cen w zależności od wybranej ilości przez klienta końcowego.

W przypadku kwot **netto**, konfigurator może sam doliczyć VAT 23% w zależności od ustawień klienta końcowego.

Wysłanie żądania o podanie cen następuje każdorazowo po dodaniu lub usunięciu akcesoriów z szafy. Ten sposób pozwala w sposób bieżący na uwzględnianie promocji i rabatów w zależności od wybranej liczby sztuk produktów.

2.4.1.2) Dodanie produktów do koszyka

Dzięki implementacji tej metody, końcowi klienci korzystając z konfiguratora będą mogli po skonfigurowaniu szafy w prosty sposób wypełnić koszyk (Twojego) systemu sprzedażowego naciskając na dedykowany do tego przycisk w konfiguratorze. Jego uruchomienie pozwala na sprawne i szybkie wysłanie listy produktów (szafa i akcesoria), a następnie zamianę koszyka na zamówienie w (Twoim) systemie sprzedażowym.

Przykład prawidłowego żądania wysłanego do (Twojego) systemu sprzedażowego:

```
{
  "timestamp": 1583144382,
  "type": "addToCart",
  "items":
  [
    {
      "count": 1,
      "ean": "0000000000001",
      "manufacturerCode": "ABCD001"
    },
    {
      "count": 3,
      "ean": "0000000000002",
      "manufacturerCode": "ABCD002"
    }
  ]
}
```

Przykład prawidłowej odpowiedzi wysłanej do konfiguratora:

```
{
  "timestamp": 1583144382,
  "success": true,
  "type": "cart"
}
```

Gdzie:

timestamp - unikalny identyfikator zapytania-odpowiedzi w postaci liczbowej.

type - typ wymiany danych. Aktualnie obsługiwane wartości to:

- addToCart - żądanie dodania produktów do koszyka.
- cart - odpowiedź na żądanie dodania produktów do koszyka.

items - tablica zawierająca listę produktów.

count - liczba sztuk na produkt.

ean - 13 znakowy Europejski Kod Towarowy produktu.

manufacturerCode - kod producenta produktu.

success - informacja zwrotna o pomyślności dodania produktów do koszyka. Aktualnie obsługiwane wartości to:

- true - pomyślnie dodano wszystkie produkty do koszyka.
- false - wystąpił błąd uniemożliwiający dodanie nawet jednego produktu do koszyka.

INFO

Żądania i odpowiedzi muszą być prezentowane w formacie **JSON**.

W **odpowiedzi**, konieczne jest zwrócenie tej samej wartości **timestamp** co zapytania, którego odpowiedź dotyczy.

Konfigurator przesyła listę wszystkich akcesoriów, nawet tych, które należą do firm trzecich. W takim przypadku podczas wysyłania **żądania** może nastąpić sytuacja, w której wypełniony zostanie tylko **EAN** lub **kod producenta**. Uwarunkowane jest to informacjami, jakie przekaże klient końcowy dodając (zewnętrzny) produkt.

2.4.1.3) Wystąpienie błędów

W każdym momencie w (Twoim) systemie sprzedażowym mogą wystąpić błędy. Konfigurator może obsłużyć je zupełnie niezależnie od wykonywanych działań. Mogą, ale nie muszą być powiązane z żądaniami pochodzącymi z konfiguratora. W zależności od typu błędu i przyczyny powstania może on podjąć odpowiednie czynności względem zachowania systemu oraz powiadomienia klienta końcowego.

Przykład prawidłowej odpowiedzi wysłanej do konfiguratora:

```
{
  "timestamp": 1583144382,
  "type": "error",
  "relation": 0,
  "originType": "addToCart",
  "message": "Nie możemy odnaleźć wybranych produktów",
  "show": true,
  "items":
  [
    {
      "ean": "00000000000001"
    },
    {
      "manufacturerCode": "ABCD002"
    },
    {
      "ean": "00000000000003",
      "manufacturerCode": "ABCD003"
    }
  ]
}
```

Gdzie:

timestamp - unikalny identyfikator zapytania-odpowiedzi w postaci liczbowej.

type - typ wymiany danych. Aktualnie obsługiwane wartości to error.

relation - typ / relacja błędu. Aktualnie obsługiwane wartości to:

- 0 - produkt nie został znaleziony w systemie.
- 1 - produkt został znaleziony, ale nie posiada w ogóle ceny.
- 2 - produkt został znaleziony, ale nie posiada ceny dla klienta końcowego, który używa konfiguratora.
- 9 - niezależny błąd / błąd systemu.

originType - typ oryginalnego żądania, którego dotyczy odpowiedź o błędzie.

- null - jeżeli błąd nie jest związany z żądaniem wysłanym przez konfigurator.
- "addToCart" - wartość (w cudzysłowach) typu wymiany danych z żądania otrzymanego od konfiguratora.

message - zwrotna krótką i merytoryczną wiadomość (w cudzysłowach) o błędzie.

show - warunkuje (dla pewnych sytuacji) wyświetlanie komunikatu o błędzie. Parametr ten jest opcjonalny.

Aktualnie obsługiwane wartości to:

- true - komunikat będzie wyświetlony. Dla pewnych sytuacji parametr jest pomijany przez konfigurator.
- false - komunikat nie będzie wyświetlony. Dla pewnych sytuacji parametr jest pomijany przez konfigurator.

items - tablica zawierająca listę produktów. Parametr ten jest opcjonalny.

INFO

Żądania i odpowiedzi muszą być prezentowane w formacie **JSON**.

W **odpowiedzi**, konieczne jest zwrócenie tej samej wartości **timestamp** co zapytania, którego odpowiedź dotyczy. Należy również pamiętać, że dla tablicy **items** w przypadku wystąpienia błędu kluczem jest identyfikator produktu (**EAN** lub **kod producenta**) wysłany w **żądaniu**. Należy w **odpowiedzi** użyć tego samego identyfikatora produktu, który spowodował błąd. Nie dotyczy to jednak sytuacji, gdy błąd spowodowany jest przez inne czynniki nie związane z **żądaniem**.

Jeżeli w przypadku wysłania błędu użyto parametr **items**, należy pamiętać, aby w tej tablicy zdefiniować minimum 1 parametr: **ean** lub **manufacturerCode**.

O aktualne zachowanie konfiguratora dla konkretnego typu błędu można zapytać przedstawicieli marki Lanberg. Kontakt został opisany w punkcie 1.2.

Język, w którym należy wysyłać **odpowiedzi** zwrotne ustalany jest każdorazowo przy uruchomieniu konfiguratora, co opisano w punkcie 2.1. Z tego względu język nie jest dołączany do **żądania**.

1) Introductory information

1.1) Access

In order to embed Lanberg's server cabinet configurator on your own website, please contact Lanberg representatives to submit an application to receive an access key that authorizes access to the system.

The access key is strictly assigned to the address of the website (for a given domain or subdomain). In the case of an incorrect key or domain, the configurator will not be displayed.

1.2) Contact

In order to gain access to the configurator, please contact us at given e-mail address: support@lanberg.eu

2) Configurator implementation on the website

2.1) Launching the configurator

The entire server cabinet configuration system is located on Lanberg's servers. People who would like to implement it on their site do not require any additional modules, plugins, etc.

In order to embed the configurator, in the HTML document, where the configurator is to be displayed, paste the following code containing the iframe, including the language and the previously obtained authorization key:

```
<iframe src="https://configurator.lanberg.pl?key=_KEY_&locale=_LANG_" height="1000px" width="100%" style="border: 0;"></iframe>
```

Whereas:

KEY - previously obtained authorization key to run the configurator. Each key is unique to the domain or subdomain. Launching the key on a website other than the one directly assigned to the key, will result in configurator being blocked from running.

LANG - language version to be displayed in the configurator's graphic interface and when generating a design summary file of server cabinet. Currently supported values are:

- pl
- en

2.2) Library (code) support for altering the height of iframe

Due to the different height of the cabinets (and associated with that various height of pictures) you should set the dynamic change of the side menu height, using an additional library associated with the iframe. The height of the iframe will be modified in accordance with the changes of the configurator content, e.g. when switching the size of the server cabinets.

In order to correctly identify the iframe height, place the **iframeResizer** library and activate it in the javascript code.

```
<script src="_PATH_" type="text/javascript"></script>
```

Whereas:

PATH - path to file with the **iframeResizer.js** library.

NOTICE

You can include this library in a simple way by converting **_PATH_** into a path referring to the **iframeResizer.js** library already located on our server. The link to the library is as follows:

```
https://configurator.lanberg.pl/iframeResizer.js
```

If the library is already correctly loaded, it must be invoked with the appropriate parameters:

```
iFrameResize({  
  log: false,  
  inPageLinks: true,  
  checkOrigin: false  
});
```

2.3) Product URL links

Each of the goods available in the menu can have a URL link to your website in the form of a small blue information icon in the upper-left edge of the photo. After hovering and clicking on it, you will be redirected to a new tab.

To enable end-customers to proceed to the product in your system directly, please request us via E-mail to assign an authorization key, provide a direct link to the search engine or product for which EAN is the variable. For example:

```
https://lanberg.pl/search?query=[EAN]
```

NOTICE

Variable **[EAN]** must be supported on your Website in form of clear / plain text.

2.4) Integration with sales system (online store + shopping cart)

In order to grant even greater comfort for end-customers using the configurator, the possibility of cabinet configurator integration with the sales system was recently introduced. Due to this, end-customer using the configurator will be able to see dedicated, prepared exclusively for him / her prices for products from (his / her) store in real time and with just one click add all products from the configured cabinet to the shopping cart in (his / her) online store.

The whole communication takes place between the website (sales system) on which the configurator is located and the iframe (cabinet configurator), via a simple script written in JavaScript using **JSON** as a data transmission format. Integration as a whole is based on calling and handling the event. Below you can find the entire, completed code allowing you to integrate the configurator with the website.

NOTICE

Access to the API only works if Lanberg grants permission to use it for a given authorization key.

```
const iframe = document.getElementById('configurator');

function bindEvent(element, eventName, eventHandler)
{
    if (element.addEventListener) {
        element.addEventListener(eventName, eventHandler, false);
    }
    else if (element.attachEvent) {
        element.attachEvent('on' + eventName, eventHandler);
    }
}

function sendMessage(msg)
{
    iframe.contentWindow.postMessage(msg, '*')
}

bindEvent(window, 'message', function (e)
{
    try {
        let data = JSON.parse(e.data);
        switch (data.type) {
            case 'getPrice':
                sendMessage(getPrices(data.items));
                break;
            case 'addToCart':
                sendMessage(addToCart(data.items));
                break;
            default:
                return false;
        }
    }
    catch (e) {
        return false;
    }
});
```

```
function getPrices(items)
{
```

NOTICE

Here you should place (your very own) implementation of the function, which task is to check the prices of individual products, which are given through variable called **items** in (your) sales system.

Then, return the appropriate message in the appropriate data format, as discussed in point 2.4.1.1.

Data of **items** variable are presented in **JSON** data format.

```
}
```

```
function addToCart(items)
{
```

NOTICE

Here you should place (your very own) implementation of the function, which task is to add products to the shopping cart, which are given through variable called **items** in (your) sales system.

Then, return the appropriate message in the appropriate data format, as discussed in point 2.4.1.2.

Data of **items** variable are presented in **JSON** data format.

```
}
```

2.4.1) Available methods for data exchange

2.4.1.1) Downloading product prices

Due to implementation of this method, end-customers using the configurator will be able to obtain the current price of all products, on an ongoing basis, after each addition or deletion of rackmount accessories and the cabinet itself.

Below is a correct example of a request (code) sent to (your) sales system:

```
{
  "timestamp": 1583144382,
  "type": "getPrice",
  "items":
  [
    {
      "count": 1,
      "ean": "0000000000001",
      "manufacturerCode": "ABCD001"
    },
    {
      "count": 3,
      "ean": "0000000000002",
      "manufacturerCode": "ABCD002"
    }
  ]
}
```

Below is a correct example of a response sent to the cabinet configurator:

```
{
  "timestamp": 1583144382,
  "currency": "pln",
  "net": true,
  "type": "price",
  "items":
  {
    "00000000000001":
    {
      "prices":
      [
        {
          "count": 1,
          "price": 100.99,
          "promoText": null,
          "promoUrl": null
        }
      ],
      "productUrl": "https://lanberg.pl",
      "photoUrl": "https://lanberg.pl"
    },
    "00000000000002":
    {
      "prices":
      [
        {
          "count": 2,
          "price": 400.50,
          "promoText": null,
          "promoUrl": null
        },
        {
          "count": 1,
          "price": 350.00,
          "promoText": "Buy 3, and the last one you will get cheaper!",
          "promoUrl": "https://lanberg.pl"
        }
      ],
      "productUrl": "https://lanberg.pl",
      "photoUrl": "https://lanberg.pl"
    }
  }
}
```

Whereas:

timestamp - unique inquiry-response identifier in numerical form.

type - type of data exchange. Currently supported values are:

- getPrice - request for product prices.
- price - response to a request to obtain product prices.

items - table containing the list of products.

count - number of pieces per product.

ean - 13-character European Product Code.

manufacturerCode - manufacturer product code.

currency - the price currency for all products. Currently supported values are:

- pln
- usd
- eur

net - determines whether the product amounts are net or gross amounts. Currently supported values are:

- true - net amount.
- false - gross amount.

0000000000001 - product identifier in form of an EAN code, which is the key in the response to the cabinet configurator.

prices - table containing the list of prices.

price - price per one piece, where "." (dot) is a separator after which are 2 decimals.

promoText - the name of product sale / promotion (if existent). Currently supported values are:

- null - if the promotion does not apply to a given product.
- "Product sale / promotion" - the name of the sale / promotion (in quotes), under which a given product is subject.

promoUrl - URL to the sale / promotion (if existent). Currently supported values are:

- null - if the promotion does not apply to a given product.
- "Hyperlink to the sale / promotion" - URL to the sale / promotion (in quotes) under which a given product is subject.

productUrl - URL to the product.

photoUrl - direct URL to the main product photo.

NOTICE

Requests and responses must be in **JSON** format.

The configurator sends a list of all accessories, even those that belong to third parties. In this case, when sending the **request**, there may be a situation in which only the **EAN** or the **manufacturer code** will be filled. This is conditioned by the information that the end-customer will provide by adding (external) product.

In **response**, it is necessary to return the same **timestamp** value as the question to which the answer relates. Also note that for the **items** array, the key is the **EAN code** - please do not use the manufacturer code. Each product may have many different prices depending on the quantity chosen by the end-customer.

In the case of **net** amounts, the configurator can add 23% VAT depending on the end-customer's settings.

Sending a request for prices is made each time accessories are added or removed from the cabinet. This way allows you to include discounts and sales, on a current basis, depending on the selected amount in pieces of given products.

2.4.1.2) Adding products to the shopping cart

Due to the implementation of this method, end-customers using the configurator will be able to, after they are done with cabinet configuration, fill the shopping cart (of your) sales system by simply pressing the dedicated button in the configurator. Its implementation allows you to quickly and efficiently send a list of products (Rack cabinet and accessories), and then to change the shopping cart for the order in (your) online sales system.

Below is a correct example of a request (code) sent to (your) sales system:

```
{
  "timestamp": 1583144382,
  "type": "addToCart",
  "items":
  [
    {
      "count": 1,
      "ean": "0000000000001",
      "manufacturerCode": "ABCD001"
    },
    {
      "count": 3,
      "ean": "0000000000002",
      "manufacturerCode": "ABCD002"
    }
  ]
}
```

Below is a correct example of a response sent to the cabinet configurator:

```
{
  "timestamp": 1583144382,
  "success": true,
  "type": "cart"
}
```

Whereas:

timestamp - unique inquiry-response identifier in numerical form.

type - type of data exchange. Currently supported values are:

- addToCart - request to add products to the shopping cart.
- cart - response to request to add products to the shopping cart.

items - table containing the list of products.

count - number of pieces per product.

ean - 13-character European Product Code.

manufacturerCode - manufacturer product code.

success - feedback on the success of adding products to the shopping cart. Currently supported values are:

- true - all products have been successfully added to the shopping cart.
- false - an error has occurred that prevents adding even one product to the shopping cart.

NOTICE

Requests and **responses** must be in **JSON** format.

In **response**, it is necessary to return the same **timestamp** value as the question to which the answer relates.

The configurator sends a list of all accessories, even those that belong to third parties. In this case, when sending the **request**, there may be a situation in which only the **EAN** or the **manufacturer code** will be filled. This is conditioned by the information that the end-customer will provide by adding (external) product.

2.4.1.3) Occurrence of errors

At any time, errors can occur in your (sales) system. The configurator can handle them completely independently of the actions performed. They may or may not be associated with requests from the configurator. Depending on the type of error and the reason for its appearance, cabinet configurator can take appropriate actions in relation to system's behavior and notify the end customer.

Below is a correct example of a response sent to the cabinet configurator:

```
{
  "timestamp": 1583144382,
  "type": "error",
  "relation": 0,
  "originType": "addToCart",
  "message": "Nie możemy odnaleźć wybranych produktów",
  "show": true,
  "items":
  [
    {
      "ean": "0000000000001"
    },
    {
      "manufacturerCode": "ABCD002"
    },
    {
      "ean": "0000000000003",
      "manufacturerCode": "ABCD003"
    }
  ]
}
```

Whereas:

timestamp - unique inquiry-response identifier in numerical form.

type - type of data exchange. Currently supported values are error.

relation - error type / relation. Currently supported values are:

- 0 - the product was not found in the system.
- 1 - the product was found, but it does not have a price at all.
- 2 - the product was found, but it does not have a price for the end-customer who uses the configurator.
- 9 - independent error / system error.

originType - type of unique request, to which the reply concerns.

- null - if the error is not related to the request sent by the configurator.
- "addToCart" - value (in quotes) of the type of data exchange from the request received by the configurator.

message - short and substantive feedback (in quotes) regarding the error.

show - conditions (for certain situations) displaying of an error message. This parameter is optional. Currently supported values are:

- true - the message will be displayed. For certain situations, this parameter is omitted by the configurator.
- false - the message will be displayed. For certain situations, this parameter is omitted by the configurator.

items - table containing the list of products. This parameter is optional.

NOTICE

Requests and **responses** must be in **JSON** format.

In **response**, it is necessary to return the same **timestamp** value as the question to which the answer relates. Also note that for the **items** array in the event of an error, the key is the product identifier (**EAN** or **manufacturer code**) sent in the **request**. You must in **response**, use the same product identifier that caused the error. However, this does not apply if the error is caused by other factors not related to the **request**.

If in the case of sending an error in which **items** parameter was used, remember to define at least 1 parameter in this table: **ean** or **manufacturerCode**.

You can ask Lanberg representatives about the current cabinet configurator behavior for a specific type of error. For contact information, please refer to the section 1.2.

The language in which the **response** should be sent is determined each time the configurator is started, as described in section 2.1. For this reason, the language is not included in the **request**.