



INFO TECHNOLOGIES Sp. z o.o.

Dokumentacja API E-opisy.pl w wersji 2.3

System E-opisy

Ostatnia aktualizacja: 17.05.2017r.

SPIS TREŚCI

1. CEL DOKUMENTU.....	4
1.1 MISJA.....	4
1.2 ZASTOSOWANE STANDARDY.....	4
1.3 KORZYŚCI Z INTEGRACJI Z PLATFORMĄ E-OPISY.PL.....	4
2. DOSTĘPY	4
2.1 ADRESY URL DOSTĘPNYCH USŁUG	4
2.2 FUNKCJONOWANIE ŹRÓDEŁ DANYCH	4
3. ELEMENTY BAZY DANYCH SYSTEMU E-OPISY.PL	5
3.1 DRZEWO KATEGORII.....	5
3.2 PRODUKT	5
3.3 ATRYBUTY	6
4. ELEMENTY SYSTEMU E-OPISY.PL.....	6
5. ZAGADNIENIA LICENCYJNE.....	7
6. PODSTAWOWE ZAGADNIENIA INTEGRACYJNE.....	7
6.1 MAPOWANIE PRODUKTÓW	8
6.2 SPECYFIKACJE TECHNICZNE PRODUKTÓW	8
6.3 ZDJĘCIA PRODUKTU.....	9
6.4 PORÓWNYWANIE PRODUKTÓW.....	9
6.5 ETYKIETA ENERGETYCZNA	10
6.6 ZNAK WODNY.....	10
7. METODY API W FORMACIE XML DOSTĘPNE W SYSTEMIE E-OPISY.PL	11
7.1. API.GETPRODUCTDATA.....	11
7.1.1 DEKLARACJA	11
7.1.2 ARGUMENTY	11
7.1.3 WYNIK.....	11
7.2 API.GETLOGISTICSDATA	14
7.2.1 DEKLARACJA	14
7.2.2 ARGUMENTY	14
7.2.3 WYNIK.....	14
7.3 API.GETBRANDSDATA	14
7.3.1 DEKLARACJA	14
7.3.2 ARGUMENTY	14
7.3.3 WYNIK.....	14
7.4 API.SEARCHPRODUCTS	15
7.4.1 DEKLARACJA	15
7.4.2 ARGUMENTY	15
7.4.3 WYNIK.....	15
7.5 API.GETPRODUCTFILES	15
7.5.1 DEKLARACJA	15
7.5.2 ARGUMENTY	15
7.5.3 WYNIK.....	15
7.6 API.GETFILTERPARAMETERS	16
7.6.1 DEKLARACJA	16
7.6.2 ARGUMENTY.....	16



7.6.3	WYNIK	16
7.6.4	PRZYKŁADY	17
7.7	API.GETFILTERPARAMETERVALUES	17
7.7.1	DEKLARACJA	17
7.7.2	ARGUMENTY	17
7.7.3	WYNIK	17
7.7.4	PRZYKŁADY	18
7.8	API.GETFILTEREDPRODUCTS	18
7.8.1	DEKLARACJA	18
7.8.2	ARGUMENTY	18
7.8.3	WYNIK	19
7.8.4	PRZYKŁADY	19
7.9	API.CREATEMISSING	20
7.9.1	DEKLARACJA	20
7.9.2	ARGUMENTY	20
7.9.3	WYNIK	20
7.10	API.GETCATEGORIESHIERARCHY	20
7.10.1	DEKLARACJA	20
7.10.2	ARGUMENTY	20
7.10.3	WYNIK	20
7.11	API.ADDPRODUCTTOMy	21
7.11.1	DEKLARACJA	21
7.11.2	ARGUMENTY	21
7.11.3	WYNIK	21
7.12	API.MYRESOLVEDDEFECTS	21
7.12.1	DEKLARACJA	21
7.12.2	ARGUMENTY	21
7.12.3	WYNIK	21
7.13	API.GETPRODUCTSBYEAN	22
7.13.1	DEKLARACJA	22
7.13.2	ARGUMENTY	22
7.13.3	WYNIK	22
7.14	API.GETUPDATED	23
7.14.1	DEKLARACJA	23
7.14.2	ARGUMENTY	23
7.14.3	WYNIK	23
7.15	API.GETUPDATEDINMyPRODUCTS	23
7.15.1	DEKLARACJA	23
7.15.2	ARGUMENTY	23
7.15.3	WYNIK	23
8.	METODY API W FORMACIE JSON DOSTĘPNE W SYSTEMIE E-OPISY.PL	24
8.1	API.GETPRODUCTDATA	24
8.1.1	DEKLARACJA	24
8.1.2	ARGUMENTY	24
8.1.3	WYNIK	24
8.2	API.GETLOGISTICSDATA	26
8.2.1	DEKLARACJA	26
8.2.2	ARGUMENTY	26
8.2.3	WYNIK	26
8.3	API.GETBRANDSDATA	27
8.3.1	DEKLARACJA	27
8.3.2	ARGUMENTY	27



8.3.3	WYNIK	27
8.4	API.GETPRODUCTFILES	27
8.4.1	DEKLARACJA	27
8.4.2	ARGUMENTY	27
8.4.3	WYNIK	27
8.5	API.GETFILTERPARAMETERS	28
8.5.1	DEKLARACJA	28
8.5.2	ARGUMENTY	28
8.5.3	WYNIK	28
8.5.4	PRZYKŁADY	28
8.6	API.GETFILTERPARAMETERVALUES	29
8.6.1	DEKLARACJA	29
8.6.2	ARGUMENTY	29
8.6.3	WYNIK	29
8.6.4	PRZYKŁADY	29
8.7	API.GETFILTEREDPRODUCTS	30
8.7.1	DEKLARACJA	30
8.7.2	ARGUMENTY	30
8.7.3	WYNIK	30
8.7.4	PRZYKŁADY	31
8.8	API.CREATEMISSING	32
8.8.1	DEKLARACJA	32
8.8.2	ARGUMENTY	32
8.8.3	WYNIK	32
8.9	API.ADDPRODUCTTOMY	32
8.9.1	DEKLARACJA	32
8.9.2	ARGUMENTY	32
8.9.3	WYNIK	32
8.10	API.MYRESOLVEDDEFECTS	33
8.10.1	DEKLARACJA	33
8.10.2	ARGUMENTY	33
8.10.3	WYNIK	33
8.11	API.GETUPDATED	33
8.11.1	DEKLARACJA	33
8.11.2	ARGUMENTY	33
8.11.3	WYNIK	33
8.12	API.GETUPDATEDINMYPRODUCTS	34
8.12.1	DEKLARACJA	34
8.12.2	ARGUMENTY	34
8.12.3	WYNIK	34
9.	KODY BŁĘDÓW	34
10.	KONTAKT	35



1. CEL DOKUMENTU

Dokument zawiera opis interfejsu komunikacyjnego pomiędzy systemem Klienta, a Bazą Danych opisów produktów systemu E-opisy, a także zestaw standardów i formatów plików wykorzystywanych w ramach mechanizmów integracyjnych.

1.1 MISJA

Naszym celem jest dostarczenie sklepom internetowym opisów produktów o jak najwyższej jakości za pomocą interfejsu API. Aby ułatwić naszym Klientom pobieranie danych o produktach z systemu E-opisy, zastosowaliśmy standardy XML-RPC oraz JSON, które definiują zasady wymiany danych oraz ich reprezentację w formacie XML i/lub JSON.

1.2 ZASTOSOWANE STANDARDY

1. XML-RPC do wymiany danych o produktach.
2. JSON do wymiany danych o produktach.
3. UTF-8 jako kodowanie znaków wykorzystywane w systemie.
4. JPEG, GIF jako formaty wszystkich plików graficznych.

1.3 KORZYŚCI Z INTEGRACJI Z PLATFORMĄ E-OPISY.PL

Integrując się z systemem E-opisy.pl, Klient uzyskuje dostęp do ponad 6 000 producentów oraz około 800 000 opisów produktów, jakie mamy aktualnie w Bazie Danych, która jest sukcesywnie rozbudowywana o nowe produkty pojawiające się na rynku.

2. DOSTĘPY

Każdy z Klientów, który chce zintegrować się z systemem E-opisy otrzymuje imienny login i hasło do Panelu Klienta, gdzie ma możliwość zarządzania swoim koszykiem produktów. Po wygenerowaniu loginu i hasła, system generuje klucz dostępu do interfejsu API. Każdy Klient posiada swój unikalny klucz.

2.1 ADRESY URL DOSTĘPNYCH USŁUG

1. api.e-opisy.pl – adres URL usługi API.
2. klient.e-opisy.pl – adres URL Panelu Klienta.

2.2 FUNKCJONOWANIE ŹRÓDEŁ DANYCH

Wszystkie dane z systemu E-opisy udostępniane są za pomocą interfejsu programistycznego API. Na interfejs ten składa się zestaw metod, które zwracają różne elementy Bazy Danych, zorganizowane w struktury danych, które umożliwiają ich przetwarzanie i organizację.



Dostęp do API podlega ograniczeniom. Każdy z serwisów łączących się z interfejsem jest identyfikowany na podstawie adresu IP, które zostały zgłoszone przez Klienta jako adresy maszyn produkcyjnych i developerskich.

3. ELEMENTY BAZY DANYCH SYSTEMU E-OPISY.PL

3.1 DRZEWO KATEGORII

Produkty udostępniane poprzez API osadzone są w strukturze, która ułatwia zarządzanie nadawaniem dostępu do nich, poprzez system licencji oraz jest pomocna przy filtrowaniu produktów za pomocą metod API.

Struktura ta ma postać:



PRZYKŁAD

Produkt: Telefon Samsung Galaxy S5.

Struktura kategorii: Telefony i Akcesoria → Telefony → Smartfony.

3.2 PRODUKT

Produkt jest głównym elementem Bazy Danych udostępnianym przez API. Produkt rozumiany jest jako poniższe składowe:

1. Informacje podstawowe:
 - a. nazwa produktu,
 - b. model,
 - c. marka,
 - d. opis krótki,
 - e. opis marketingowy (sformatowany opis o bardziej szczegółowej treści).
2. Informacje graficzne:
 - a. zdjęcia produktu,
 - b. zdjęcia wykorzystywanych technologii,
 - c. wizualizacje 3D.
3. Informacje o gwarancji.
4. Lista atrybutów produktu rozumiana jako szczegółową, uporządkowaną specyfikację produktu.
5. Informacje logistyczne oraz KGO (Koszt Gospodarowania Odpadami), które zawierają znacznik logiczny (prawda - czyli średnia wartość dla całej kategorii, fałsz - czyli konkretna wartość, właściwa dla danego produktu). Znaczniki te pozwalają na opracowanie dodatkowych funkcjonalności w obszarach logistyki, np. koszt wysyłki danego produktu.



UWAGA

Ustawa o zużytym sprzęcie elektrycznym i elektronicznym nakazuje sprzedawcom prezentować w cenie produktu koszt gospodarowania odpadami (KGO). Baza Danych zawiera wartości kosztów gospodarowania odpadami. Wartość KGO określonego produktu jest osiągalna tylko przez identyfikator. Wysokość opłaty KGO jest wartością brutto wyrażoną w PLN. Jej wysokość jest zależna od informacji podawanych przez producenta. Dla niektórych kategorii sprzętowych producenci nie podają wartości KGO.

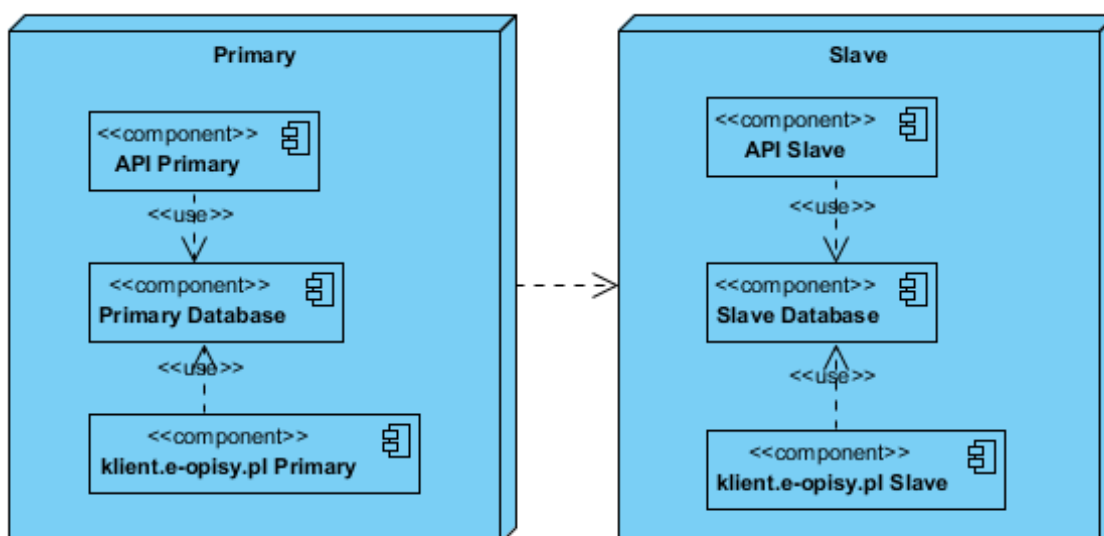
3.3 ATRYBUTY

Atrybut jest podstawowym składnikiem specyfikacji technicznej produktu. Każdy atrybut może przyjąć jedną lub więcej wartości. Na atrybut składają się:

1. grupa atrybutów,
2. nazwa atrybutu,
3. opis atrybutu - wyjaśnia jego znaczenie i zastosowanie,
4. wskazanie czy jest to atrybut podstawowy, czyli reprezentujący najistotniejsze cechy produktów w danej kategorii,
5. lista wartości atrybutu, gdzie jeden wpis zawiera:
 - a. wartość atrybutu,
 - b. opcjonalne wyjaśnienie, na które składać się może:
 - i. opis/definicję wartości, jeśli wartość tego wymaga lub jest zastosowaną w produkcie technologią,
 - ii. logotyp technologii, jeśli wartość atrybutu jest zastosowaną w produkcie technologią.

Atrybuty podstawowe pozwalają na wyeksponowanie z długiego opisu jego najważniejszych cech, a także są podstawą cech filtrujących funkcjonalności *Filtrowania Produktów*.

4. ELEMENTY SYSTEMU E-OPISY.PL



5. ZAGADNIENIA LICENCYJNE

Informacje pochodzące z Bazy Danych można podzielić na dwie grupy:

1. Informacje, których każdorazowe wyświetlenie musi być poprzedzone ich pobraniem z serwera API. Dane te nie mogą być przechowywane w lokalnych bazach danych ani innych mechanizmach typu cache (pamięć podręczna, pliki tymczasowe). Informacje takie zwracają m.in.:
 - a. pełny opis produktu przed wyświetleniem go na stronie serwisu,
 - b. elementy filtrujące dla zbudowania filtrów na stronie serwisu,
 - c. wyniki filtrowania,
 - d. wyniki wyszukiwania.
2. Informacje mogące podlegać przechowywaniu w pamięci podręcznej, czyli te, które mogą wpływać na wydajność serwisu Klienta. Do informacji tych należą:
 - a. wybrane dane opisu produktu – używane do budowania listy produktów oraz wyświetlania informacji o produkcji przy realizacji zamówień:
 - i. informacje ogólne:
 - nazwa rodzajowa produktu,
 - marka produktu,
 - pełna nazwa produktu,
 - model produktu,
 - opis skrócony,
 - miniatura zdjęcia produktu,
 - położenie produktu w strukturze kategorii,
 - dane marek,
 - dane logistyczne – używane do realizacji zamówień.

6. PODSTAWOWE ZAGADNIENIA INTEGRACYJNE

W celu przeprowadzenia integracji z Bazą Danych systemu E-opisy, należy w pierwszym kroku dokonać mapowania oferty znajdującej się w systemie Klienta z indeksami udostępnianymi przez system E-opisy. Aby przygotować plik do inicjalnego mapowania oferty należy pamiętać, aby był on w formacie .csv, oraz zawierał kolumny z kodem EAN produktu oraz jego part numberem (part number nie jest obligatoryjny).

Po przesłaniu tak przygotowanego pliku do zespołu systemu E-opisy, zostanie on w odpowiedni sposób zinterpretowany, a następnie odesłany do Klienta z dodaną kolumną o nazwie id_produktu zawierającą wewnętrzne identyfikatory produktów, które są niezbędne do pobrania specyfikacji technicznych poszczególnych produktów. Założenia dotyczące mapowania produktów zostały opisane w rozdziale 6.1 MAPOWANIE PRODUKTÓW.

Kolejne, nowododawane produkty do oferty Klienta, mogą być dodawane samodzielnie przez Klienta, z wykorzystaniem Panelu Klienta dostępnego pod adresem URL klient.e-opisy.pl.



6.1 MAPOWANIE PRODUKTÓW

Aby oprogramowanie Klienta mogło pobierać odpowiedni opis produktu, serwis Klienta musi stworzyć mapowanie identyfikatorów produktów Klienta na wewnętrzne identyfikatory produktów stosowane przez system E-opisy. Mapowanie powinno zostać przygotowane tak, aby było możliwe odwzorowanie poniższych relacji:

1. **1 : 1** - jeden produkt z Bazy Danych E-opisy mapowany na jeden produkt występujący w serwisie Klienta,
2. **1 : N** - jeden produkt z Bazy Danych E-opisy mapowany na N produktów występujących w systemie klienta.
3. **N : 1** - kilka produktów z Bazy Danych E-opisy.pl mapowanych na 1 produkt występujący w systemie klienta.

Należy pamiętać, że produkty mogą podlegać migracji pomiędzy strukturami Bazy Danych, bez zmiany swojego identyfikatora. Spowodowane jest to procesem ciągłych zmian, które mają na celu poprawę jakości danych oraz standaryzowania, co w implementacji integracji z API powinno zostać przewidziane. O takich jednostkowych migracjach (nie połączonych z relokacją, scaleniem lub podziałem kategorii) E-opisy.pl nie będą informować.

Usługodawca dysponuje Narzędziem (Panel Klienta), które umożliwia wyszukiwanie danego Produktu w celu wyszukania jego identyfikatora w Bazie Danych na potrzeby przeprowadzenia procesu zmapowania.

6.2 SPECYFIKACJE TECHNICZNE PRODUKTÓW

Prezentacja opisu produktu rozpoczyna się wywołaniem metody `getProductData(int)` opisanej w rozdziale 7.1. Otrzymane dane wymagają jeszcze przetwarzania przed ich wyświetleniem. Poszczególne atrybuty mogą być powiązane ze sobą, np. funkcjonalnie, wówczas mają taką samą wartość atrybutu „group”.

PRZYKŁAD 1

1. attr_1
 - a. group => Fizyczne
 - b. name => Waga [g]
2. attr_2
 - a. group => Fizyczne
 - b. name => Wysokość [mm]
3. attr_3
 - a. group => Fizyczne
 - b. name => Szerokość [mm]

PRZYKŁAD 2

1. attr_1
 - a. group => Przetwornik obrazu
 - b. name => Przestrzeń kolorów
2. attr_2



- a. group => Przetwornik obrazu
- b. name => Procesor obrazu

Takie atrybuty tworzą logiczną grupę w związku z tym wskazane jest aby były prezentowane w swoim sąsiedztwie.

Atrybuty produktów mogą przyjmować następujące, specyficzne wartości:

1. *brak danych* – w przypadku, gdy dla konkretnego produktu wartość danego atrybutu nie jest znana,
2. *tak/nie* – wartość logiczna, która może zostać zastąpiona dowolną infografiką reprezentującą dany symbol. Zastosowanie infografik leży po stronie systemu Klienta integrującego się z Bazą Danych E-opisy.

6.3 ZDJĘCIA PRODUKTU

Dla każdego produktu dostępne jest jedno lub więcej jego zdjęć. Pobranie zdjęć produktu możliwe jest za pomocą adresów URL zwracanych przez metodę `getProductData(int)`, która została szczegółowo opisana w rozdziale 7.1. Adresy URL odwołują się do serwerów, na których zdjęcia są przechowywane. Aby uzyskać zdjęcie w dowolnej wielkości, do uzyskanego adresu URL należy dodać parametr o nazwie `size` z żądaną wartością np. `&size=300`.

Zdjęcie, którego rozmiar nie przekracza wymiarów 150x150 pikseli, jest traktowany jako miniatura produktu możliwa do zaprezentowania np. na liście wyszukiwania. Wszystkie zdjęcia produktów (poza miniaturami) podlegają oznaczaniu poprzez znaki wodne dostarczone przez Klienta. Podstawowe zdjęcie produktu (o rozmiarze nie przekraczającym 1200x1200 pikseli) zawsze posiada odpowiadającą mu miniaturkę.

6.4 PORÓWNYWANIE PRODUKTÓW

Funkcja porównywania produktów pozwala na zapoznanie się ze specyfikacją techniczną kilku produktów należących do jednej podkategorii w sposób, który pozwoli na szybkie wychwycenie istotnych różnic w ich parametrach. API systemu E-opisy nie zostało wyposażone w dedykowaną funkcję do realizowania takiej funkcjonalności, ale dane zwracane przez metodę `getProductData()` opisaną w rozdziale 7.1, są zwracane w taki sposób, który wykonanie takiego porównania umożliwia. Opisy produktów z tej samej podkategorii pobrane przez w/w metodę posiadają atrybuty posortowane według pewnej ustalonej kolejności. W celu porównania parametrów kilku produktów należy w odpowiedni sposób zaprezentować wyniki zwracane przez system E-opisy, np. w formie tabeli.



6.5 ETYKIETA ENERGETYCZNA

Etykieta energetyczna zawiera informacje o klasie energetycznej i podstawowych parametrach urządzenia, np. zużyciu energii, poziomu hałasu. W Unii Europejskiej muszą w nią być zaopatrzone urządzenia AGD oraz źródła światła. Etykieta taka daje możliwość porównania ze sobą kilku urządzeń. Cechami wspólnymi wszystkich etykiet są:

- a. marka i model urządzenia,
- b. klasa efektywności energetycznej,
- c. zużycie innych zasobów naturalnych, np. wody,
- d. wydajność innych funkcji, np. wirowania,
- e. poziomie emitowanego hałasu.

Aby wyjść naprzeciw wymaganiom Klientów, udostępniliśmy możliwość pobrania z naszego systemu E-opisy etykiet energetycznych w formie elektronicznej. Etykiety w takiej formie można prezentować w swoim serwisie internetowym jako element karty produktu. Pliki serwowane są w formacie .jpg lub .png. Opis sposobu pobierania etykiet energetycznych znajduje się w rozdziale 7.5. Za pobieranie etykiet odpowiada metoda getProductFiles().

6.6 ZNAK WODNY

Znak wodny jest graficznym elementem, który nakładany jest na wszystkie zdjęcia stanowiące część specyfikacji technicznej produktu. Znak wodny musi trwale oznaczać zawartość zdjęcia, tak aby nie było możliwe jego usunięcie bez szkody dla zawartości zdjęcia i tym samym ukrycie rzeczywistego pochodzenia zdjęcia. Znak wodny nie powinien jednocześnie całkowicie zasłaniać fragmentu zdjęcia uniemożliwiając dostrzeżenie istotnych cech przedstawianego produktu. Dlatego też znak wodny nakładany jest na zdjęcie centralnie z zastosowaniem wysokiego stopnia przezroczystości, który zapewnia zarówno trwałe oznaczanie, jak i dostateczną widoczność produktu.

W przypadku, gdy znak graficzny Klienta naturalnie składa się z jednego koloru lub występujące kolory są w podobnej gamie kolorystycznej, wymagane jest wprowadzenie dodatkowego kontrastowego koloru, np. poprzez efekt apli, czyli dodanie tła do logo, albo efekt outline-u, czyli otoczenie aktualnego znaku obramowaniem w kontrastowym kolorze. Wymóg ten związany jest z faktem, że zdjęcia udostępniane wraz z opisami przedstawiają produkty w bardzo zróżnicowanej kolorystyce.

Jeżeli logo będzie składać się tylko z jednego koloru lub kolorów z tej samej gamy kolorystycznej oraz oznaczany będzie produkt o podobnej kolorystyce, wymóg oznaczenia zdjęcia nie zostanie osiągnięty, a dostarczone logo odrzucone.



Logo nakładane jest na zdjęcia o różnych rozmiarach. Dla zachowania satysfakcjonującej jakości znaku wodnego istotne jest, aby nie miała miejsca operacja skalowania „w górę”. Polega ona na rozciągnięciu logo, tak aby przykrywał odpowiednią część zdjęcia. Skalowanie tego typu wiąże się zwykle ze spadkiem jakości skalowanej grafiki. Dlatego też logo powinno być dostatecznie duże, aby zawsze miało miejsce tzw. skalowanie „w dół”. Algorytm nakładania znaku wodnego jest zoptymalizowany pod kątem logo w formie poziomego napisu (współczynnik proporcji szerokość/wysokość równy 4/3). Dla tego typu logo najmniejsza zalecana szerokość wynosi 800px.

Logo powinno być przesłane w formacie .gif z ustawionym kolorem tła na przezroczysty. Inne formaty plików nie będą przyjmowane.

Jeżeli stosowanie logo jest obłożone ograniczeniami wynikającymi z identyfikacji wizualnej, to należy przekazać do INFO TECHNOLOGIES Sp. z o.o. odpowiednią księgę znaku. Zwykle jednak nie jest możliwe zastosowanie się do wszystkich wskazówek zawartych w tego typu instrukcji. Traktują one bowiem o użyciu znaku graficznego firmy, a nie o stosowaniu logo jako znaku wodnego. W takiej sytuacji prosimy o wskazanie, które z obostrzeń dotyczących stosowania logo są niezbędne do zachowania (np.: strefa ochronna, zniekształcenia itd.), a które z nich można pominąć i w jakim zakresie.

7. METODY API W FORMACIE XML DOSTĘPNE W SYSTEMIE E-OPISY.PL

7.1. *api.getProductData*

7.1.1 DEKLARACJA

```
array = api.getProductData(int)
array = api.getProductData(string)
```

7.1.2 ARGUMENTY

Metoda wymaga jednego argumentu, którym powinna być liczba całkowita (typ XML-RPC: int) albo ciąg znaków (typ XML-RPC: string).

Wartość ta powinna być prawidłowym identyfikatorem produktu.

7.1.3 WYNIK

parametr	typ	znaczenie
product_id	int	identyfikator produktu
	string	
type	string	nazwa rodzajowa produktu (np.: Telefon)
brand	string	marka produktu (np.: Samsung)

name	string	nazwa produktu (np.: SAMSUNG Galaxy S5)
trade_name	string	nazwa handlowa
model	string	model produktu
part_number	string	kod producenta
category_path	string	lokalizacja produktu w strukturze kategorii w formacie 'dział/kategoria/podkategoria'
short_description	string	skrótowy opis produktu
multimedia_description	struct	marketingowy opis produktu

Tablica multimedia_description zawiera strukturę (typ XML-RPC: struct) o następującej zawartości:		
parametr	typ	znaczenie
content	string	formatowany opis produktu, który może zawierać elementy multimedialne, takie jak grafiki, filmy itp.
stylesheetlink	string	adres URL do deklaracji stylu CSS używanych w opisie multimedialnym

verify	string	kod html do wklejenia na stronie serwisu internetowego wyświetlającej pobrany opis produktu, w celu umożliwienia kontroli poprawności korzystania z Bazy Danych
EANs	array	Kody EAN produktów
media	struct	zdjęcia, wizualizacje, itp.

Tablica media zawiera strukturę (typ XML-RPC: struct) o następującej zawartości:		
parametr	typ	znaczenie
photos	array	zdjęcia produktu

Elementy tablicy photos zawierają struktury (typ XML-RPC: struct) o następującej zawartości:		
parametr	typ	znaczenie
link	string	adres URL do zdjęcia produktu
		aby uzyskać zdjęcie w dowolnej wielkości, do otrzymanego adresu należy dodać parametr size z żadaną wartością
		np: &size=333
type	string	typ zdjęcia:
		miniatura bez znaku wodnego, ograniczone od wielkości 150x150
		duże zdjęcia produktu/widoku/kadru/etc. ze znakiem wodnym, ograniczone od wielkości 1200x1200
press_releases	array	informacje prasowe dostępne dla produktu

Elementy tablicy press_releases zawierają struktury (typ XML-RPC: struct) o następującej zawartości:		
parametr	typ	znaczenie
link	string	adres URL zawartości informacji prasowej wstępnie sformatowane tagami html, przygotowane do ostylowania css.
type	string	typ informacji prasowej
date	ISO 8601 DateTime	data wprowadzenia informacji prasowej w formacie YYYYMMDDTHH:MM:SS.

	visualisations	array	prezentacje/wizualizacje 3D
	Elementy tablicy visualisations zawierają struktury (typ XML-RPC: struct) o następującej zawartości:		
	parametr	typ	znaczenie
	link	string	kod html do wklejenia na stronie Serwisu Internetowego wyświetlającej pobrany opis produktu, osadzający wizualizację/animację 3D jeśli jest dostępna dla wybranego produktu, wg specyfikacji
	type	string	typ wizualizacji/animacji 3D
	attributes	array	cechy produktu
	Tablica attributes zawiera elementy typu struktura (typ XML-RPC: struct) o następującej zawartości		
	parametr	typ	znaczenie
	group	string	grupa atrybutu
	name	string	nazwa atrybutu
	primary	boolean	czy atrybut podstawowy?
	description	string	opis atrybutu
	values	array	wartości atrybutu
	Elementy tablicy values zawierają elementy struct (typ XML-RPC: struct) stanowiących kolejne wartości atrybutu.		
	parametr	typ	znaczenie
	value	string	wartość atrybutu
	explanation	struct	opcjonalne wyjaśnienie wartości atrybutu
	Dla wybranych wartości atrybutów, pole explanation zawierających użyte technologie wartości atrybutów będą strukturami (typ XML-RPC: struct) o następującej zawartości:		
	parametr	typ	znaczenie
	logo	string	adres URL do grafiki obrazującej wartość atrybutu, np.: logotyp użytej technologii
	description	string	wyjaśnienie wartości atrybutu, np.: wyjaśnienie użytej technologii

7.2 *api.getLogisticsData*

7.2.1 DEKLARACJA

```
array = api.getLogisticsData(int)
array = api.getLogisticsData(string)
```

7.2.2 ARGUMENTY

Metoda wymaga jednego argumentu, którym powinna być liczba całkowita (typ XML-RPC: int) albo ciąg znaków (typ XML-RPC: string).
Wartość ta powinna być prawidłowym identyfikatorem produktu.

7.2.3 WYNIK

parametr	typ	znaczenie
product_id	int	identyfikator produktu
	string	
weight_gross	struct	wartość wagi brutto
kgo_gross	struct	wartość kgo brutto
Pola weight_gross , kgo_gross są strukturami (typ XML-RPC: struct) z następującymi elementami:		
parametr	typ	znaczenie
value	string	wartość elementu
is_default	boolean	czy wartość jest domyślna dla kategorii czy dla konkretnego produktu (true - wartość uśredniona dla kategorii)

7.3 *api.getBrandsData*

7.3.1 DEKLARACJA

```
array = api.getBrandsData()
```

7.3.2 ARGUMENTY

Brak.

7.3.3 WYNIK

parametr	typ	znaczenie
brand id	int	numer id marki
name	string	nazwa marki (np.: NIKON)



logo	string	adres URL obrazka przedstawiajacego logo marki
------	--------	--

7.4 *api.searchProducts*

7.4.1 DEKLARACJA

```
array = api.searchProducts(string)
```

7.4.2 ARGUMENTY

Metoda wymaga jednego argumentu, którym powinien być ciąg znaków (typ XML-RPC: string). Ciąg znaków będzie traktowany jako wyrażenie, pod kątem którego należy przeszukiwać Bazę Danych w poszukiwaniu produktów. Ciąg znaków może zawierać spacje.

7.4.3 WYNIK

parametr	typ	znaczenie
category_path	string	położenie produktu w strukturze kategorii w formacie dział/kategoria/podkategoria
product_id	int	identyfikator produktu
	string	
product_name	string	pełna nazwa produktu w formacie Typ MARKA Nazwa
		np.: Telefon Samsung Galaxy S5
trade_name	string	nazwa handlowa
part_number	string	kod producenta

7.5 *api.getProductFiles*

7.5.1 DEKLARACJA

```
array = api.getProductFiles(int)
```

7.5.2 ARGUMENTY

Metoda wymaga jednego argumentu, a jest nim dedykowany identyfikator produktu z Bazy Danych E-opisy.

7.5.3 WYNIK

parametr	typ	znaczenie
file_id	int	identyfikator produktu z bazy danych systemu e-opisy
file_type	int	Typ przekazywanego pliku: 3 - schemat montażowy 12 - instrukcja obsługi 14 - certyfikaty 16 - specyfikacja 15 - plik wewnętrzny 17 - etykieta energetyczna 18 - karta produktu
path	string	pełna ścieżka lokalizacji pliku
file_format	string	format pliku

7.6 api.getFilterParameters

7.6.1 DEKLARACJA

array = api.getFilterParameters(string)

7.6.2 ARGUMENTY

Metoda wymaga jednego argumentu i jest nim ścieżka do podkategorii, w której chcemy wyszukać dany produkt. Ścieżka kategorii jest rozumiana jako struktura w systemie e-opisy.

7.6.3 WYNIK

Metoda zwraca listę dostępnych marek i atrybutów, które dostępne są jako elementy filtrujące.

parametr	typ		znaczenie	
category_path	string		ścieżka kategorii	
category_id	int		identyfikator kategorii	
filters	array		pełna ścieżka do pliku	
	parametr	typ		znaczenie
	brands	array		tablica dostępnych marek produktów
		parametr	typ	znaczenie
		id	int	identyfikator marki
		name	string	nazwa marki
	attributes	array		tablica dostępnych atrybutów
		Parametr	typ	Znaczenie
		id	int	identyfikator atrybutu
		name	string	nazwa atrybutu

7.6.4 PRZYKŁADY

Komputery i akcesoria/Laptopy i tablety/Notebooki, Netbooki, Ultrabooki"

wynik:

Lista marek i cech możliwych do przefiltrowania.

```
category_path => Komputery i akcesoria/Laptopy i tablety/Notebooki, Netbooki, Ultrabooki,  
category_id => 168,  
filters => [  
  brands => [  
    ...  
    31 => [  
      id => 480  
      name => HP  
    ],  
    ...  
    55 => [  
      id => 5614  
      name => VORDON  
    ]  
    ...  
  ],  
  attributes => [  
    0 => [  
      id => 51718,  
      name => Rodzaj  
    ],  
    ...  
    19 => [  
      id => 7011,  
      name => Przekątna ekranu (cal)  
    ],  
    ...  
  ]  
]
```

7.7 *api.getFilterParameterValues*

7.7.1 DEKLARACJA

array = api.getFilterParameterValues(int, int)

7.7.2 ARGUMENTY

Metoda wymaga dwóch argumentów i są nimi id podkategorii oraz id atrybutu.

7.7.3 WYNIK

Metoda zwraca listę atrybutów umożliwiających filtrowanie.



parametr	typ	znaczenie
category_id	int	identyfikator kategorii
attribute_id	int	identyfikator atrybutu
values	array	tablica atrybutów umożliwiających filtrowanie

7.7.4 PRZYKŁADY

```
category_id => 168,
attribute_id => 7011,
values => [
    0 => 13.3,
    ...
    3 => 15.6,
    ...
    32 => 15
    ...
]
```

7.8 *api.getFilteredProducts*

7.8.1 DEKLARACJA

`array = api.getFilteredProducts(struct)`

7.8.2 ARGUMENTY

Metoda wymaga argumentów zgodnie z poniższym przykładem:

```
category_path => sciezka kategorii,
parameters => [
    page => kolejna strona wyniku. Opcjonalne. Domyślnie 1,
    perPage => ilość wyników na stronie. Opcjonalne. Domyślnie 2000. Nie więcej
    niż 2000.
],
filters => [
    brands => [
        id marki 1,
        id marki 2,
        ....
    ],
    attributes => [
        [
            attribute_id => id atrybutu 1,
            values => [
                pierwsza wartość atrybutu 1,
                druga wartosc atrybutu 1,
                ...
            ]
        ],
        [
            attribute_id => id atrybutu 2,
            values => [
                pierwsza wartość atrybutu 2,
                druga wartosc atrybutu 2,
                ...
            ]
        ]
    ]
]
```



```
]
]
```

7.8.3 WYNIK

Metoda zwraca listę produktów spełniających zadane kryteria.

parametr	typ	znaczenie
category_path	string	ścieżka kategorii
total	Int	ilość produktów spełniających zadane kryteria
page	Int	numer zwróconej strony
perPage	int	ilość wyników na stronę
products	array	lista produktów spełniających dane kryteria
	parametr	typ
product_id	int	identyfikator produktu
	string	
name	string	nazwa produktu
trade_name	String	nazwa handlowa
type	String	typ produktu
brand	String	nazwa marki
model	String	model produktu
part_number	String	part numer produktu
short_description	String	krótki opis produktu
photo	string	link do miniatury produktu

7.8.4 PRZYKŁADY

```
category_path => Komputery i akcesoria/Laptopy i tablety/Notebooki, Netbooki, Ultrabooki,
total => 1639,
page => 1,
perPage => 10,
products => [
  0 => [
    product_id => 17528,
    name => nx9105,
    type => Notebook,
    brand => HP,
    model =>
    part_number => PG690EA,
    short_description =>
    photo => http://api.e-
opisy.pl/img/getProductMiniature.php?product_id=17528
  ],
  ....
  9 => [
    product_id => 20065,
    name => nx6125,
    type => Notebook,
    brand => HP,
    model =>
    part_number => PY416EA,
    short_description =>
    photo => http://api.e-
opisy.pl/img/getProductMiniature.php?product_id=20065
```



7.9 *api.createMissing*

7.9.1 DEKLARACJA

```
array = api.createMissing(struct)
```

7.9.2 ARGUMENTY

Metoda przyjmuje następujące argumenty:

parametr	typ	znaczenie
name	string	nazwa produktu
ean	string	kod EAN produktu
pn	string	part number produktu
producer	string	nazwa producenta
note	string	notatka

7.9.3 WYNIK

Wynikiem wywołania metody jest zwrócenie ID braku. W przypadku gdy zostanie zwrócony inny tekst, wywołanie metody nie powiodło się.

7.10 *api.getCategoriesHierarchy*

7.10.1 DEKLARACJA

```
array = api.getCategoriesHierarchy()
```

7.10.2 ARGUMENTY

Brak.

7.10.3 WYNIK

Metoda zwraca listę atrybutów umożliwiających filtrowanie.



parametr	typ		znaczenie
id	string		id kategorii
name	string		nazwa kategorii
children	struct		kategoria podrzędna
	Parametr	typ	znaczenie
	id	string	id kategorii
	name	string	nazwa kategorii
	children	struct	kategoria podrzędna
	parametr	typ	znaczenie
	id	string	id kategorii
	name	string	nazwa kategorii
	children	struct	kategoria podrzędna

7.11 *api.addProductToMy*

7.11.1 DEKLARACJA

array = api.addProductToMy(int)

7.11.2 ARGUMENTY

Metoda wymaga jednego argumentu, a jest nim dedykowany identyfikator produktu z Bazy Danych E-opisy.

7.11.3 WYNIK

Wynikiem wywołania metody jest zwrócenie komunikatu: „Product has been added”. W przypadku, gdy zostanie zwrócony inny tekst, wywołanie metody nie powiodło się.

7.12 *api.myResolvedDefects*

7.12.1 DEKLARACJA

array = api.myResolvedDefects(ISO 8601 DateTime)

7.12.2 ARGUMENTY

Metoda wymaga wpisania daty i godziny w następującym formacie: rrrr-mm-dd hh:mm:ss

7.12.3 WYNIK

Wynikiem wywołania metody jest zwrócenie listy maksymalnie 2000 rozwiązanych zgłoszeń typu BrP (brak produktu).



parametr	typ	znaczenie
defect_id	int	id braku
product_id	int	id produktu
	string	
reporte_date	ISO 8601 DateTime	data zgłoszenia
reporte_note	string	notatka zgłoszenia
resolve_date	ISO 8601 DateTime	data rozwiązania zgłoszenia
resolve_note	string	notatka rozwiązania zgłoszenia
category_id	int	id kategorii
EAN	int	kod EAN
name	struct	nazwa

7.13 api.getProductsByEAN

7.13.1 DEKLARACJA

```
array = api.getProductsByEAN(int)
array = api.getProductsByEAN(string)
```

7.13.2 ARGUMENTY

Metoda wymaga jednego argumentu, którym powinna być liczba całkowita (typ XML-RPC: int) albo ciąg znaków (typ XML-RPC: string).

7.13.3 WYNIK

Wynikiem wywołania metody jest zwrócenie listy produktów, które mają wskazany kod EAN.

parametr	typ	znaczenie
product_id	int	id produktu
	string	
name	string	nazwa produktu
trade_name	String	nazwa handlowa
model	string	model produktu
pn	int	part numer produktu
brand	string	marka produktu
type	string	typ produktu
EAN description	Int	opis kodu EAN produktu

7.14 *api.getUpdated*

7.14.1 DEKLARACJA

array = api.getUpdated(ISO 8601 DateTime)

7.14.2 ARGUMENTY

Metoda wymaga wpisania daty w następującym formacie: rrrr-mm-dd

7.14.3 WYNIK

Wynikiem wywołania metody jest zwrócenie listy zaktualizowanych po wskazanej dacie produktów. Wyświetlone zostaną tylko te produkty, w które uległy modyfikacji maksymalnie 30 dni po wskazanej dacie.

7.15 *api.getUpdatedInMyProducts*

7.15.1 DEKLARACJA

array = api.getUpdatedInMyProducts(ISO 8601 DateTime)

7.15.2 ARGUMENTY

Metoda wymaga wpisania daty w następującym formacie: rrrr-mm-dd

7.15.3 WYNIK

Wynikiem wywołania metody jest zwrócenie listy produktów znajdujących się w moich produktach zaktualizowanych po wskazanej dacie. Wyświetlone zostaną tylko te produkty, które były edytowane lub dodane do moich produktów maksymalnie 30 dni po wskazanej dacie.



8. METODY API W FORMACIE JSON DOSTĘPNE W SYSTEMIE E-OPISY.PL

8.1 *api.getProductData*

8.1.1 DEKLARACJA

```
array = api.getProductData(int)
array = api.getProductData(string)
```

8.1.2 ARGUMENTY

Metoda wymaga jednego argumentu, którym powinna być liczba całkowita (int) albo ciąg znaków (string).

Wartość ta powinna być prawidłowym identyfikatorem produktu.

8.1.3 WYNIK

parametr	typ	znaczenie									
product_id	int	identyfikator produktu									
	string										
type	string	nazwa rodzajowa produktu (np.: Telefon)									
brand	string	marka produktu (np.: Samsung)									
name	string	nazwa produktu (np.: SAMSUNG Galaxy S5)									
trade_name	String	nazwa handlowa									
model	string	model produktu									
part_number	string	kod producenta									
category_path	string	lokalizacja produktu w strukturze kategorii w formacie 'dział/kategoria/podkategoria'									
short_description	string	skrótowy opis produktu									
multimedia_description	struct	marketingowy opis produktu									
Tablica multimedia_description zawiera strukturę (typ XML-RPC: struct) o następującej zawartości:											
	<table><tr><th>parametr</th><th>typ</th><th>znaczenie</th></tr><tr><td>content</td><td>string</td><td>formatowany opis produktu, który może zawierać elementy multimedialne, takie jak grafiki, filmy itp.</td></tr><tr><td>stylesheetlink</td><td>string</td><td>adres URL do deklaracji stylu CSS używanych w opisie multimedialnym</td></tr></table>	parametr	typ	znaczenie	content	string	formatowany opis produktu, który może zawierać elementy multimedialne, takie jak grafiki, filmy itp.	stylesheetlink	string	adres URL do deklaracji stylu CSS używanych w opisie multimedialnym	
parametr	typ	znaczenie									
content	string	formatowany opis produktu, który może zawierać elementy multimedialne, takie jak grafiki, filmy itp.									
stylesheetlink	string	adres URL do deklaracji stylu CSS używanych w opisie multimedialnym									
verify	string	kod html do wklejenia na stronie serwisu internetowego wyświetlającej pobrany opis produktu, w celu umożliwienia kontroli poprawności korzystania z Bazy Danych									
EANs	array	Kody EAN produktów									



media	struct	zdjęcia, wizualizacje, itp.
Tablica media zawiera strukturę (typ XML-RPC: struct) o następującej zawartości:		
parametr	typ	znaczenie
photos	array	zdjęcia produktu
Elementy tablicy photos zawierają struktury (typ XML-RPC: struct) o następującej zawartości:		
parametr	typ	znaczenie
link	string	adres URL do zdjęcia produktu
		aby uzyskać zdjęcie w dowolnej wielkości, do otrzymanego adresu należy dodać parametr size z żadaną wartością
		np: &size=333
type	string	typ zdjęcia:
		miniatura bez znaku wodnego, ograniczone od wielkości 150x150
		duże zdjęcie produktu/widoku/kadru/etc. ze znakiem wodnym, ograniczone od wielkości 1200x1200
press_releases	array	informacje prasowe dostępne dla produktu
Elementy tablicy press_releases zawierają struktury (typ XML-RPC: struct) o następującej zawartości:		
parametr	typ	znaczenie
link	string	adres URL zawartości informacji prasowej wstępnie sformatowane tagami html, przygotowane do ostylowania css.
type	string	typ informacji prasowej
date	ISO 8601 DateTime	data wprowadzenia informacji prasowej w formacie YYYYMMDDTHH:MM:SS.
visualisations	array	prezentacje/wizualizacje 3D
Elementy tablicy visualisations zawierają struktury (typ XML-RPC: struct) o następującej zawartości:		
parametr	typ	znaczenie
link	string	kod html do wklejenia na stronie Serwisu Internetowego wyświetlającej pobrany opis produktu, osadzający wizualizację/animację 3D jeśli jest dostępna dla wybranego produktu, wg specyfikacji
		typ wizualizacji/animacji 3D
attributes	array	cechy produktu
Tablica attributes zawiera elementy typu struktura (typ XML-RPC: struct) o następującej zawartości		
parametr	typ	znaczenie
group	string	grupa atrybutu
name	string	nazwa atrybutu
primary	boolean	czy atrybut podstawowy?
description	string	opis atrybutu

values	array	wartości atrybutu
Elementy tablicy values zawierają elementy struct (typ XML-RPC: struct) stanowiących kolejne wartości atrybutu.		
parametr	typ	znaczenie
value	string	wartość atrybutu
explanation	struct	opcjonalne wyjaśnienie wartości atrybutu
Dla wybranych wartości atrybutów, pole explanation zawierających użyte technologie wartości atrybutów będą strukturami (typ XML-RPC: struct) o następującej zawartości:		
parametr	typ	znaczenie
logo	string	adres URL do grafiki obrazującej wartość atrybutu,
		np.: logotyp użytej technologii
description	string	wyjaśnienie wartości atrybutu,
		np.: wyjaśnienie użytej technologii

8.2 api.getLogisticsData

8.2.1 DEKLARACJA

```
array = api.getLogisticsData(int)
array = api.getLogisticsData(string)
```

8.2.2 ARGUMENTY

Metoda wymaga jednego argumentu, którym powinna być liczba całkowita (typ XML-RPC: int) albo ciąg znaków (typ XML-RPC: string).
Wartość ta powinna być prawidłowym identyfikatorem produktu.

8.2.3 WYNIK

parametr	typ	znaczenie
product_id	int	identyfikator produktu
	string	
weight_gross	struct	wartość wagi brutto
kgo_gross	struct	wartość kgo brutto
Pola weight_gross , kgo_gross są strukturami (typ XML-RPC: struct) z następującymi elementami:		
parametr	typ	znaczenie
value	string	wartość elementu
is_default	boolean	czy wartość jest domyślna dla kategorii czy dla konkretnego produktu (true - wartość uśredniona dla kategorii)

8.3 *api.getBrandsData*

8.3.1 DEKLARACJA

```
array = api.getBrandsData()
```

8.3.2 ARGUMENTY

Brak.

8.3.3 WYNIK

parametr	typ	znaczenie
brand id	int	numer id marki
name	string	nazwa marki (np.: NIKON)
logo	string	adres URL obrazka przedstawiającego logo marki

8.4 *api.getProductFiles*

8.4.1 DEKLARACJA

```
array = api.getProductFiles(int)
```

8.4.2 ARGUMENTY

Metoda wymaga jednego argumentu, a jest nim dedykowany identyfikator produktu z Bazy Danych E-opisy.

8.4.3 WYNIK

parametr	typ	znaczenie
file_id	int	identyfikator produktu z bazy danych systemu e-opisy
file_type	int	Typ przekazywanego pliku: 3 - schemat montażowy 12 - instrukcja obsługi 14 - certyfikaty 16 - specyfikacja 15 - plik wewnętrzny 17 - etykieta energetyczna 18 - karta produktu
path	string	pełna ścieżka lokalizacji pliku
file_format	string	format pliku



8.5 api.getFilterParameters

8.5.1 DEKLARACJA

```
array = api.getFilterParameters(string)
```

8.5.2 ARGUMENTY

Metoda wymaga jednego argumentu i jest nim ścieżka do podkategorii, w której chcemy wyszukać dany produkt. Ścieżka kategorii jest rozumiana jako struktura w systemie e-opisy.

8.5.3 WYNIK

Metoda zwraca listę dostępnych marek i atrybutów, które dostępne są jako elementy filtrujące.

parametr	typ		znaczenie
category_path	string		ścieżka kategorii
category_id	int		identyfikator kategorii
filters	array		pełna ścieżka do pliku
	parametr	typ	znaczenie
	brands	array	tablica dostępnych marek produktów
	parametr	typ	znaczenie
	id	int	identyfikator marki
	name	string	nazwa marki
	attributes	array	tablica dostępnych atrybutów
	Parametr	typ	Znaczenie
	id	int	identyfikator atrybutu
	name	string	nazwa atrybutu

8.5.4 PRZYKŁADY

Komputery i akcesoria/Laptopy i tablety/Notebooki, Netbooki, Ultrabooki"

wynik:

Lista marek i cech możliwych do przefiltrowania.

```
category_path => Komputery i akcesoria/Laptopy i tablety/Notebooki, Netbooki, Ultrabooki,  
category_id => 168,  
filters => [  
  brands => [  
    ...  
    31 => [  
      id => 480  
      name => HP  
    ],  
    ...  
  ],  
  ...  
]
```



```

55 => [
    id => 5614
    name => VORDON
]
...
],
attributes => [
    0 => [
        id => 51718,
        name => Rodzaj
    ],
    ...
    19 => [
        id => 7011,
        name => Przekątna ekranu (cal)
    ],
    ...
]
]

```

8.6 *api.getFilterParameterValues*

8.6.1 DEKLARACJA

`array = api.getFilterParameterValues(int, int)`

8.6.2 ARGUMENTY

Metoda wymaga dwóch argumentów i są nimi id podkategorii oraz id atrybutu.

8.6.3 WYNIK

Metoda zwraca listę atrybutów umożliwiających filtrowanie.

parametr	typ	znaczenie
category_id	int	identyfikator kategorii
attribute_id	int	identyfikator atrybutu
values	array	tablica atrybutów umożliwiających filtrowanie

8.6.4 PRZYKŁADY

```

category_id => 168,
attribute_id => 7011,
values => [
    0 => 13.3,
    ...
    3 => 15.6,
    ...
    32 => 15
    ...
]

```

8.7 *api.getFilteredProducts*

8.7.1 DEKLARACJA

array = api.getFilteredProducts(struct)

8.7.2 ARGUMENTY

Metoda wymaga argumentów zgodnie z poniższym przykładem:

```
category_path => sciezka kategorii,
parameters => [
  page => kolejna strona wyniku. Opcjonalne. Domyślnie 1,
  perPage => ilość wyników na stronie. Opcjonalne. Domyślnie 2000. Nie więcej
niż 2000.
],
filters => [
  brands => [
    id marki 1,
    id marki 2,
    ....
  ],
  attributes => [
    [
      attribute_id => id atrybutu 1,
      values => [
        pierwsza wartość atrybutu 1,
        druga wartosc atrybutu 1,
        ...
      ]
    ],
    [
      attribute_id => id atrybutu 2,
      values => [
        pierwsza wartość atrybutu 2,
        druga wartosc atrybutu 2,
        ...
      ]
    ]
  ]
]
```

8.7.3 WYNIK

Metoda zwraca listę produktów spełniających zadane kryteria.



parametr	typ	znaczenie
category_path	string	ścieżka kategorii
total	Int	ilość produktów spełniających zadane kryteria
page	Int	numer zwróconej strony
perPage	int	ilość wyników na stronę
products	array	lista produktów spełniających dane kryteria

parametr	typ	znaczenie
product_id	int	identyfikator produktu
	string	
name	string	nazwa produktu
trade_name	String	nazwa handlowa
type	String	typ produktu
brand	String	nazwa marki
model	String	model produktu
part_number	String	part numer produktu
short_description	String	krótki opis produktu
photo	string	link do miniatury produktu

8.7.4 PRZYKŁADY

```
category_path => Komputery i akcesoria/Laptopy i tablety/Notebooki, Netbooki, Ultrabooki,
total => 1639,
page => 1,
perPage => 10,
products => [
  0 => [
    product_id => 17528,
    name => nx9105,
    type => Notebook,
    brand => HP,
    model =>
    part_number => PG690EA,
    short_description =>
    photo => http://api.e-
opisy.pl/img/getProductMiniature.php?product_id=17528
  ],
  ....
  9 => [
    product_id => 20065,
    name => nx6125,
    type => Notebook,
    brand => HP,
    model =>
    part_number => PY416EA,
    short_description =>
    photo => http://api.e-
opisy.pl/img/getProductMiniature.php?product_id=20065
  ]
]
```



8.8 *api.createMissing*

8.8.1 DEKLARACJA

```
array = api.createMissing(struct)
```

8.8.2 ARGUMENTY

Metoda przyjmuje następujące argumenty:

parametr	typ	znaczenie
name	string	nazwa produktu
ean	string	kod EAN produktu
pn	string	part number produktu
producer	string	nazwa producenta
note	string	notatka

8.8.3 WYNIK

Wynikiem wywołania metody jest zwrócenie ID braku. W przypadku gdy zostanie zwrócony inny tekst, wywołanie metody nie powiodło się.

8.9 *api.addProductToMy*

8.9.1 DEKLARACJA

```
array = api.addProductToMy(int)
```

8.9.2 ARGUMENTY

Metoda wymaga jednego argumentu, a jest nim dedykowany identyfikator produktu z Bazy Danych E-opisy.

8.9.3 WYNIK

Wynikiem wywołania metody jest zwrócenie komunikatu: „Product has been added”. W przypadku, gdy zostanie zwrócony inny tekst, wywołanie metody nie powiodło się.



8.10 *api.myResolvedDefects*

8.10.1 DEKLARACJA

array = api.myResolvedDefects(ISO 8601 DateTime)

8.10.2 ARGUMENTY

Metoda wymaga wpisania daty i godziny w następującym formacie: rrrr-mm-dd hh:mm:ss

8.10.3 WYNIK

Wynikiem wywołania metody jest zwrócenie listy maksymalnie 2000 rozwiązanych zgłoszeń typu BrP (brak produktu).

parametr	typ	znaczenie
defect_id	int	id braku
product_id	int	id produktu
	string	
reporte_date	ISO 8601 DateTime	data zgłoszenia
reporte_note	string	notatka zgłoszenia
resolve_date	ISO 8601 DateTime	data rozwiązania zgłoszenia
resolve_note	string	notatka rozwiązania zgłoszenia
category_id	int	id kategorii
EAN	int	kod EAN
name	struct	nazwa

8.11 *api.getUpdated*

8.11.1 DEKLARACJA

array = api.getUpdated(ISO 8601 DateTime)

8.11.2 ARGUMENTY

Metoda wymaga wpisania daty w następującym formacie: rrrr-mm-dd

8.11.3 WYNIK

Wynikiem wywołania metody jest zwrócenie listy zaktualizowanych po wskazanej dacie produktów. Wyświetlone zostaną tylko te produkty, w które uległy modyfikacji maksymalnie 30 dni po wskazanej dacie.



8.12 api.getUpdatedInMyProducts

8.12.1 DEKLARACJA

array = api.getUpdatedInMyProducts(ISO 8601 DateTime)

8.12.2 ARGUMENTY

Metoda wymaga wpisania daty w następującym formacie: rrrr-mm-dd

8.12.3 WYNIK

Wynikiem wywołania metody jest zwrócenie listy produktów znajdujących się w moich produktach zaktualizowanych po wskazanej dacie. Wyświetlone zostaną tylko te produkty, które były edytowane lub dodane do moich produktów maksymalnie 30 dni po wskazanej dacie.

9. KODY BŁĘDÓW

fault code	fault string	znaczenie
1	wrong list of arguments	nieprawidłowe argumenty
1	unknown method	nieznana metoda API
2	bad argument type	nieprawidłowy typ argumentów
3	db query failed	błąd zapytania SQL
4	db connection failed	błąd połączenia z bazą SQL
5	no such item	nie ma takiego obiektu
6	empty result set	rezultat funkcji to pusta tablica
7	ambiguous EAN	niejednoznaczny kod EAN
7	limit has been exceeded products	limit przypisanych produktów został przekroczony
9	already have this product	produkt jest już dodany
10	incorect request format	nieprawidłowy format JSON
19	invalid date format	błąd w formacie daty
20	too large time interval (max 30 days from the current date)	za duża różnica w czasie (max 30 dni od bieżącej daty)
1024	db connection failed	błąd połączenia z bazą danych
1025	db query failed	błąd zapytania SQL
1280	access denied1	próba podłączenia do API z nieautoryzowanego adresu IP
1281	apikey not given	próba podłączenia do API bez podania klucza
1282	unknow apikey	próba podłączenia do API z użyciem nieznanego klucza
1283	service unavailable	dostęp do API został zablokowany
7011	fault string	produkt nie ma dołączonych plików



10. KONTAKT

E-OPISY.PL

INFO TECHNOLOGIES Sp. z o.o.

T: +48 22 538 40 40 | E-MAIL: biuro@e-opisy.pl

hd@e-opisy.pl

Pomoc Techniczna – zgłoszenia i zapytania związane z poprawnym działaniem i udostępnianiem naszych usług.

kontent@e-opisy.pl

Dział Opisów Produktów - zgłoszenia i zapytania związane z poprawnością opisów produktów (kontentem), błędy, braki, itp.

bok@e-opisy.pl

Biuro Obsługi Klienta - wszystkie pozostałe zapytania i sugestie.

Adres rejestrowy: Plac Wolności 13/2, 35-073 Rzeszów

Biuro Lublin: ul. Tomasza Zana 11A, 20-601 Lublin

Biuro Warszawa: ul. Wołodyjowskiego 36B, 02-724 Warszawa

Biuro Poznań: ul. Petera Mansfelda 4, 60-855 Poznań

NIP 8952012144 | REGON 021857164 | KRS 0000416743

Sąd Rejonowy dla M. St. Warszawy w Warszawie, XIII Wydział Gospodarczy Krajowego Rejestru Sądowego.

Kapitał zakładowy: 30 000,00 zł.

