

otoDom WebAPI

dokumentacja usługi sieciowej serwisu otodom.pl
dla biur nieruchomości i deweloperów

wersja 1.16



Wstęp

Serwis otoDom.pl udostępnia usługę sieciową (web service) opartą na protokole SOAP, służącą do komunikacji z serwisem i realizacji usług takich jak zamieszczanie czy edycja ogłoszeń. Usługa zdefiniowana jest za pomocą języka WSDL i może być łatwo implementowana niezależnie od wykorzystywanej platformy po stronie klienta.

W dalszej części dokumentu usługa nazwana jest **otoDom WebAPI**.

Dostęp do otoDom WebAPI w wersji testowej
oraz opis wykorzystywanych metod i typów danych:

<http://beta.otodom.pl/webapi/web.php>

specyfikacja WSDL:

<http://beta.otodom.pl/webapi/web.php?wsdl>

Dostęp do otoDom WebAPI w wersji produkcyjnej
oraz opis wykorzystywanych metod i typów danych:

<http://webapi.otodom.pl/web.php>

specyfikacja WSDL:

<http://webapi.otodom.pl/web.php?wsdl>

Historia zmian

1.14

- nowe metody getAllInsertions, getUserInfo
- nowe pola w obiekcie OfferWsdL (InsertionStatus, InsertionModificationDate, InsertionExpirationDate)

1.15

- nowa metoda getInsertionByRemoteId,

1.16

- wyłączenie metod getBuildingFloorTypes (dla \$insertionKind == 'Flat'), getHeightTypes oraz getRoomNumTypes
- pola obiektu OfferWsdL: BulidingFloorsNum, roomsNum, przyjmują wartości liczbowe zamiast słownikowych
- dla Hall pole height nie jest wymagane
- metody do obsługi danych adresowych oddziału i danych kontaktowych: doAdditionalAddressEdit, doAdditionalUserContactEdit, getAdditionalAddresses, getAdditionalUserContacts
- nowe opcjonalne pola w obiekcie OfferWsdL: AdditionalUserContact, AdditionalAddress

Dostęp do otoDom WebAPI

Aby uzyskać dostęp do interfejsu otoDom WebAPI niezbędne jest posiadanie własnego klucza dostępowego oraz aktywne konto biura nieruchomości lub dewelopera w serwisie otoDom. W celu otrzymania klucza należy skontaktować się z Działem Technicznym otoDom telefonicznie lub wysyłając w tej sprawie email na adres: tech@otodom.pl.

Dodawanie/edycja ogłoszeń

W celu dodawania/edycji ogłoszenia należy w pierwszej kolejności zalogować się wywołując metodę `doLogin()`. Wywołanie metody zwróci identyfikator sesji – `sessionKey`. Identyfikator sesji będzie wykorzystany w wywołaniach innych metod np. `doInsertion()` pozwalającej dodawać/edytować ogłoszenia.

Kolejnym krokiem jest przygotowanie obiektu reprezentującego pojedyncze ogłoszenie (typ `OfferWsd1` w otoDom WebAPI). Po przygotowaniu obiektu można już wywołać metodę `doInsertion()` przekazując do niej następujące parametry: identyfikator sesji oraz obiekt ogłoszenia. Każde wywołanie `doInsertion()` zwraca identyfikator ogłoszenia w serwisie otoDom lub też informacje o ewentualnych błędach.

Metoda `doInsertion()` pozwala zarówno dodawać oraz edytować ogłoszenie. Jeśli w obiekcie ogłoszenia nie będzie podany atrybut `InsertionId` wtedy zostanie dodane nowe ogłoszenie natomiast gdy atrybut ten zostanie podany wtedy odbędzie się edycja ogłoszenia i będzie możliwa korekta danych ogłoszenia.

Aby uprościć i przyspieszyć dodawanie/edycję ogłoszeń metodą `doInsertion()` można wywoływać wielokrotnie w ramach jednej sesji podstawiając do kolejnych wywołań kolejne ogłoszenia.

W celu zmniejszenia ilości danych, w czasie edycji ogłoszeń nie jest konieczne przekazywanie wszystkich pól. Wyłącznie pola, które zostaną zdefiniowane, będą uaktualnione w ogłoszeniu. Aby wyczyścić pole w czasie edycji konieczne jest wysłanie pustego ciągu znaków w przypadku pola typu `string`, 0 w przypadku pola typu `int`, lub pustej tablicy w przypadku pola typu `ArrayOf`. Wyjątek stanowią pola opisujące lokalizację. Jeżeli chcemy zmienić którekolwiek z nich konieczne jest wysłanie pozostałych.

Deaktywacja ogłoszeń

Deaktywacja ogłoszenia jest realizowana przez metodę `doInsertionInactivate()`. Jako jeden z parametrów metoda ta przyjmuje identyfikator ogłoszenia.

Aktywacja/przedłużanie ogłoszeń

Aktywacja ogłoszenia nieaktywnego lub przedłużanie okresu aktywności ogłoszenia aktywnego realizowane jest przez metodę `doInsertionActivate(string)`

`sessionKey,int insertionId[,int periodCount])`. Jako parametry podajemy klucz sesji, identyfikator ogłoszenia oraz opcjonalnie liczbę okresów, na które ma zostać przedłużone ogłoszenie.

Przykłady

W przykładach wykorzystano testowe konto agencji z aktywnym kluczem WebAPI:

webapiKey: ABSDFB34FI651N4MJAXY564F

login: test@otodom.pl

hasło: testotodom

Przykłady napisane są w języku PHP w wersji 5 i wykorzystują wbudowaną klasę SoapClient.

Logowanie

```
<?php
```

```
try {
```

```
    // klucz dostępu do WebAPI
```

```
    $accessKey = "ABSDFB34FI651N4MJAXY564F";
```

```
    // utworzenie obiektu klienta Soap
```

```
    $soapClient = new SoapClient(  
        "http://beta.otodom.pl/webapi/web.php?wsdl");
```

```
    // zalogowanie się do WebAPI. Metoda doLogin zwraca klucz sesji.
```

```
    $sessionKey = $soapClient->doLogin(  
        $accessKey,"test@otodom.pl","testotodom" );
```

```
    echo "Logowanie powiodło się. Klucz sesji to $sessionKey.n\";
```

```
} catch (SoapFault $fault) {
```

```
    echo "Logowanie nie powiodło się. Kod błędu: [{ $fault->faultcode}]  
        { $fault->faultstring}\n\";
```

```
}
```

```
...
```

```
?>
```

Dodawanie/edycja ogłoszenia

```
<?php
```

```
//przykładowy sessionId: 113f4d58048d5dd567f7947443efb9ae
```

```
...
```

```
// obiekty wykorzystywane do utworzenia i edycji ogłoszeń
```

```
class OfferWsdL {
```

```
    public $InsertionId, $InsertionStatus, $InsertionExpirationDate ,  
        $InsertionModificationDate $Country, $Province, $District, $CityId,  
        $CityName, $QuarterId, $QuarterName, $Street, $Category, $Investment,  
        $RemoteId, $Price, $PriceType, $PriceCurrency, $Area, $Keywords,  
        $MarketType,$BuildingOwnership, $OfferType, $Description,  
        $AllegroCategory, $Pictures, $Height, $AccessTypes, $Structure,  
        $MediaTypes, $Heating, $UseTypes, $ParkingType, $Flooring, $Lighting,  
        $Localization, $Type, $Dimensions, $Fence, $VicinityTypes,  
        $ExtrasTypes, $FloorNo, $BuildingType, $BulidingFloorsNum, $RoomsNum,  
        $TerrainArea, $ConstructionStatus, $BuildYear, $RoofType, $GarretType,  
        $WindowsType, $HeatingTypes, $FenceTypes, $PersonsTypes, $Free,  
        $Furnished, $AdditionalUserContact, $AdditionalAddress;
```

```
}
```

```
class PictureWsdL {
```

```
    public $Data;
```

```
}
```

```
try {
```

```
    //utworzenie obiektu ogłoszenia
```

```
    $offer = new OfferWsdL ();
```

```
    // W celu dokonania edycji należy podać prawidłowy identyfikator ogłoszenia.
```

```
    // W trybie edycji nie jest konieczne przekazywanie wszystkich zmiennych.
```

```
    // Tylko zmienne zdefiniowane zostaną zmodyfikowane w ogłoszeniu
```

```
    //$offer->InsertionId = 1047;
```

```
    // zmienne określające lokalizację. W czasie edycji aby zmodyfikować jedną
```

```
    // z poniższych zmiennych konieczne jest zdefiniowanie ich wszystkich, ponieważ
```

```

// są od siebie zależne. Zmienna QuarterId/QuarterName jest opcjonalna.
$offer->Country = 1;
$offer->Province = 15;
$offer->District = 21;
$offer->CityId = 1;
// Zamiast identyfikatorów miast można używać ich nazw
// $offer->CityName = "Poznań";
$offer->QuarterName = "Piątkowo";

// typ oferty
$offer->OfferType = "rent_House";

// cena
$offer->PriceCurrency = 1;
$offer->PriceType = "brutto";
$offer->Price = 500000;

// powierzchnia
$offer->Area = 100;

// identyfikator kategorii w Allegro
$offer->AllegroCategory = 27491;

// pozostałe, opcjonalne dane opisujące ofertę
$offer->MarketType = "secondary";
$offer->MediaTypes = array("prad","woda","gaz","oczyszczalnia");
$offer->BuildingFloorsNum = "2 floors";

// dodawanie zdjęć do ogłoszenia
$picture = new PictureWSDL();
$picture->Data = base64_encode(file_get_contents("otodom.jpg"));

$offer->Pictures = array($picture);

// publikacja ogłoszenia
$insertionId = $soapClient->doInsertion($sessionKey,$offer);
echo "Dodanie ogłoszenia powiodło się. Id ogłoszenia to $insertionId.\n";

} catch (SoapFault $fault) {

echo "Dodanie ogłoszenia nie powiodło się. Kod błędu: [{"$fault->faultcode}]
{"$fault->faultstring}.\n";

```

```

}
...
?>

```

Macierz powiązań pól obiektu OfferWsdł z typami ogłoszeń i metodami udostępniającymi dopuszczalne wartości.

* nazwa pola

metoda \ typ obiektu	Flat	House	Terrain	Room	CommercialProperty	Hall	Garage
getAccessTypes			AccessTypes			AccessTypes	
getMediaTypes		MediaTypes	MediaTypes			MediaTypes	
getUseTypes					UseTypes	UseTypes	
getParkingTypes						ParkingType	
getFlooringTypes						Flooring	
getLocalizationTypes		Localization					Localization
getStructureTypes						Structure	Structure
getVicinityTypes			VicinityTypes				
getExtrasTypes	ExtrasTypes				ExtrasTypes		
getFloorTypes	FloorNo				FloorNo		
getBuildingFloorTypes		BulidingFloorsNum					
getRoomsNumTypes							
getBuildingTypes	BuildingType	BuildingType		BuildingType			
getBuildingKindTypes		Type	Type				
getRoofTypes		RoofType					
getGarretTypes		GarretType					
getWindow sTypes		Window sType					
getHeatingTypes		HeatingTypes					
getFenceTypes		FenceTypes					
getRoomFreeTypes				Free			
getHeightTypes							
getConstructionStatuses		ConstructionStatus					
getBuildingOw nerships	BuildingOw nership						
getPersonsTypes				PersonsTypes			

Lista pól wymaganych

Flat	House	Terrain	Room	CommercialProperty	Hall	Garage
Price	Price	Price	Price	Price	Price	Price
PriceType	PriceType	PriceType	PriceType	PriceType	PriceType	PriceType
PriceCurrency	PriceCurrency	PriceCurrency	PriceCurrency	PriceCurrency	PriceCurrency	PriceCurrency
Area	Area	Area	MarketType	Area	Area	MarketType
MarketType	MarketType	MarketType		MarketType	MarketType	Localization
		Fence			Heating	Lighting
		Type				Heating
						Structure

Poza polami wymienionymi powyżej konieczne jest ustalenie lokalizacji. W przypadku Polski oznacza to obowiązkowe wypełnienie pól Country, Province, Distinct, CityId/CityName. W

przypadku zagranicy Country oraz CityId/CityName.

Ograniczenia dotyczące wybranych pól

- CityName – maksymalna długość 32 znaki
- QuarterName – maksymalna długość 64 znaki
- Keywords – maksymalnie 3 słowa rozdzielone średnikiem, każde słowo o maksymalnej długości 16 znaków
- Description – nie ma ograniczenia długości, z treści wycinane są wszystkie tagi HTML
- Pictures – liczba zdjęć jest ograniczona do 9ciu, dozwolone formaty plików to JPEG i GIF, maksymalny rozmiar pojedynczego pliku to 300 KB

Lista pól obiektu OfferWsdI

- InsertionId (int) - identyfikator istniejącego ogłoszenia. Identyfikator należy podać w przypadku edycji ogłoszenia
- InsertionStatus (string) – status ogłoszenia
- InsertionExpirationDate (string) – data wygaśnięcia ogłoszenia
- InsertionModificationDate (string) – data modyfikacji ogłoszenia
- Country (int) - Kraj
- Province (int) - Województwo
- District (int) - Powiat
- CityId (int) - Miejscowość
- CityName (string) - Miejscowość (w razie braku id)
- QuarterId (int) - Dzielnica
- QuarterName (string) - Dzielnica (w razie braku id)
- Street (string) - Ulica
- Category (int) - Kategoria
- Investment (string) - Inwestycja
- RemoteId (string) - Numer oferty w biurze - lokalne ID wpisu, unikalne dla jednego konta użytkownika. Numer ten może posłużyć również do weryfikacji unikalności przekazywanych za pośrednictwem WebApi ofert.
- Price (float) - Cena
- PriceType (string) - typ ceny netto/brutto
- PriceCurrency (int) - Waluta
- Area (int) - Powierzchnia w metrach kwadratowych
- Keywords (string) - Słowa kluczowe rozdzielone średnikiem
- MarketType (string) - Rynek wtorny, pierwotny...
- BuildingOwnership (string) - Forma własności
- OfferType (string) - Rodzaj oferty podany przez metode getInsertionKinds zwraca rodzajOferty_podmiotOferty
- Description (string) - Opis oferty

- AllegroCategory (int) - Kategoria Allegro.pl
- Pictures (ArrayOfPictureWsd) - Zdjęcia oferty
- Height (int) - Wysokość stropu
- AccessTypes (ArrayOfstring) - Dojazd
- Structure (string) - Konstrukcja
- MediaTypes (ArrayOfstring) - Media
- Heating (string) - Ogrzewanie y/n
- UseTypes (ArrayOfstring) - Przeznaczenie przedmiotu oferty
- ParkingType (string) - Parking
- Flooring (string) - Posadzka
- Lighting (string) - Oświetlenie y/n
- Localization (string) - Lokalizacja
- Type (string) - typ obiektu (dla działki, domu)
- Dimensions (string) - Wymiary
- Fence (string) - Ogrodzenie y/n
- VicinityTypes (ArrayOfstring) - Okolica
- ExtrasTypes (ArrayOfstring) - Inform. dodatkowe
- FloorNo (string) - Piętro usytuowania oferty
- BuildingType (string) - Rodzaj zabudowy
- BuildingFloorsNum (string/int) - Liczba pięter budynku
- RoomsNum (int) - Liczba pokoi
- TerrainArea (int) - Powierzchnia działki
- ConstructionStatus (string) - Stan wykończenia
- BuildYear (int) - Rok budowy
- RoofType (string) - Typ dachu
- GarretType (string) - Typ poddasza
- WindowsType (string) - Typ okien
- HeatingTypes (ArrayOfstring) - Typ ogrzewania
- FenceTypes (ArrayOfstring) - Typ ogrodzenie
- PersonsTypes (ArrayOfstring) - Ilość osób na który przystosowany jest pokój
- Free (string) - Wolny od
- Furnished (string) - Umeblowany
- AdditionalUserContact (int) - Opcjonalna osoba kontaktowa
- AdditionalAddress (int) - Opcjonalny adres oddziału

Dopuszczalne wartości pól można pobrać przy użyciu metod wymienionych w znajdującej się wcześniej macierzy powiązań. W zależności od typu obiektu dopuszczalne wartości dla danego pola mogą się różnić.

Metody operujące na ogłoszeniach użytkownika

- doInsertion – metoda służy do dodawania i edycji ogłoszeń
- doInsertionActivate – metoda służy do aktywacji ogłoszeń o podanym identyfikatorze
- doInsertionInactivate – metoda pozwala deaktywować ogłoszenie o wskazanym

identyfikatorze

- `getAllInsertions` – zwraca listę wszystkich ogłoszeń dla zalogowanego użytkownika
- `getActiveInsertions` – zwraca listę aktywnych ogłoszeń dla zalogowanego użytkownika
- `getInactiveInsertions` – zwraca listę nieaktywnych ogłoszeń dla zalogowanego użytkownika
- `getInvestments` – zwraca listę inwestycji dla zalogowanego dewelopera
- `getInsertion` – zwraca ogłoszenie na podstawie identyfikatora `otoDom`
- `getInsertionByRemoteId` – zwraca ogłoszenie na podstawie identyfikatora oferty w biurze

Metody zwracające listy ogłoszeń / inwestycji mogą opcjonalnie zwrócić również zdjęcia powiązane z tymi ogłoszeniami / inwestycjami za co odpowiada parametr `withPictures`.