1. Get authenticated user

Uzyskanie zautoryzowanego użytkownika

Wywołanie: GET /users/me

Dane w GET: -

Parametry ścieżki: -

W odpowiedzi serwer zwraca wszystkie dane użytkownika: id, email, status konta, imie, nazwisko, nr telefonu, nr telefonu do logowania, data utworzenia konta, data ostatniego loginu, avatar, czy konto jest biznesem

Zwracane dane:

{

"id": 123,

"email": "john@mail.com",

"status": "confirmed",

"name": "John",

"phone": 123123123,

"phone\_login": 123123123,

"created\_at": "2018-01-29 14:04:13",

"last\_login\_at": "2018-01-30 08:20:28",

"avatar": null,

"is\_business": true

}

1. Get user

Uzyskanie jakiegokolwiek uzytkownika

Wywołanie: GET /users/{id}

Dane w GET: id

Parametry śćieżki: id – wymagane – id użytkownika

Serwer zwraca dane użytkownika takie jak: id użytkownika, imie użytkownika, avatar

Zwracane dane:

{

"id": 1,

"name": "John",

"avatar": null

}

1. Get account balance

Uzyskanie balansu konta

Wywołanie: GET/users/me/account-balance

Dane w GET: -

Parametry ścieżki: -

W odpowiedzi serwer zwraca sumę pieniędzy, balans portfela, balans bonusu, balans zwrotów, waluta

Zwracane dane:

{

"sum": 6988.02,

"wallet": 6988.02,

"bonus": 0,

"refund": 0,

"currency": "PLN"

}

1. Get available payment methods

Uzyskanie możliwych metod platnosci

Wywołanie: GET /users/me/payment-methods

Dane w GET: -

Parametry scieżki: -

Serwer zwraca mozliwe platnosci

Zwracane dane:

[

"account",

"postpaid"

]

1. List of country regions

Uzyskanie listy regionów

Wywołanie: GET /regions

Dane w GET: -

Parametry scieżki: -

Serwer zwraca id regionu i nazwę

Zwracane dane:

[

{

"id": 1,

"name": "Wielkopolskie"

}

]

1. Get region

Uzyskanie regionu po id

Wywołanie: GET /regions/{regionId}

Dane w GET: regionId

Parametry scieżki: regionId – wymagane – id regionu

Serwer zwraca id regionu i nazwę

Zwracane dane:

{

"id": 1,

"name": "Wielkopolskie"

}

1. Get cities

Uzyskanie listy miast

Wywołanie: GET /cities

Dane w GET: regionId

Parametry zapytania: limit – ogranicza liczbę wyników na stronę (domyślnie 1000), offset – element startowy

Serwer zwraca listę miast zawierających id miasta, id regionu, nazwe miasta, nazwe powiatu, nazwe miasta, współrzędne geograficzne

Zwracane dane:

[

{

"id": 627,

"region\_id": 123,

"name": "Aleksandrów Kujawski",

"county": "aleksandrowski",

"municipality": "Aleksandrów Kujawski",

"latitude": 52.86653,

"longitude": 18.69801

}

]

1. Get city

Uzyskanie miasta po id

Wywołanie: GET /cities/{cityId}

Dane w GET: cityId

Parametry scieżki: cityId – wymagane – id miasta

Serwer zwraca dane miasta: id, id regionu, nazwe miasta, powiat, dane geograficzne

Zwracane dane:

{

"id": 1,

"name": "Wielkopolskie"

}

1. Get city districts

Uzyskanie dzielnic miasta

Wywołanie: GET /cities/{cityId}/districts

Dane w GET: cityId

Parametry scieżki: cityId – wymagane – id miasta

Serwer zwraca dane dzielnic miasta: id dzielnicy, id miasta, nazwa dzielnicy, dane geograficzne

Zwracane dane:

[

{

"id": 97,

"city\_id": 5659,

"name": "Aniołki",

"latitude": 54.36306,

"longitude": 18.63193

}

]

1. Get district

Uzyskanie informacji o dzielnicy

Wywołanie: GET /districts/{districtId}

Dane w GET: districtId

Parametry scieżki: districtId – wymagane – id dzielnicy

Serwer zwraca dane dzielnicy: id dzielnicy, id miasta, nazwa dzielnicy, dane geograficzne

Zwracane dane:

[

{

"id": 97,

"city\_id": 5659,

"name": "Aniołki",

"latitude": 54.36306,

"longitude": 18.63193

}

]

1. Get locations

Uzyskanie informacji o lokalizacji

Wywołanie: GET /locations

Dane w GET: -

Parametry zapytania bazy danych: latitude – szerokość geog. lokalizacji - wymagane, longitude – długość geog. - wymagane

Serwer zwraca dane miasta i dzielnicy: id dzielnicy, id miasta, nazwa dzielnicy, dane geograficzne, id, id regionu, nazwe miasta, powiat, dane geograficzne

Zwracane dane:

[

{

"city": {

"id": 627,

"region\_id": 123,

"name": "Aleksandrów Kujawski",

"county": "aleksandrowski",

"municipality": "Aleksandrów Kujawski",

"latitude": 52.86653,

"longitude": 18.69801

},

"district": {

"id": 97,

"city\_id": 5659,

"name": "Aniołki",

"latitude": 54.36306,

"longitude": 18.63193

}

}

]

1. Get languages

Uzyskanie listy języków dostępnych na stronie

Wywołanie: GET /languages

Dane w GET: -

Parametry scieżki: -

Serwer zwraca dane języków: kod języka, czy jest domyślny

Zwracane dane:

[

{

"code": "pl",

"is\_default": true

}

]

1. Get categories

Uzyskanie listy kategorii

Wywołanie: GET /categories

Dane w GET: -

Parametry zapytania: parent\_id – id nad kategorii

Serwer zwraca dane kategorii: id kategorii, nazwa kategorii, id nad kategorii, limit zdjęć, czy jest „liściem”

Zwracane dane:

[

{

"id": 1291,

"name": "Akcesoria kuchenne\\",

"parent\_id": 1409,

"photos\_limit": 0,

"is\_leaf": true

}

]

1. Get category

Uzyskanie informacji o kategorii

Wywołanie: GET /categories/{categoryId}

Dane w GET: categoryId

Parametry ścieżki: parent\_id – id nad kategorii

Serwer zwraca dane kategorii: id kategorii, nazwa kategorii, id nad kategorii, limit zdjęć, czy jest „liściem”

Zwracane dane:

{

"id": 1291,

"name": "Akcesoria kuchenne\\",

"parent\_id": 1409,

"photos\_limit": 0,

"is\_leaf": true

}

1. Get category atributes

Uzyskanie definicji atrybutów danej kategorii

Wywołanie: GET /categories/{categoryId}/attributes

Dane w GET: categoryId

Parametry ścieżki: parent\_id – id nad kategorii

Serwer zwraca liste atrybutów danej kategorii i ich definicje

Zwracane dane:

[

{

"code": "price",

"label": "Cena",

"unit": null,

"validation": {

"type": "price",

"required": true,

"numeric": true,

"min": null,

"max": null,

"allow\_multiple\_values": false

},

"values": []

}

]

1. Get thread

Uzyskanie informacji o danym wątku

Wywołanie: GET /threads/{threadId}

Dane w GET: threadId

Parametry ścieżki: threadId – id wątku - wymagane

Serwer zwraca dane danego wątku: id, id postu, id rozmówcy, liczba wiadomości, liczba nieodczytanych, data utworzenia, czy ulubiony

Zwracane dane:

{

"id": 1,

"advert\_id": 1,

"interlocutor\_id": 1,

"total\_count": 1,

"unread\_count": 0,

"created\_at": "2018-01-30 08:26:36",

"is\_favourite": false

}

1. Get messages

Uzyskanie listy wiadomości wątku

Wywołanie: GET /threads/{threadId}/messages

Dane w GET: threadId

Parametry ścieżki: threadId – id wątku - wymagane

Parametry zapytania: offset – element początkowy, limit – limit wiadomości na stronę

Serwer zwraca dane wiadomości: id, id wątku, data utworzenia, typ, treść, załączniki, czy przeczytany

Zwracane dane:

[

{

"id": 1,

"thread\_id": 1,

"created\_at": "2018-01-30 08:26:36",

"type": "sent",

"text": "This is my message",

"attachments": null,

"is\_read": true

}

]

1. Post Message

Zapostowanie nowej wiadomości w danym wątku

Wywołanie: POST /threads/{threadId}/messages

Dane w POST: threadId

Parametry ścieżki: threadId – id wątku - wymagane

Schemat żądania – text – treść wiadomośći – wymagane, attachments - załączniki

Serwer otrzymuje dane wiadomości: text, attachments

Postowane dane:

{

"text": "string",

"attachments": [

{

"url": "string"

}

]

}

1. Get message

Uzyskanie informacji o wiadomości

Wywołanie: GET /threads/{threadId}/messages/{messageId}

Dane w GET: threadId, messageId

Parametry ścieżki: threadId – id wątku – wymagane, messageId – id wiadomości - wymagane

Serwer zwraca dane wiadomości: id, id wątku, data utworzenia, typ, treść, załączniki, czy przeczytany

Zwracane dane:

{

"id": 1,

"thread\_id": 1,

"created\_at": "2018-01-30 08:26:36",

"type": "sent",

"text": "This is my message",

"attachments": null,

"is\_read": true

}

1. Get active paid features

Uzyskanie informacji o aktywnych opłaconych funkcjach

Wywołanie: GET /adverts/{advertId}/paid-features

Dane w GET: advertId

Parametry ścieżki: advertId – id ogłoszenia – wymagane

Serwer zwraca dane o aktywnych opłaconych funkcjach – kod, typ, czas trwania, nazwa, ważne do

Zwracane dane:

[

{

"code": "promoted\_ad\_7",

"type": "topads",

"duration": 7,

"name": "Wyróżnienie ogłoszenia na 7 dni",

"valid\_to": "2018-02-05 12:12:12"

}

]