

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
}
```

2. Get user

Uzyskanie jakiegokolwiek użytkownika

Wywołanie: GET /users/{id}

Dane w GET: id

Parametry ścieżki: id – wymagane – id użytkownika

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

Zwracane dane:

```
{
  "id": 1,
  "name": "John",
  "avatar": null
}
```

3. 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"
}
```

4. Get available payment methods

Uzyskanie możliwych metod płatności

Wywołanie: GET /users/me/payment-methods

Dane w GET: -

Parametry ścieżki: -

Serwer zwraca możliwe płatności

Zwracane dane:

```
[
  "account",
  "postpaid"
]
```

5. List of country regions

Uzyskanie listy regionów

Wywołanie: GET /regions

Dane w GET: -

Parametry ścieżki: -

Serwer zwraca id regionu i nazwę

Zwracane dane:

```
[
  {
    "id": 1,
    "name": "Wielkopolskie"
  }
]
```

6. Get region

Uzyskanie regionu po id

Wywołanie: GET /regions/{regionId}

Dane w GET: regionId

Parametry ścieżki: regionId – wymagane – id regionu

Serwer zwraca id regionu i nazwę

Zwracane dane:

```
{
  "id": 1,
  "name": "Wielkopolskie"
}
```

7. 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, nazwę miasta, nazwę powiatu, nazwę miasta, współrzędne geograficzne

Zwracane dane:

```
[
  {
    "id": 627,
    "region_id": 123,
    "name": "Aleksandrów Kujawski",
    "county": "aleksandrowski",
    "municipality": "Aleksandrów Kujawski",
    "latitude": 52.86653,
    "longitude": 18.69801
  }
]
```

8. Get city

Uzyskanie miasta po id

Wywołanie: GET /cities/{cityId}

Dane w GET: cityId

Parametry ścieżki: cityId – wymagane – id miasta

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

Zwracane dane:

```
{
  "id": 1,
  "name": "Wielkopolskie"
}
```

9. Get city districts

Uzyskanie dzielnic miasta

Wywołanie: GET /cities/{cityId}/districts

Dane w GET: cityId

Parametry ścieżki: 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  
}
```

```
]
```

10. Get district

Uzyskanie informacji o dzielnicy

Wywołanie: GET /districts/{districtId}

Dane w GET: districtId

Parametry ścieżki: 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  
  }  
]
```

11. 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, nazwa 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  
    }  
  }  
]
```

12. Get languages

Uzyskanie listy języków dostępnych na stronie

Wywołanie: GET /languages

Dane w GET: -

Parametry ścieżki: -

Serwer zwraca dane języków: kod języka, czy jest domyślny

Zwracane dane:

```
[  
  {  
    "code": "pl",  
    "is_default": true  
  }  
]
```

13. Get categories

Uzyskanie listy kategorii

Wywołanie: GET /categories

Dane w GET: -

Parametry zapytania: parent_id – id nad kategorią

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

Zwracane dane:

```
[
  {
    "id": 1291,
    "name": "Akcesoria kuchenne\\",
    "parent_id": 1409,
    "photos_limit": 0,
    "is_leaf": true
  }
]
```

14. Get category

Uzyskanie informacji o kategorii

Wywołanie: GET /categories/{categoryId}

Dane w GET: categoryId

Parametry ścieżki: parent_id – id nad kategorią

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

Zwracane dane:

```
{
  "id": 1291,
  "name": "Akcesoria kuchenne\\",
  "parent_id": 1409,
  "photos_limit": 0,
  "is_leaf": true
}
```

15. Get category attributes

Uzyskanie definicji atrybutów danej kategorii

Wywołanie: GET /categories/{categoryId}/attributes

Dane w GET: categoryId

Parametry ścieżki: parent_id – id nad kategorią

Serwer zwraca listę 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": []
  }
]
```

16. 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  
}
```

17. 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  
  }  
]
```

18. 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 wiadomości – wymagane, attachments - załączniki

Serwer otrzymuje dane wiadomości: text, attachments

Postowane dane:

```
{  
  "text": "string",  
  "attachments": [  
    {  
      "url": "string"  
    }  
  ]  
}
```

19. 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  
}
```

20. 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"
  }
]
```