

Designmatch API

API aplikacji Designmatch

More information: <https://aleks-2.mat.umk.pl/pz2022/zesp09/>

Contact Info: 308297@stud.umk.pl

Version: 1.0.0

All rights reserved

<http://apache.org/licenses/LICENSE-2.0.html>

Access

1. HTTP Basic Authentication
- 2.

Methods

[Jump to [Models](#)]

Table of Contents

[Autoryzacja](#)

- [GET /auth/confirmEmail](#)
- [GET /auth/decodeToken](#)
- [POST /auth/token](#)
- [POST /auth/login](#)
- [POST /auth/registerArtist](#)
- [POST /auth/registerCompany](#)

[Chat](#)

- [GET /messages/{senderUsername}/{recipientUsername}/count](#)
- [GET /messages/{senderUsername}/{recipientUsername}](#)
- [GET /messages/conversations/{username}](#)
- [GET /messages/{id}](#)

[Filtr](#)

- [GET /public/api/filter/getAvailableCategories](#)
- [GET /public/api/filter/getAvailableCities](#)
- [GET /public/api/filter/getAvailableLanguages](#)
- [GET /public/api/filter/getAvailableLevels](#)
- [GET /public/api/filter/getAvailableTags](#)

[Profil Artysty](#)

- [DELETE /api/artist/deletePortfolioEntry/{username}/{imageId}](#)
- [POST /public/api/artist/filter](#)
- [GET /public/api/artist/getArtistProfileByUsername/{username}](#)
- [GET /public/api/artist/getPortfolioEntries/{username}](#)
- [GET /public/api/artist/getPortfolioImage/{username}/{portfolioEntryId}](#)
- [GET /public/api/artist/getProfileImageByUsername/{username}](#)
- [GET /public/api/artist/getShortArtistProfileByUsername/{username}](#)
- [PUT /api/artist/updateProfileByUsername/{username}](#)
- [POST /api/artist/updateProfileImage/{username}](#)
- [POST /api/artist/createPortfolioEntry/{username}](#)

[Profil Firmy](#)

- [GET /public/api/company/getProfileByUsername/{username}](#)
- [GET /public/api/company/getProfileImageByUsername/{username}](#)
- [PUT /api/company/updateProfileByUsername/{username}](#)
- [POST /api/company/uploadProfileImage/{username}](#)

[Użytkownik](#)

- [GET /api/user/getAllUsernames](#)
- [GET /api/user/getUserById](#)

[Zlecenia](#)

- [POST /api/commission/create](#)
- [DELETE /api/deleteCommission/{id}](#)
- [POST /public/api/commission/filter](#)
- [GET /public/api/commission/getById/{id}](#)
- [GET /public/api/commission/getAllCommissionFirmByUsername/{username}](#)
- [PUT /api/commission/setCompletedById/{id}](#)
- [PUT /api/commission/updateById/{id}](#)

Autoryzacja

GET /auth/confirmEmail

Potwierdzenie adresu email (confirm)

Query parameters

token (required)
Query Parameter —

Return type

[MyApiResponse](#)

Example data

Content-Type: application/json

```
{
  "success" : true,
  "message" : "Pomyślnie przetworzono zapytanie"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

Pomyślnie potwierdzono adres email [MyApiResponse](#)

400

Niepoprawny token [MyApiResponse](#)

GET /auth/decodeToken

[Up](#)

Zwraca dane aktualnie zalogowanego użytkownika (`decodeToken`)

Request headers

Return type

[MyApiResponse](#)

Example data

Content-Type: application/json

```
{
  "success" : true,
  "message" : "Pomyślnie przetworzono zapytanie"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

Pomyślnie zwrócono dane użytkownika [MyApiResponse](#)

401

Autoryzacja nie powiodła się [MyApiResponse](#)

POST /auth/token

[Up](#)

Zwraca nowy token JWT dla użytkownika (`getToken`)

Return type

[MyApiResponse](#)

Example data

Content-Type: application/json

```
{
  "success" : true,
  "message" : "Pomyślnie przetworzono zapytanie"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

Pomyślnie wygenerowano nowy token JWT [MyApiResponse](#)

401

Autoryzacja nie powiodła się [MyApiResponse](#)

POST /auth/login

[Up](#)

Logowanie użytkownika (**login**)

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

body [LoginRequest](#) (required)
Body Parameter —

Return type

[JwtAuthenticationResponse](#)

Example data

Content-Type: application/json

```
{
  "accessToken" : "accessToken"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

Zalogowano pomyślnie [JwtAuthenticationResponse](#)

401

Niepoprawne dane logowania [MyApiResponse](#)

POST /auth/registerArtist

[Up](#)

Rejestracja artysty (**registerArtist**)

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

body [RegisterRequest](#) (required)
Body Parameter —

Return type

[MyApiResponse](#)

Example data

Content-Type: application/json

```
{
  "success" : true,
  "message" : "Pomyślnie przetworzono zapytanie"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

Pomyślnie zarejestrowano użytkownika [MyApiResponse](#)

400

Niepoprawne dane rejestracji, być może są już zajęte [MyApiResponse](#)

POST /auth/registerCompany

[Up](#)

Rejestracja firmy (**registerCompany**)

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

body [RegisterRequest](#) (required)

Return type

[MyApiResponse](#)

Example data

Content-Type: application/json

```
{
  "success" : true,
  "message" : "Pomyślnie przetworzono zapytanie"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

Pomyślnie zarejestrowano użytkownika [MyApiResponse](#)

400

Niepoprawne dane rejestracji, być może są już zajęte [MyApiResponse](#)

Chat

GET /messages/{senderUsername}/{recipientUsername}/count [Up](#)

Zwraca liczbę nowych wiadomości w konwersacji (`countNewMessages`)

Path parameters

senderUsername (required)

Path Parameter — Nazwa użytkownika nadawcy

recipientUsername (required)

Path Parameter — Nazwa użytkownika odbiorcy

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

Liczba nowych wiadomości w konwersacji

404

Nie znaleziono konwersacji

GET /messages/{senderUsername}/{recipientUsername} [Up](#)

Zwraca wszystkie wiadomości w konwersacji (`findChatMessages`)

Path parameters

senderUsername (required)

Path Parameter — Nazwa użytkownika nadawcy

recipientUsername (required)

Path Parameter — Nazwa użytkownika odbiorcy

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

Lista wiadomości w konwersacji

404

Nie znaleziono konwersacji

GET /messages/conversations/{username} [Up](#)

Zwraca wszystkie konwersacje użytkownika (`findConversations`)

Path parameters

username (required)

Path Parameter — Nazwa użytkownika

Return typearray[[InterlocutorResponse](#)]**Example data**

Content-Type: application/json

```
[ {
  "company_name" : "company_name",
  "last_name" : "Kasinski",
  "last_message" : "Cześć, chciałbym zająć się stworzeniem strony internetowej",
  "first_name" : "Jakub",
  "username" : "jkasinski1"
}, {
  "company_name" : "company_name",
  "last_name" : "Kasinski",
  "last_message" : "Cześć, chciałbym zająć się stworzeniem strony internetowej",
  "first_name" : "Jakub",
  "username" : "jkasinski1"
} ]
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses**200**

Lista konwersacji użytkownika

404

Nie znaleziono użytkownika

GET /messages/{id}

[Up](#)

Zwraca wiadomość o podanym id ([findMessage](#))

Path parameters**id (required)***Path Parameter* — Id wiadomości format: int64**Produces**

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses**200**

Wiadomość o podanym id

404

Nie znaleziono wiadomości o podanym id

Filtry

GET /public/api/filter/getAvailableCategories

[Up](#)

Zwraca dostępne kategorie ([getAvailableCategories](#))

Return type

array[String]

Example data

Content-Type: application/json

```
[ "", "" ]
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses**200**

Pobrano dostępne kategorie

500Wystąpił błąd podczas pobierania dostępnych kategorii [AvailableSkillsCategoriesResponse](#)

GET /public/api/filter/getAvailableCities

[Up](#)

Zwraca dostępne miasta ([getAvailableCities](#))

Return type

array[String]

Example data

Content-Type: application/json

```
[ [{"Zdalnie\\",\\"Warszawa\\",\\"Kraków\\",\\"Wrocław\\",\\"Poznań\\",\\"Gdańsk\\",\\"Szczecin\\",\\"Bydgoszcz\\",\\"Lublin\\"}], [{"Zdalnie
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

Pobrano dostępne miasta

404

Wystąpił błąd podczas pobierania dostępnych miast

GET /public/api/filter/getAvailableLanguages

Up

Zwraca dostępne języki (**getAvailableLanguages**)

Return type

array[String]

Example data

Content-Type: application/json

```
[ "", "" ]
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

Pobrano dostępne języki

500

Wystąpił błąd podczas pobierania dostępnych języków

GET /public/api/filter/getAvailableLevels

Up

Zwraca dostępne poziomy (**getAvailableLevels**)

Return type

array[String]

Example data

Content-Type: application/json

```
[ "", "" ]
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

Pobrano dostępne poziomy

500

Wystąpił błąd podczas pobierania dostępnych poziomów

GET /public/api/filter/getAvailableTags

Up

Zwraca dostępne tagi (**getAvailableTags**)

Return type

array[String]

Example data

Content-Type: application/json

```
[ "", "" ]
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

Pobrano dostępne tagi

500

Wystąpił błąd podczas pobierania dostępnych tagów

ProfilArtysty

DELETE `/api/artist/deletePortfolioEntry/{username}/{imageId}`

Up

Usuwa wpis z portfolio artysty o podanej nazwie użytkownika i id wpisu w portfolio (`deletePortfolioEntry`)

Path parameters

username (required)

Path Parameter — Nazwa użytkownika artysty

imageId (required)

Path Parameter — Id zdjęcia w portfolio format: int64

Return type

[MyApiResponse](#)

Example data

Content-Type: application/json

```
{
  "success" : true,
  "message" : "Pomyślnie przetworzono zapytanie"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

Usunięto zdjęcie z portfolio artysty [MyApiResponse](#)

401

Autoryzacja nie powiodła się

403

Nie znaleziono artysty lub wpisu w portfolio

POST `/public/api/artist/filter`

Up

Zwraca pofiltrowaną listę krótkich profili artystów (`filterArtists`)

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

body [ArtistFilterRequest](#) (required)

Body Parameter — Kryteria filtrowania

Query parameters

page (optional)

Query Parameter — Numer strony format: int32

size (optional)

Query Parameter — Ilość elementów na stronie format: int32

Return type

[ShortArtistProfileResponse](#)

Example data

Content-Type: application/json

```
{
  "skills" : [ "skills", "skills" ],
  "firstname" : "Jakub",
  "city" : "Zdalnie",
  "level" : "Mid",
  "username" : "jkasinski1",
  "lastname" : "lastname"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

Znaleziono artystów [ShortArtistProfileResponse](#)

400

Niepoprawne parametry filtrowania

404

Nie udało się pobrać artystów

GET /public/api/artist/getArtistProfileByUsername/{username}

Zwraca profil artysty o podanej nazwie użytkownika ([getArtistProfileByUsername](#))

Path parameters

username (required)

Path Parameter — Nazwa użytkownika

Return type

[ArtistProfileResponse](#)

Example data

Content-Type: application/json

```
{
  "firstname" : "Jakub",
  "website" : "https://www.jkasinski.com",
  "education" : [ {
    "end_date" : "01/10/2021",
    "degree" : "Inżynier",
    "school_name" : "Politechnika Warszawska",
    "description" : "Studia inżynierskie na kierunku Informatyka",
    "field_of_study" : "Informatyka",
    "faculty" : "Wydział Informatyki, Elektroniki i Telekomunikacji",
    "start_date" : "01/10/2017"
  }, {
    "end_date" : "01/10/2021",
    "degree" : "Inżynier",
    "school_name" : "Politechnika Warszawska",
    "description" : "Studia inżynierskie na kierunku Informatyka",
    "field_of_study" : "Informatyka",
    "faculty" : "Wydział Informatyki, Elektroniki i Telekomunikacji",
    "start_date" : "01/10/2017"
  } ],
  "languages" : [ "Polski", "Angielski" ],
  "level" : "Mid",
  "facebook" : "https://www.facebook.com/jkasinski",
  "bio" : "Jestem grafikiem z 5-letnim doświadczeniem.",
  "pinterest" : "https://www.pinterest.com/jkasinski",
  "linkedin" : "https://www.linkedin.com/jkasinski",
  "instagram" : "https://www.instagram.com/jkasinski",
  "experience" : [ {
    "end_date" : "01/10/2021",
    "city" : "Warszawa",
    "description" : "Praca nad backendem aplikacji",
    "company" : "Google",
    "position" : "Backend developer",
    "start_date" : "01/10/2021"
  }, {
    "end_date" : "01/10/2021",
    "city" : "Warszawa",
    "description" : "Praca nad backendem aplikacji",
    "company" : "Google",
    "position" : "Backend developer",
    "start_date" : "01/10/2021"
  } ],
  "lastname" : "Kasinski",
  "tags" : [ "Bankowość", "Biznes" ],
  "skills" : [ "Kubek", "Logo" ],
  "twitter" : "https://www.twitter.com/jkasinski",
  "dribble" : "https://www.dribble.com/jkasinski",
  "location" : "Warszawa",
  "id" : 1,
  "username" : "jkasinski1"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

Znaleziono artystę o podanej nazwie użytkownika [ArtistProfileResponse](#)

401

Nie udało się pobrać artysty o podanej nazwie użytkownika

403

Nie udało się pobrać artysty o podanej nazwie użytkownika

GET /public/api/artist/getPortfolioEntries/{username}

Up

Zwraca listę wpisów z portfolio artysty o podanej nazwie użytkownika (**getPortfolioEntries**)

Path parameters

username (required)

Path Parameter — Nazwa użytkownika artysty

Query parameters

page (optional)

Query Parameter — Numer strony format: int32

size (optional)

Query Parameter — Ilość elementów na stronie format: int32

Return type

[PortfolioEntryResponse](#)

Example data

Content-Type: application/json

```
{
  "name" : "Kubek",
  "description" : "Kubek z logo",
  "id" : 1
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

Znaleziono artystę [PortfolioEntryResponse](#)

403

Nie udało się pobrać wpisów z portfolio artysty

GET /public/api/artist/getPortfolioImage/{username}/{portfolioEntryId}

Up

Zwraca zdjęcie z portfolio artysty o podanej nazwie użytkownika i id wpisu w portfolio (**getPortfolioImage**)

Path parameters

username (required)

Path Parameter — Nazwa użytkownika artysty

portfolioEntryId (required)

Path Parameter — Id wpisu w portfolio format: int64

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- image/jpeg

Responses

200

Znaleziono wpis w portfolio dla tego artysty

401

Nie udało się pobrać zdjęcia z portfolio artysty

403

Nie udało się pobrać zdjęcia z portfolio artysty

GET /public/api/artist/getProfileImageByUsername/{username}

Up

Zwraca zdjęcie profilowe artysty o podanej nazwie użytkownika (**getProfileImage**)

Path parameters

username (required)

Path Parameter — Nazwa użytkownika artysty

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- image/jpeg

Responses

200

Znaleziono zdjęcie profilowe artysty

401

Nie udało się pobrać zdjęcia profilowego artysty

403

Nie udało się pobrać zdjęcia profilowego artysty

GET /public/api/artist/getShortArtistProfileByUsername/{username}

Zwraca krótki profil artysty o podanej nazwie użytkownika ([getShortArtistProfileByUsername](#))

Path parameters

username (required)

Path Parameter — Nazwa użytkownika

Return type

[ShortArtistProfileResponse](#)

Example data

Content-Type: application/json

```
{
  "skills" : [ "skills", "skills" ],
  "firstname" : "Jakub",
  "city" : "Zdalnie",
  "level" : "Mid",
  "username" : "jkasinski1",
  "lastname" : "lastname"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

Znaleziono artystę o podanej nazwie użytkownika [ShortArtistProfileResponse](#)

401

Nie udało się pobrać artysty o podanej nazwie użytkownika

403

Nie udało się pobrać artysty o podanej nazwie użytkownika

PUT /api/artist/updateProfileByUsername/{username}

Aktualizuje profil artysty o podanej nazwie użytkownika ([updateArtistProfileByUsername](#))

Path parameters

username (required)

Path Parameter — Nazwa użytkownika artysty

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

body [ArtistProfileRequest](#) (required)

Body Parameter — Dane do aktualizacji profilu artysty

Return type

[ArtistProfileResponse](#)

Example data

Content-Type: application/json

```
{
  "firstname" : "Jakub",
  "website" : "https://www.jkasinski.com",
  "education" : [ {
    "end_date" : "01/10/2021",
    "degree" : "Inżynier",
    "school_name" : "Politechnika Warszawska",
    "description" : "Studia inżynierskie na kierunku Informatyka",
    "field_of_study" : "Informatyka",
    "faculty" : "Wydział Informatyki, Elektroniki i Telekomunikacji",
    "start_date" : "01/10/2017"
  }, {

```

```

    "end_date" : "01/10/2021",
    "degree" : "Inżynier",
    "school_name" : "Politechnika Warszawska",
    "description" : "Studia inżynierskie na kierunku Informatyka",
    "field_of_study" : "Informatyka",
    "faculty" : "Wydział Informatyki, Elektroniki i Telekomunikacji",
    "start_date" : "01/10/2017"
  } ],
  "languages" : [ "Polski", "Angielski" ],
  "level" : "Mid",
  "facebook" : "https://www.facebook.com/jkasinski",
  "bio" : "Jestem grafikiem z 5-letnim doświadczeniem.",
  "pinterest" : "https://www.pinterest.com/jkasinski",
  "linkedin" : "https://www.linkedin.com/jkasinski",
  "instagram" : "https://www.instagram.com/jkasinski",
  "experience" : [ {
    "end_date" : "01/10/2021",
    "city" : "Warszawa",
    "description" : "Praca nad backendem aplikacji",
    "company" : "Google",
    "position" : "Backend developer",
    "start_date" : "01/10/2021"
  }, {
    "end_date" : "01/10/2021",
    "city" : "Warszawa",
    "description" : "Praca nad backendem aplikacji",
    "company" : "Google",
    "position" : "Backend developer",
    "start_date" : "01/10/2021"
  } ],
  "lastname" : "Kasinski",
  "tags" : [ "Bankowość", "Biznes" ],
  "skills" : [ "Kubek", "Logo" ],
  "twitter" : "https://www.twitter.com/jkasinski",
  "dribble" : "https://www.dribble.com/jkasinski",
  "location" : "Warszawa",
  "id" : 1,
  "username" : "jkasinski1"
}

```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

Zaktualizowano profil artysty o podanej nazwie użytkownika [ArtistProfileResponse](#)

401

Autoryzacja nie powiodła się

403

Aktualizacja profilu artysty nie powiodła się

POST /api/artist/updateProfileImage/{username}

Up

Aktualizuje zdjęcie profilowe artysty o podanej nazwie użytkownika ([updateProfileImageByUsername](#))

Path parameters

username (required)

Path Parameter — Nazwa użytkownika artysty

Consumes

This API call consumes the following media types via the Content-Type request header:

- multipart/form-data

Form parameters

image (required)

Form Parameter — format: binary

Return type

[MyApiResponse](#)

Example data

Content-Type: application/json

```

{
  "success" : true,
  "message" : "Pomyślnie przetworzono zapytanie"
}

```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

Zaktualizowano zdjęcie profilowe artysty [MyApiResponse](#)

400

Nie udało się zaktualizować zdjęcia profilowego artysty

401

Autoryzacja nie powiodła się

403

Nie znaleziono artysty o podanej nazwie użytkownika

POST /api/artist/createPortfolioEntry/{username}

Dodaje wpis do portfolio artysty o podanej nazwie użytkownika (`uploadPortfolioImageByUsername`)

Path parameters

username (required)

Path Parameter — Nazwa użytkownika artysty

Consumes

This API call consumes the following media types via the Content-Type request header:

- multipart/form-data

Query parameters

name (required)

Query Parameter — Nazwa wpisu w portfolio

description (required)

Query Parameter — Opis wpisu w portfolio

Form parameters

image (required)

Form Parameter — format: binary

Return type

[MyApiResponse](#)

Example data

Content-Type: application/json

```
{
  "success" : true,
  "message" : "Pomyślnie przetworzono zapytanie"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

Dodano zdjęcie do portfolio artysty [MyApiResponse](#)

400

Nie udało się dodać wpisu do portfolio artysty

401

Autoryzacja nie powiodła się

403

Nie znaleziono artysty o podanej nazwie użytkownika

ProfilFirmy

GET /public/api/company/getProfileByUsername/{username}

Zwraca profil firmy o podanej nazwie użytkownika (`getCompanyProfileByUsername`)

Path parameters

username (required)

Path Parameter — Nazwa użytkownika

Return type

[CompanyProfileResponse](#)

Example data

Content-Type: application/json

```
{
  "website" : "www.pz.com",
  "address" : "ul. Główna 1, 00-000 Warszawa",
  "regon" : "850704562",
  "facebook" : "www.facebook.com/pz",
  "description" : "Firma zajmująca się tworzeniem stron internetowych",
  "linkedin" : "www.linkedin.com/pz",
  "instagram" : "www.instagram.com/pz",
  "krs" : "0000385161",
  "twitter" : "www.twitter.com/pz",
  "nip" : "5247289916",
  "name" : "PZ",
  "id" : 1,
  "username" : "jkasinski1"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

Znaleziono profil firmy dla podanej nazwy użytkownika [CompanyProfileResponse](#)

401

Nie udało się pobrać profilu firmy dla podanej nazwy użytkownika

GET /public/api/company/getProfileImageByUsername/{username}

Up

Zwraca zdjęcie profilowe firmy o podanej nazwie użytkownika ([getProfileImageByUsername](#))

Path parameters

username (required)

Path Parameter — Nazwa użytkownika

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- image/jpeg

Responses

200

Znaleziono zdjęcie profilowe firmy o podanej nazwie użytkownika

401

Nie udało się pobrać zdjęcia profilowego firmy dla podanej nazwy użytkownika

PUT /api/company/updateProfileByUsername/{username}

Up

Aktualizuje profil firmy o podanej nazwie użytkownika ([updateCompanyProfileByUsername](#))

Path parameters

username (required)

Path Parameter — Nazwa użytkownika

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

body [CompanyProfileRequest](#) (required)

Body Parameter —

Return type

[CompanyProfileResponse](#)

Example data

Content-Type: application/json

```
{
  "website" : "www.pz.com",
  "address" : "ul. Główna 1, 00-000 Warszawa",
  "regon" : "850704562",
  "facebook" : "www.facebook.com/pz",
  "description" : "Firma zajmująca się tworzeniem stron internetowych",
  "linkedin" : "www.linkedin.com/pz",
  "instagram" : "www.instagram.com/pz",
  "krs" : "0000385161",
  "twitter" : "www.twitter.com/pz",
}
```

```
{
  "nip" : "5247289916",
  "name" : "PZ",
  "id" : 1,
  "username" : "jkasinski1"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

Zaktualizowano profil firmy [CompanyProfileResponse](#)

401

Autoryzacja nie powiodła się

403

Nie udało się zaktualizować profilu firmy dla podanej nazwy użytkownika

POST /api/company/uploadProfileImage/{username}

Up

Aktualizuje zdjęcie profilowe firmy o podanej nazwie użytkownika ([uploadProfileImage](#))

Path parameters

username (required)

Path Parameter — Nazwa użytkownika

Consumes

This API call consumes the following media types via the Content-Type request header:

- multipart/form-data

Form parameters

file (required)

Form Parameter — format: binary

Return type

[MyApiResponse](#)

Example data

Content-Type: application/json

```
{
  "success" : true,
  "message" : "Pomyślnie przetworzono zapytanie"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

Zaktualizowano zdjęcie profilowe firmy [MyApiResponse](#)

400

Nie udało się zaktualizować zdjęcia profilowego firmy

401

Autoryzacja nie powiodła się

403

Nie udało się zaktualizować zdjęcia profilowego firmy dla podanej nazwy użytkownika

Użytkownik

GET /api/user/getAllUsernames

Up

Zwraca listę nazw użytkowników ([getAllUsernames](#))

Return type

array[String]

Example data

Content-Type: application/json

```
[ "", "" ]
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

Pomyślnie pobrano listę nazw użytkowników

500

Internal Server Error

GET /api/user/getUserById

Up

Zwraca dane użytkownika od podanym id (`getAllUsers`)

Return type

[UserResponse](#)

Example data

Content-Type: application/json

```
{
  "email" : "email",
  "username" : "username"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

Pomyślnie pobrano użytkownika [UserResponse](#)

500

Internal Server Error [UserResponse](#)

Zlecenia

POST /api/commission/create

Up

Tworzy zlecenie (`createCommission`)

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

body [CommissionRequest](#) (required)

Body Parameter — Dane zlecenia

Return type

[CommissionResponse](#)

Example data

Content-Type: application/json

```
{
  "languages" : [ "Polski", "Angielski" ],
  "level" : [ "Junior", "Mid" ],
  "description" : "Wykonanie strony internetowej dla firmy PZ",
  "title" : "Wykonanie strony internetowej",
  "is_completed" : true,
  "tags" : [ "Bankowość", "Biznes" ],
  "contractor_username" : "jkasinski2",
  "skills" : [ "Logo", "Identyfikacja wizualna" ],
  "completed_at" : "01/06/2021",
  "client_username" : "jkasinski1",
  "commissioned_at" : "2021-06-01 12:00:00",
  "rate" : 10000,
  "company_name" : "PZ",
  "location" : [ "Warszawa", "Kraków" ],
  "id" : 1,
  "deadline" : "01/10/2022"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

Zlecenie zostało utworzone [CommissionResponse](#)

400

Nie udało się utworzyć zlecenia

DELETE /api/deleteCommission/{id}

Up

Usuwa zlecenie o podanym id (**deleteCommission**)

Path parameters

id (required)

Path Parameter — Id zlecenia format: int64

Responses

200

Zlecenie zostało usunięte

401

Autoryzacja nie powiodła się

403

Nie udało się usunąć zlecenia o podanym id

POST /public/api/commission/filter

Up

Zwraca pofiltrowane zlecenia (**filterCommissions**)

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

body [CommissionFilterRequest](#) (required)

Body Parameter — Filtrowanie zleceń

Query parameters

page (optional)

Query Parameter — Numer strony format: int32

size (optional)

Query Parameter — Ilość elementów na stronie format: int32

Return type

[Page](#)

Example data

Content-Type: application/json

```
{
  "number" : 5,
  "size" : 1,
  "last" : true,
  "numberOfElements" : 5,
  "totalPages" : 6,
  "pageable" : {
    "paged" : true,
    "pageNumber" : 7,
    "offset" : 2,
    "pageSize" : 9,
    "unpaged" : true
  },
  "sort" : {
    "unsorted" : true,
    "sorted" : true,
    "empty" : true
  },
  "content" : [ { }, { } ],
  "first" : true,
  "totalElements" : 0,
  "empty" : true
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

Znaleziono zlecenia [Page](#)

400

Nie udało się pobrać zleceń

404

Nie udało się pobrać zleceń

GET /public/api/commission/getById/{id}

[Up](#)

Zwraca zlecenie o podanym id ([getCommissionById](#))

Path parameters

id (required)

Path Parameter — Id zlecenia format: int64

Return type

[CommissionResponse](#)

Example data

Content-Type: application/json

```
{
  "languages" : [ "Polski", "Angielski" ],
  "level" : [ "Junior", "Mid" ],
  "description" : "Wykonanie strony internetowej dla firmy PZ",
  "title" : "Wykonanie strony internetowej",
  "is_completed" : true,
  "tags" : [ "Bankowość", "Biznes" ],
  "contractor_username" : "jkasinski2",
  "skills" : [ "Logo", "Identyfikacja wizualna" ],
  "completed_at" : "01/06/2021",
  "client_username" : "jkasinski1",
  "commissioned_at" : "2021-06-01 12:00:00",
  "rate" : 10000,
  "company_name" : "PZ",
  "location" : [ "Warszawa", "Kraków" ],
  "id" : 1,
  "deadline" : "01/10/2022"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

Znaleziono zlecenie o podanym id [CommissionResponse](#)

401

Nie udało się pobrać zlecenia o podanym id

403

Nie udało się pobrać zlecenia o podanym id

GET /public/api/commission/getAllCommissionFirmByUsername/{username}

[Up](#)

Zwraca wszystkie zlecenia dla firmy o podanej nazwie użytkownika ([getCommissionsByUsername](#))

Path parameters

username (required)

Path Parameter — Nazwa użytkownika

Return type

[CommissionResponse](#)

Example data

Content-Type: application/json

```
{
  "languages" : [ "Polski", "Angielski" ],
  "level" : [ "Junior", "Mid" ],
  "description" : "Wykonanie strony internetowej dla firmy PZ",
  "title" : "Wykonanie strony internetowej",
  "is_completed" : true,
  "tags" : [ "Bankowość", "Biznes" ],
  "contractor_username" : "jkasinski2",
  "skills" : [ "Logo", "Identyfikacja wizualna" ],
  "completed_at" : "01/06/2021",
  "client_username" : "jkasinski1",
  "commissioned_at" : "2021-06-01 12:00:00",
  "rate" : 10000,
  "company_name" : "PZ",
  "location" : [ "Warszawa", "Kraków" ],
  "id" : 1,
  "deadline" : "01/10/2022"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

Zlecenia dla firmy o podanej nazwie użytkownika [CommissionResponse](#)

404

Nie znaleziono zleceń dla firmy o podanej nazwie użytkownika

PUT /api/commission/setCompletedById/{id}

Up

Ustawia zlecenie o podanym id jako zakończone (**setCompletedCommissionById**)

Path parameters

id (required)

Path Parameter — Id zlecenia format: int64

Return type

[CommissionResponse](#)

Example data

Content-Type: application/json

```
{
  "languages" : [ "Polski", "Angielski" ],
  "level" : [ "Junior", "Mid" ],
  "description" : "Wykonanie strony internetowej dla firmy PZ",
  "title" : "Wykonanie strony internetowej",
  "is_completed" : true,
  "tags" : [ "Bankowość", "Biznes" ],
  "contractor_username" : "jkasinski2",
  "skills" : [ "Logo", "Identyfikacja wizualna" ],
  "completed_at" : "01/06/2021",
  "client_username" : "jkasinski1",
  "commissioned_at" : "2021-06-01 12:00:00",
  "rate" : 10000,
  "company_name" : "PZ",
  "location" : [ "Warszawa", "Kraków" ],
  "id" : 1,
  "deadline" : "01/10/2022"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

Zlecenie o podanym id zostało zaktualizowane [CommissionResponse](#)

401

Autoryzacja nie powiodła się

403

Nie udało się zaktualizować zlecenia o podanym id

PUT /api/commission/updateById/{id}

Up

Aktualizuje zlecenie o podanym id (**updateCommissionById**)

Path parameters

id (required)

Path Parameter — Id zlecenia format: int64

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

body [CommissionRequest](#) (required)

Body Parameter — Dane zlecenia

Return type

[CommissionResponse](#)

Example data

Content-Type: application/json

```
{
  "languages" : [ "Polski", "Angielski" ],
  "level" : [ "Junior", "Mid" ],
  "description" : "Wykonanie strony internetowej dla firmy PZ",
}
```

```
{
  "title" : "Wykonanie strony internetowej",
  "is_completed" : true,
  "tags" : [ "Bankowość", "Biznes" ],
  "contractor_username" : "jkasinski2",
  "skills" : [ "Logo", "Identyfikacja wizualna" ],
  "completed_at" : "01/06/2021",
  "client_username" : "jkasinski1",
  "commissioned_at" : "2021-06-01 12:00:00",
  "rate" : 10000,
  "company_name" : "PZ",
  "location" : [ "Warszawa", "Kraków" ],
  "id" : 1,
  "deadline" : "01/10/2022"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

Zlecenie zostało zaktualizowane [CommissionResponse](#)

401

Autoryzacja nie powiodła się

403

Nie udało się zaktualizować zlecenia o podanym id

Models

[[Jump to Methods](#)]

Table of Contents

1. [ArtistFilterRequest](#)
2. [ArtistProfileRequest](#)
3. [ArtistProfileResponse](#)
4. [AvailableSkillsCategoriesResponse](#)
5. [CommissionFilterRequest](#)
6. [CommissionRequest](#)
7. [CommissionResponse](#)
8. [CompanyProfileRequest](#)
9. [CompanyProfileResponse](#)
10. [EducationDto](#)
11. [ExperienceDto](#)
12. [InterlocutorResponse](#)
13. [JwtAuthenticationResponse](#)
14. [LoginRequest](#)
15. [MyApiResponse](#)
16. [Page](#)
17. [PageableObject](#)
18. [PortfolioEntryResponse](#)
19. [RegisterRequest](#)
20. [ShortArtistProfileResponse](#)
21. [SkillsCategoryResponse](#)
22. [SortObject](#)
23. [UserResponse](#)
24. [createPortfolioEntry_username_body](#)
25. [updateProfileImage_username_body](#)
26. [uploadProfileImage_username_body](#)

ArtistFilterRequest

[Up](#)

levels (optional)

[array\[String\]](#) Poziomy
example: ["Junior", "Mid"]

locations (optional)

[array\[String\]](#) Lokalizacje
example: ["Warszawa", "Kraków"]

skills (optional)

[array\[String\]](#) Umiejętności
example: ["Logo", "Identyfikacja wizualna"]

languages (optional)

[array\[String\]](#) Języki
example: ["Polski", "Angielski"]

tags (optional)

[array\[String\]](#) Tagi
example: ["Bankowość", "Biznes"]

ArtistProfileRequest

[Up](#)

firstname (optional)

[String](#) Imię

example: Jakub

lastname (optional)

[String](#) Nazwisko

example: Kasinski

bio (optional)

[String](#) Opis

example: Jestem grafikiem z 5-letnim doświadczeniem.

level (optional)

[String](#) Poziomy doświadczenia

Enum:

Junior

Mid

Senior

example: Mid

location (optional)

[String](#) Lokalizacje

Enum:

Zdalnie

Warszawa

Kraków

Wrocław

Łódź

Poznań

Gdańsk

Szczecin

Bydgoszcz

Lublin

example: Warszawa

skills (optional)

[array\[String\]](#) Umiejętności

example: ["Kubek", "Logo"]

tags (optional)

[array\[String\]](#) Tagi

example: ["Bankowość", "Biznes"]

languages (optional)

[array\[String\]](#) Języki

example: ["Polski", "Angielski"]

education (optional)

[array\[EducationDto\]](#) Edukacja

experience (optional)

[array\[ExperienceDto\]](#) Doświadczenie

website (optional)

[String](#) Strona artysty

example: https://www.jkasinski.com

facebook (optional)

[String](#) Facebook

example: https://www.facebook.com/jkasinski

linkedin (optional)

[String](#) LinkedIn

example: https://www.linkedin.com/jkasinski

instagram (optional)

[String](#) Instagram

example: https://www.instagram.com/jkasinski

dribbble (optional)

[String](#) Dribble

example: https://www.dribbble.com/jkasinski

pinterest (optional)

[String](#) Pinterest

example: https://www.pinterest.com/jkasinski

twitter (optional)

[String](#) Twitter

example: https://www.twitter.com/jkasinski

ArtistProfileResponse

[Up](#)

id (optional)

[Long](#) Identyfikator format: int64

example: 1

username

[String](#) Nazwa użytkownika

example: jkasinski1

firstname

[String](#) Imię

example: Jakub

lastname

[String](#) Nazwisko

example: Kasinski

bio (optional)

[String](#) Opis

example: Jestem grafikiem z 5-letnim doświadczeniem.

level (optional)

[String](#) Poziomy doświadczenia

Enum:

Junior

Mid

Senior

example: Mid

location (optional)

[String](#) Lokalizacje

Enum:

Zdalnie

Warszawa

Kraków

Wrocław

Łódź

Poznań

Gdańsk

Szczecin

Bydgoszcz

Lublin

example: Warszawa

skills (optional)

[array\[String\]](#) Umiejętności

example: ["Kubek", "Logo"]

tags (optional)

[array\[String\]](#) Tagi

example: ["Bankowość", "Biznes"]

languages (optional)

[array\[String\]](#) Języki

example: ["Polski", "Angielski"]

education (optional)

[array\[EducationDto\]](#) Edukacja

experience (optional)

[array\[ExperienceDto\]](#) Doświadczenie

website (optional)

[String](#) Strona artysty

example: https://www.jkasinski.com

facebook (optional)

[String](#) Facebook

example: https://www.facebook.com/jkasinski

linkedin (optional)

[String](#) LinkedIn

example: https://www.linkedin.com/jkasinski

instagram (optional)

[String](#) Instagram

example: https://www.instagram.com/jkasinski

dribbble (optional)

[String](#) Dribble

example: https://www.dribbble.com/jkasinski

pinterest (optional)

[String](#) Pinterest

example: https://www.pinterest.com/jkasinski

twitter (optional)

[String](#) Twitter

example: https://www.twitter.com/jkasinski

AvailableSkillsCategoriesResponse

[Up](#)

categories (optional)

[SkillsCategoryResponse](#)

CommissionFilterRequest

[Up](#)

levels (optional)

[array\[String\]](#) Poziomy doświadczenia

example: ["Junior", "Mid"]

locations (optional)

[array\[String\]](#) Lokalizacje

example: ["Warszawa", "Kraków"]

skills (optional)

[array\[String\]](#) Umiejętności

example: ["Logo", "Identyfikacja wizualna"]

tags (optional)

[array\[String\]](#) Tagi

example: ["Bankowość", "Biznes"]

languages (optional)

[array\[String\]](#) Języki

example: ["Polski", "Angielski"]

CommissionRequest

[Up](#)

client_username

[String](#) Nazwa użytkownika klienta

example: jkasinski1

contractor_username (optional)

[String](#) Nazwa użytkownika wykonawcy

example: jkasinski2

title

[String](#) Tytuł

example: Wykonanie strony internetowej

description

[String](#) Opis

example: Wykonanie strony internetowej dla firmy PZ

deadline (optional)

[String](#) Data zakończenia

example: 01/10/2022

level (optional)

[array\[String\]](#) Poziomy

example: ["Junior", "Mid"]

location (optional)

[array\[String\]](#) Lokalizacje

example: ["Warszawa", "Kraków"]

skills (optional)

[array\[String\]](#) Umiejętności

example: ["Logo", "Identyfikacja wizualna"]

tags (optional)

[array\[String\]](#) Tagi

example: ["Bankowość", "Biznes"]

languages (optional)

[array\[String\]](#) Języki

example: ["Polski", "Angielski"]

rate (optional)

[Integer](#) Stawka format: int32

example: 10000

CommissionResponse

[Up](#)

id (optional)

[Long](#) Identyfikator zlecenia format: int64

example: 1

client_username (optional)

[String](#) Nazwa użytkownika klienta

example: jkasinski1

contractor_username

[String](#) Nazwa użytkownika wykonawcy

example: jkasinski2

title

[String](#) Tytuł

example: Wykonanie strony internetowej

company_name

[String](#)

example: PZ

description

[String](#)

example: Wykonanie strony internetowej dla firmy PZ

commissioned_at

[String](#)

example: 2021-06-01 12:00:00

deadline (optional)

[String](#)

example: 01/10/2022

completed_at (optional)

[String](#)

example: 01/06/2021

level (optional)

[array\[String\]](#) Poziomy

example: ["Junior", "Mid"]

location (optional)

[array\[String\]](#) Lokalizacje

example: ["Warszawa", "Kraków"]

skills (optional)

[array\[String\]](#) Umiejętności

example: ["Logo", "Identyfikacja wizualna"]

tags (optional)[array\[String\]](#) Tagi*example: ["Bankowość", "Biznes"]***languages (optional)**[array\[String\]](#) Języki*example: ["Polski", "Angielski"]***rate (optional)**[Integer](#) Stawka format: int32*example: 10000***is_completed (optional)**[Boolean](#) Czy zlecenie zostało zakończone*example: true***CompanyProfileRequest**[Up](#)**name**[String](#) Nazwa firmy*example: PZ***description (optional)**[String](#)*example: Firma zajmująca się tworzeniem stron internetowych***address (optional)**[String](#)*example: ul. Główna 1, 00-000 Warszawa***nip**[String](#) Numer NIP*example: 5247289916***regon**[String](#) Numer REGON*example: 850704562***krs (optional)**[String](#) Numer KRS*example: 0000385161***website (optional)**[String](#) Strona internetowa*example: www.pz.com***facebook (optional)**[String](#) Profil na Facebooku*example: www.facebook.com/pz***linkedin (optional)**[String](#) Profil na LinkedIn*example: www.linkedin.com/pz***twitter (optional)**[String](#) Profil na Twitterze*example: www.twitter.com/pz***instagram (optional)**[String](#) Profil na Instagramie*example: www.instagram.com/pz***CompanyProfileResponse**[Up](#)**id (optional)**[Long](#) Identyfikator profilu firmy format: int64*example: 1***username**[String](#) Nazwa użytkownika*example: jkasinski1***name**[String](#) Nazwa firmy*example: PZ***description (optional)**[String](#)*example: Firma zajmująca się tworzeniem stron internetowych***address (optional)**[String](#)*example: ul. Główna 1, 00-000 Warszawa***nip**[String](#) Numer NIP*example: 5247289916***regon**[String](#) Numer REGON*example: 850704562***krs (optional)**[String](#) Numer KRS*example: 0000385161***website (optional)**

[String](#) Strona internetowa

example: *www.pz.com*

facebook (optional)

[String](#) Profil na Facebooku

example: *www.facebook.com/pz*

linkedin (optional)

[String](#) Profil na LinkedIn

example: *www.linkedin.com/pz*

twitter (optional)

[String](#) Profil na Twitterze

example: *www.twitter.com/pz*

instagram (optional)

[String](#) Profil na Instagramie

example: *www.instagram.com/pz*

EducationDto

[Up](#)

school_name (optional)

[String](#) Nazwa szkoły

example: *Politechnika Warszawska*

faculty (optional)

[String](#) Wydział

example: *Wydział Informatyki, Elektroniki i Telekomunikacji*

field_of_study (optional)

[String](#) Kierunek

example: *Informatyka*

degree (optional)

[String](#) Stopień

example: *Inżynier*

start_date (optional)

[String](#) Data rozpoczęcia

example: *01/10/2017*

end_date (optional)

[String](#) Data zakończenia

example: *01/10/2021*

description (optional)

[String](#) Opis

example: *Studia inżynierskie na kierunku Informatyka*

ExperienceDto

[Up](#)

company (optional)

[String](#) Nazwa firmy

example: *Google*

city (optional)

[String](#) Miasto

example: *Warszawa*

position (optional)

[String](#) Stanowisko

example: *Backend developer*

description (optional)

[String](#) Opis

example: *Praca nad backendem aplikacji*

start_date (optional)

[String](#) Data rozpoczęcia

example: *01/10/2021*

end_date (optional)

[String](#) Data zakończenia

example: *01/10/2021*

InterlocutorResponse

[Up](#)

username

[String](#) Nazwa użytkownika

example: *jkasinski1*

first_name (optional)

[String](#) Imię

example: *Jakub*

last_name (optional)

[String](#) Nazwisko

example: *Kasinski*

company_name (optional)

[String](#) Nazwa firmy

last_message

[String](#) Ostatnia wiadomość

example: Cześć, chciałbym zająć się stworzeniem strony internetowej

JwtAuthenticationResponse

[Up](#)

accessToken (optional)
[String](#)

LoginRequest

[Up](#)

email
[String](#) Adres email
example: jkasinski1@gmail.com

password
[String](#) Hasło
example: password

MyApiResponse

[Up](#)

success (optional)
[Boolean](#) Czy status to sukces
example: true

message (optional)
[String](#) Wiadomość
example: Pomyślnie przetworzono zapytanie

Page

[Up](#)

totalElements (optional)
[Long](#) format: int64

totalPages (optional)
[Integer](#) format: int32

size (optional)
[Integer](#) format: int32

content (optional)
[array\[Object\]](#)

number (optional)
[Integer](#) format: int32

sort (optional)
[SortObject](#)

first (optional)
[Boolean](#)

last (optional)
[Boolean](#)

numberOfElements (optional)
[Integer](#) format: int32

pageable (optional)
[PageableObject](#)

empty (optional)
[Boolean](#)

PageableObject

[Up](#)

offset (optional)
[Long](#) format: int64

sort (optional)
[SortObject](#)

pageNumber (optional)
[Integer](#) format: int32

pageSize (optional)
[Integer](#) format: int32

paged (optional)
[Boolean](#)

unpaged (optional)
[Boolean](#)

PortfolioEntryResponse

[Up](#)

id (optional)
[Long](#) Identyfikator format: int64
example: 1

name (optional)
[String](#) Nazwa

example: Kubek

description (optional)

[*String*](#) Opis

example: Kubek z logo

RegisterRequest

[Up](#)

email

[*String*](#) Adres email

example: jkasinski1@Gmail.com

username

[*String*](#) Nazwa użytkownika

example: jkasinski1

password

[*String*](#) Hasło

example: password

ShortArtistProfileResponse

[Up](#)

username

[*String*](#) Nazwa użytkownika

example: jkasinski1

firstname

[*String*](#) Imię

example: Jakub

lastname

[*String*](#) Nazwisko

city (optional)

[*String*](#) Lokalizacje

Enum:

Zdalnie

Warszawa

Kraków

Wrocław

Łódź

Poznań

Gdańsk

Szczecin

Bydgoszcz

Lublin

level (optional)

[*String*](#) Poziomy doświadczenia

Enum:

Junior

Mid

Senior

example: Mid

skills (optional)

[*array\[String\]*](#) Umiejętności

SkillsCategoryResponse

[Up](#)

Kategorie umiejętności

name (optional)

[*String*](#) Nazwa kategorii

example: Logo i identyfikacja wizualna

skills (optional)

[*array\[String\]*](#) Umiejętności

example: ["Logo", "Wizytówka"]

SortObject

[Up](#)

empty (optional)

[*Boolean*](#)

sorted (optional)

[*Boolean*](#)

unsorted (optional)

[*Boolean*](#)

UserResponse

[Up](#)

email (optional)

[*String*](#)

username (optional)

[*String*](#)

createPortfolioEntry_username_body	Up
image byte[] format: binary	
updateProfileImage_username_body	Up
image byte[] format: binary	
uploadProfileImage_username_body	Up
file byte[] format: binary	