PfotenFreunde.Api

Overview

Paths

GET `/Chatroom/{id}` Get information of the specified chatroom

Parameters

Type	Name	Description	Schema
path	**id** *required*	Id of the chatroom	string (uuid)

Responses

Code	Description	Links
200	Success *Content* **text/plain** **application/json** **text/json**	No Links
403	Insufficient permissions	No Links

POST `/Chatroom` Creates a new Chatroom

Responses

Code	Description	Links
200	Success	No Links

PATCH `/Chatroom` Updates information of a chatroom

Responses

Code	Description	Links
200	Success	No Links

Code	Description	Links
403	Insufficient permissions	No Links

POST /Chatroom/{id}/messages Adds a new message to the chatroom

Parameters

Type	Name	Description	Schema
path	**id** *required*		string (uuid)

Responses

Code	Description	Links
200	Success	No Links
403	Insufficient permissions	No Links

GET /Chronicle/{id} Gets information of the chronicle

Parameters

Type	Name	Description	Schema
path	**id** *required*		string (uuid)

Responses

Code	Description	Links
200	Success *Content* **text/plain** **application/json** **text/json**	No Links

POST /Chronicle/{id}/post Adds a new post to the chronicle

Parameters

Type	Name	Description	Schema
path	**id** *required*		string (uuid)

Responses

Code	Description	Links
200	Success	No Links
403	Insufficient permissions	No Links

GET `/Friendship/{userId}` **Retrieves friend list of user**

Parameters

Type	Name	Description	Schema
path	**userId** *required*		string (uuid)

Responses

Code	Description	Links
200	Success *Content* **text/plain** **application/json** **text/json**	No Links

POST `/Friendship/{userId}` **Sends a friend request to the specified user**

Parameters

Type	Name	Description	Schema
path	**userId** *required*		string (uuid)

Responses

Code	Description	Links
200	Success	No Links

Code	Description	Links
400	User is already befriended	No Links

DELETE /Friendship/{userId} Removes a friendship

Parameters

Type	Name	Description	Schema
path	**userId** *required*		string (uuid)

Responses

Code	Description	Links
200	Success	No Links
400	User is not befriended	No Links

GET /Institution Gets information about the institution

Responses

Code	Description	Links
200	Success *Content* **text/plain** **application/json** **text/json**	No Links

POST /Institution Creates a new institution

Responses

Code	Description	Links
200	Success *Content* **text/plain** **application/json** **text/json**	No Links

PATCH `/Institution/{id}` Updates information of an institution

Parameters

Type	Name	Description	Schema
path	**id** *required*		string (uuid)

Responses

Code	Description	Links
200	Success	No Links
403	Insufficient permissions	No Links

POST `/Login` Creates a new login

Responses

Code	Description	Links
200	Success	No Links

DELETE `/Login` Removes a login and its associated data (User, Pets, ...)

Responses

Code	Description	Links
200	Success	No Links

GET `/Me` Returns information about the current session user

Responses

Code	Description	Links
200	Success *Content* **text/plain** **application/json** **text/json**	No Links

GET `/Person/{id}` Gets information about the specified person

Parameters

Type	Name	Description	Schema
path	**id** *required*		string (uuid)

Responses

Code	Description	Links
200	Success *Content* **text/plain** **application/json** **text/json**	No Links

PATCH `/Person/{id}` Updates information of the specified person

Parameters

Type	Name	Description	Schema
path	**id** *required*		string (uuid)

Responses

Code	Description	Links
200	Success	No Links
403	Insufficient permissions	No Links

POST /Person Adds a new person

Responses

Code	Description	Links
200	Success	No Links

GET /Person/{id}/Pets Gets the list of pets of the specified person

Parameters

Type	Name	Description	Schema
path	**id** *required*		string (uuid)

Responses

Code	Description	Links
200	Success *Content* **text/plain** **application/json** **text/json**	No Links

GET /Pet/{id} Gets information about the specified pet

Parameters

Type	Name	Description	Schema
path	**id** *required*		string (uuid)

Responses

Code	Description	Links
200	Success *Content* **text/plain** **application/json** **text/json**	No Links

DELETE `/Pet/{id}` Deletes the specified pet

Parameters

Type	Name	Description	Schema
path	**id** *required*		string (uuid)

Responses

Code	Description	Links
200	Success	No Links
403	Insufficient permissions	No Links
404	Pet does not exist	No Links

PATCH `/Pet/{id}` Updates information about the specified pet

Parameters

Type	Name	Description	Schema
path	**id** *required*		string (uuid)

Responses

Code	Description	Links
200	Success	No Links

POST `/Pet` Adds a new pet

Responses

Code	Description	Links
200	Success	No Links

GET `/Reports` Gets all open reports

Responses

Code	Description	Links
200	Success *Content* **text/plain** **application/json** **text/json**	No Links
403	Insufficient permissions	No Links

DELETE /Reports/{id} **Deletes a report**

Parameters

Type	Name	Description	Schema
path	**id** *required*		string (uuid)

Responses

Code	Description	Links
200	Success	No Links

GET /Search/user **Searches for a user**

Parameters

Type	Name	Description	Schema
query	**query** *optional*		string

Responses

Code	Description	Links
200	Success *Content* **text/plain** **application/json** **text/json**	No Links

GET /Search/pet **Searches for a pet**

Parameters

Type	Name	Description	Schema
query	**query** *optional*		string

Responses

Code	Description	Links
200	Success *Content* **text/plain** **application/json** **text/json**	No Links

POST /Session **Creates a new session**

Responses

Code	Description	Links
200	Success *Content* **text/plain** **application/json** **text/json**	No Links

DELETE /Session/{id} **Deletes the specified session**

Parameters

Type	Name	Description	Schema
path	**id** *required*		string (uuid)

Responses

Code	Description	Links
200	Success	No Links
404	Session does not exist	No Links

GET /User **Gets all users**

Code	Description	Links
200	Success *Content* **text/plain** **application/json** **text/json**	No Links

GET /User/{id} **Gets the specified user**

Parameters

Type	Name	Description	Schema
path	**id** *required*		string (uuid)

Responses

Code	Description	Links
200	Success *Content* **text/plain** **application/json** **text/json**	No Links

DELETE /User/{id} **Deletes the specified user**

Parameters

Type	Name	Description	Schema
path	**id** *required*		string (uuid)

Responses

Code	Description	Links
200	Success *Content* **text/plain** **application/json** **text/json**	No Links
403	Insufficient permissions	No Links
404	User does not exist	No Links

POST /User/{id}/pause **Pauses the subscription of the specified user**

Parameters

Type	Name	Description	Schema
path	**id** *required*		string (uuid)

Responses

Code	Description	Links
200	Success *Content* **text/plain** **application/json** **text/json**	No Links
403	Insufficient permissions	No Links
404	User does not exist	No Links

POST /User/{id}/report **Reports the specified user**

Parameters

Type	Name	Description	Schema
path	**id** *required*		string

Responses

Code	Description	Links
200	Success *Content* **text/plain** **application/json** **text/json**	No Links
404	User does not exist	No Links

POST `/User/{id}/lock` **Locks the user for the specified time in milliseconds**

Parameters

Type	Name	Description	Schema
query	**time** *optional*		integer (int64)
path	**id** *required*		string

Responses

Code	Description	Links
200	Success *Content* **text/plain** **application/json** **text/json**	No Links
403	Insufficient permissions	No Links
404	User does not exist	No Links

Components

Schemas

RatingType

Rating

Properties

Name	Description	Schema
id *optional*		string (uuid)
sendAt *optional*		string (date-time)
ratingType *optional*		RatingType

Picture

Properties

Name	Description	Schema
id *optional*		string (uuid)
name *optional*	*nullable*	string
path *optional*	*nullable*	string
uploadAt *optional*		string (date-time)

User

Properties

Name	Description	Schema
id *optional*		string (uuid)
name *optional*	*nullable*	string
phone *optional*	*nullable*	string
mail *optional*	*nullable*	string
ratings *optional*	*nullable*	< Rating > array
pictures *optional*	*nullable*	< Picture > array

Message

Properties

Name	Description	Schema
id *optional*		string (uuid)
sendAt *optional*		string (date-time)
seenAt *optional*		string (date-time)
text *optional*	*nullable*	string
sender *optional*		User

Chatroom

Properties

Name	Description	Schema
id *optional*		string (uuid)
startedAt *optional*		string (date-time)
users *optional*	*nullable*	< User > array
messages *optional*	*nullable*	< Message > array

Post

Properties

Name	Description	Schema
id *optional*		string (uuid)
postedAt *optional*		string (date-time)
message *optional*	*nullable*	string

Name	Description	Schema
user *optional*		User
picture *optional*		Picture

Chronicle

Properties

Name	Description	Schema
id *optional*		string (uuid)
updatedAt *optional*		string (date-time)
type *optional*	*nullable*	< Post > array

FriendList

Properties

Name	Description	Schema
id *optional*		string (uuid)
users *optional*	*nullable*	< User > array

Institution

Properties

Name	Description	Schema
id *optional*		string (uuid)
name *optional*	*nullable*	string
phone *optional*	*nullable*	string
mail *optional*	*nullable*	string
ratings *optional*	*nullable*	< Rating > array

Name	Description	Schema
pictures *optional*	*nullable*	< Picture > array
homepage *optional*	*nullable*	string

LoginType

Login

Properties

Name	Description	Schema
username *optional*	*nullable*	string
password *optional*	*nullable*	string
type *optional*		LoginType

Me

Properties

Name	Description	Schema
userId *optional*		string (uuid)
type *optional*		LoginType
subscripedUntil *optional*		string (date-time)

Person

Properties

Name	Description	Schema
id *optional*		string (uuid)
name *optional*	*nullable*	string

Name	Description	Schema
phone *optional*	*nullable*	string
mail *optional*	*nullable*	string
ratings *optional*	*nullable*	< Rating > array
pictures *optional*	*nullable*	< Picture > array
surname *optional*	*nullable*	string
age *optional*		integer (int32)
sex *optional*	*nullable*	string

PetSpecies

Pet

Properties

Name	Description	Schema
id *optional*		string (uuid)
species *optional*		PetSpecies
attributes *optional*	*nullable*	object
preference *optional*	*nullable*	object
pictures *optional*	*nullable*	< Picture > array
owner *optional*		Person

ReportType

ReportStatus

Report

Properties

Name	Description	Schema
id *optional*		string (uuid)
sendAt *optional*		string (date-time)
updatedAt *optional*		string (date-time)
comment *optional*	*nullable*	string
type *optional*		ReportType
status *optional*		ReportStatus
users *optional*	*nullable*	< User > array

Session

Properties

Name	Description	Schema
id *optional*		string (uuid)
user *optional*		User
validUntil *optional*		string (date-time)