## Contents

USERS		
POST	<mark>/api/v1/login</mark>	3
POST	/api/v1/account/logout	3
POST	/api/v1/users	3
GET	/api/v1/users	
GET	/api/v1/users/{email}	
PUT	/api/v1/users	
DELE.	TE /api/v1/users	
PHOTOS	<del></del>	
POST		
PUT	/api/v1/photos/{id}	
DELE.	_ <del></del>	
GET	/api/v1/photos/count/{state}	
	S PUBLIC	
GET	/api/v1/photos/public/hot/{beg}/{end}	
GET	/api/v1/photos/public/trending/{beg}/{end}	
GET	/api/v1/photos/public/fresh/{beg}/{end}	
GET	/api/v1/photos/public/tags/any/{tags}	
GET	/api/v1/photos/public/tags/all/{tags}	
PHOTOS	S PRIVATE	
GET	/api/v1/photos	10
GET	/api/v1/photos/id/{id}	10
GET	/api/v1/photos/name/{name}	
GET	/api/v1/photos/shared	
GET	/api/v1/photos /tags/any/{tags}	12
GET	/api/v1/photos /tags/all/{tags}	12
GET	/api/v1/photos /categories/any/{tags}	13
GET	/api/v1/photos / categories /all/{tags}	
GET	/api/v1/photos /nocat	
GET	/api/v1/photos /archived	
IMAGES		
POST	/api/v1/images/{photoId}	15

<b>GET</b>	/api/v1/images/{photoid}	15
CATEGO	PRIES	16
POST	/api/v1/categories	16
GET	/api/v1/categories/{parentid}	16
<b>GET</b>	/api/v1/categories	17
PUT	/api/v1/categories/{id}	17
DELET	/api/v1/categories/{id}	17
POST	/api/v1/category	18
<b>PUT</b>	/api/v1/category/{photoId} /{categoryId}	18
DELET	TE /api/v1/category/{photoid}	18
TAGS		19
POST	/api/v1/tags	19
<b>POST</b>	/api/v1/tags/	19
<b>GET</b>	/api/v1/tags	19
GET	/api/v1/tags/{letters}	20
<b>GET</b>	/api/v1/tags/public	
GET	/api/v1/tags/public/{letters}	20
PUT	/api/v1/tag/{id}/{name}	21
DELET	FE /api/v1/tag/{id}	21
SHARES.		22
<b>POST</b>	/api/v1/shares	22
DELET	ΓΕ /api/v1/shares/{id}	22
RATES		
<b>POST</b>	- 1 - 1 - 1 - 1 - 1 - 1	
DELET	TE /api/v1/rates/{photoid}	23

# **USERS**

## POST /api/v1/login

**Description:** 

User authorization.

#### Parameters:

- In body:
  - email type string (required)
     password type string (required)

#### Responses:

- 200 OK
  - o JWT in Cookie
- 401 Wrong credentials

.....

## POST /api/v1/account/logout

Description:

Logout user, adding token to blacklist.

#### Responses:

• 200 OK

\_\_\_\_\_\_

## POST /api/v1/users

#### **Description:**

Create user. Create new catalog in main server catalog with name like user's email – place where user's photos will be stored.

#### Parameters:

• In body:

0	email – type string	(required)
0	password – type string	(required)
0	firstName – type string	(required)
0	lastName – type string	(required)

- 200 OK
- 400 Bad request

## GET /api/v1/users

**Description:** 

Load list of app users data.

#### Responses:

- 200 OK
  - o userID type long
  - o email type string
  - firstName type string
  - o lastName type string
  - o password type string
  - o role type string
- 401 Unauthorized

\_\_\_\_\_\_

## GET /api/v1/users/{email}

Description:

Load user data, or id if it is not current user.

#### **Parameters:**

- In path:
  - o email type string (required)

## Responses:

- 200 OK
  - o userID type long
  - o email type string
  - firstName type string
  - lastName type string
  - o password type string
  - o role type string

#### OR

- o userID type long
- o email type string
- 400 BAD REQUEST

## PUT /api/v1/users

**Description:** 

Edit user.

#### Parameters:

- In body:
  - password type string (not required)
     firstName type string (not required)
  - lastName type string (not required)

## **Responses:**

- 200 OK
- 401 Unauthorized

\_\_\_\_\_

## DELETE /api/v1/users

**Description:** 

Remove user and all related data.

- 200 OK
- 400 Bad request

## **PHOTOS**

## POST /api/v1/photos

**Description:** 

Create photo info.

#### Parameters:

- In body:
  - name type string
     description type string
     shareState type string
     photoState type string
     (not required default "PRIVATE")
     (not required default "ACTIVE")

#### Responses:

- 200 OK
  - o id type long
- 400 Bad request

## PUT /api/v1/photos/{id}

**Description:** 

Edit photo info.

#### Parameters:

- In body:
  - o description type string (not required)
  - shareState type string (not required)
  - photoState type string (not required)

- 200 OK
- 400 Bad request

# DELETE /api/v1/photos/{id}

## Description:

Remove photo and all related data, include image.

#### Parameters:

- In path:
  - id type long (required)

## **Responses:**

- 200 OK
- 400 Bad request
- •

## GET /api/v1/photos/count/{state}

## Description:

Returns number of photos by its state.

## Parameters:

- In path:
  - o state PUBLIC | PRIVATE (required)

- 200 OK
  - o count type long
- 204 No content

# **PHOTOS PUBLIC**

## GET /api/v1/photos/public/hot/{beg}/{end}

**Description:** 

Load list of public photos sorted by sorted by rates count descending.

## GET /api/v1/photos/public/trending/{beg}/{end}

**Description:** 

Load list of public photos from last 3 days sorted by rates count descending.

## GET /api/v1/photos/public/fresh/{beg}/{end}

**Description:** 

Load list of public photos sorted by date descending.

- 200 OK
  - o photoID type long
  - o name type string
  - user type string
  - o uploadTime type timestamp
  - o description type string
  - o rate type int
  - o tags
    - tagID type long
    - name type string
    - photo\_id type long
- 204 No content

## GET /api/v1/photos/public/tags/any/{tags}

#### **Description:**

Load list of public photos which has any of tags sorted by rate descending.

## GET /api/v1/photos/public/tags/all/{tags}

#### **Description:**

Load list of public photos which has all of tags sorted by rate descending.

#### **Parameters:**

- In path:
  - o tags type string (required) example " .../all/1,2,3...

- 200 OK
  - o photoID type long
  - o name type string
  - o user type string
  - o uploadTime type timestamp
  - o description type string
  - o rate type int
  - o tags
    - tagID type long
    - name type string
    - photo\_id type long
- 204 No content

## PHOTOS PRIVATE

## GET /api/v1/photos

## **Description:**

Load list of private photos info sorted by upload time descending.

#### Responses:

- 200 OK
  - o photoID type long
  - o name type string
  - user type string
  - o uploadTime type timestamp
  - o description type string
  - o rate type int
  - o tags
    - tagID type long
    - name type string
    - photo\_id type long
- 204 No content

\_\_\_\_\_\_

## GET /api/v1/photos/id/{id}

## **Description:**

Load private photo by id.

#### **Parameters:**

- In path:
  - id type long (required)

- 200 OK
  - o photoID type long
  - o name type string
  - user type string
  - o uploadTime type timestamp
  - o description type string
  - o rate type int
  - o tags
    - tagID type long
    - name type string
    - photo\_id type long
- 204 No content

# GET /api/v1/photos/name/{name}

#### **Description:**

Load list of private photos by name.

#### **Parameters:**

- In path:
  - name type string (required)

#### **Responses:**

- 200 OK
  - o photoID type long
  - o name type string
  - o user type string
  - uploadTime type timestamp
  - o description type string
  - o rate type int
  - o tags
    - tagID type long
    - name type string
    - photo\_id type long
- 204 No content

#### GET /api/v1/photos/shared

#### **Description:**

Load list of private photos which are shared to the current user sorted by upload time descending.

- 200 OK
  - o photoID type long
  - o name type string
  - user type string
  - o uploadTime type timestamp
  - o description type string
  - o rate type int
  - o tags
    - tagID type long
    - name type string
    - photo\_id type long
- 204 No content

## GET /api/v1/photos /tags/any/{tags}

#### **Description:**

Load list of private photos which has any of tags sorted by rate descending.

## GET /api/v1/photos /tags/all/{tags}

#### **Description:**

Load list of private photos which has all of tags sorted by rate descending.

#### **Parameters:**

- In path:
  - o tags type string (required) example " .../all|any/1,2,3... "

- 200 OK
  - o photoID type long
  - o name type string
  - user type string
  - o uploadTime type timestamp
  - o description type string
  - o rate type int
  - o tags
    - tagID type long
    - name type string
    - photo\_id type long
- 204 No content

## GET /api/v1/photos /categories/any/{tags}

#### **Description:**

Load list of private photos which has any of categories sorted by rate descending.

## GET /api/v1/photos / categories /all/{tags}

#### **Description:**

Load list of private photos which has all of categories sorted by rate descending.

#### **Parameters:**

- In path:
  - o categories type string (required) example " .../all|any/1,2,3... "

- 200 OK
  - o photoID type long
  - o name type string
  - user type string
  - o uploadTime type timestamp
  - o description type string
  - o rate type int
  - o tags
    - tagID type long
    - name type string
    - photo\_id type long
- 204 No content

# GET /api/v1/photos /nocat

## **Description:**

Load list of private photos with no category assignment.

## GET /api/v1/photos /archived

## Description:

Load list of private, archived photos.

- 200 OK
  - o photoID type long
  - o name type string
  - user type string
  - uploadTime type timestamp
  - o description type string
  - o rate type int
  - o tags
    - tagID type long
    - name type string
    - photo\_id type long
- 204 No content

## **IMAGES**

# POST /api/v1/images/{photoId}

## **Description:**

Create image. Only logged user can call this method. User must have his own catalog { catalog name = user's e-mail } in the main server catalog. Before uploading image, photo info must be sent to the server.

#### **Parameters:**

- In path
  - photoId type long (required)
- In body / in form-data:
  - Key: "file", Value: multipartFile (required)

#### Responses:

- 201 Created
- 400 Bad request

.....

## GET /api/v1/images/{photoid}

#### **Description:**

Load image by id.

#### Parameters:

- In path:
  - photoid type long (required)

- 200 OK
  - Header contentType image/jpeg
  - o type file
- 204 No content

# **CATEGORIES**

## POST /api/v1/categories

**Description:** 

Create new category.

#### Parameters:

- In body:
  - o name type string (required)
  - o parentCategory type long (not required default NULL)

#### Responses:

- 200 OK
- 400 Bad request

\_\_\_\_\_\_

## GET /api/v1/categories/{parentid}

**Description:** 

Load list of user's categories – children of parentid.

#### Parameters:

- In path:
  - o parentid type long (required)

- 200 OK
  - o categoryID type long
  - o name type string
  - user type long
  - o parentCategory type long
- 204 No content

# /api/v1/categories GET **Description:** Load list of user's root categories. Responses: 200 OK o categoryID – type long o name – type string o user – type long o parentCategory - type long 204 No content PUT /api/v1/categories/{id} **Description:** Edit category. Parameters: In path ○ Id – type long (required) • In body: o name – type string (required) o parentCategory - type long (not required – default NULL) Responses: • 200 OK • 400 Bad request DELETE /api/v1/categories/{id} **Description:** Remove category and its kids. Parameters: • In path:

(required)

## Responses:

- 200 OK
- 400 Bad request

o id – type long

# POST /api/v1/category **Description:** Assigns category to the photo. Parameters: In body: o category – type long (id) (required) photo - type long (id) (required) Responses: • 200 OK • 400 Bad request PUT /api/v1/category/{photoId}/{categoryId} Description: Assigns new category to the photo. Parameters: • In path photoId – type long (required) o categoryld – type long (required) Responses: 200 OK • 400 Bad request DELETE /api/v1/category/{photoid}

Description:

Remove category assignment.

Parameters:

• In path:

photoid – type long (required)

- 200 OK
- 400 Bad request

# **TAGS**

# POST /api/v1/tags **Description:** Create new tags. Parameters: In body: List: name – type string (required) photo – type long (required) **Responses:** • 200 OK • 400 Bad request POST /api/v1/tags/ **Description:** Create new tag. **Parameters:** • In body: (required) name – type string photo – type long (required) Responses: 200 OK 400 Bad request GET /api/v1/tags **Description:** Load list of user's tags. Responses: 200 OK o tagID – type long photo – type long o user – type long

204 No content

name - type string

## GET /api/v1/tags/{letters}

**Description:** 

Load list of user's tags started with letters.

#### **Parameters:**

- In path
  - letters type string (required)

### Responses:

- 200 OK
  - tagID type long
  - o photo type long
  - o user type long
  - o name type string
- 204 No content

## GET /api/v1/tags/public

**Description:** 

Load list of public tags.

#### Responses:

- 200 OK
  - tagID type long
  - o photo type long
  - o user type long
  - o name type string
- 204 No content

\_\_\_\_\_

# GET /api/v1/tags/public/{letters}

**Description:** 

Load list of public tags started with letters.

#### Parameters:

- In path
  - letters type string (required)

Respons	coc·		
•	200 OK  o tagID – type long o photo – type long o user – type long o name – type string		
•	204 No content		
	PUT /api/v1/tag/{id}/{name} Description:		
	Edit tag.		
Parame	ters:		
•	In path  o id – type long (required)  name – type string (required)		
Respons	ses:		
•	200 OK 400 Bad request		
DELETE Descript	/api/v1/tag/{id} tion:		
	Remove tag.		

• In path:

o id – type long (required)

## Responses:

Parameters:

- 200 OK
- 400 Bad request

# **SHARES**

## POST /api/v1/shares

## **Description:**

Create new share. Before adding share application must send request for user id by /users/{email}.

## Parameters:

- In body:
  - o user type long (required)
  - photo type long(required)

#### Responses:

- 200 OK
- 400 Bad request

.....

## DELETE /api/v1/shares/{id}

## **Description:**

Remove share.

## Parameters:

- In path:
  - id type long (required)

- 200 OK
- 400 Bad request

# RATES

PUST	/api/vi/races/{priocold}	
Descrip	otion:	
	Create new rate.	
Parame	eters:	
•	In path:  o photoid – type long	(required)
Respo	nses:	
•	201 Created 400 Bad request	
DELETE Descrip	/api/v1/rates/{photoid}	
	Remove rate.	
Parame	eters:	
•	In path:  o photoid – type long	(required)
Respon	nses:	
•	200 OK	