Contents

USERS		3
POST	<mark>/api/v1/login</mark>	3
POST	/api/v1/account/logout	3
POST	<mark>/api/v1/users</mark>	3
GET	/api/v1/users	4
GET	/api/v1/users/{email}	4
<mark>PUT</mark>	/api/v1/users	5
DELET	TE /api/v1/users	5
РНОТО	S	6
POST	/api/v1/photos	6
<mark>PUT</mark>	<mark>/api/v1/photos/{id}</mark>	6
DELET	TE /api/v1/photos/{id}	7
GET	/api/v1/photos/count/{state}	7
PHOTOS	S PUBLIC	8
GET	/api/v1/photos/public/hot/{beg}/{end}	8
GET	/api/v1/photos/public/trending/{beg}/{end}	8
GET	/api/v1/photos/public/fresh/{beg}/{end}	8
GET	/api/v1/photos/public/tags/any/{tags}	9
GET	/api/v1/photos/public/tags/all/{tags}	9
PHOTOS	S PRIVATE	11
GET	<mark>/api/v1/photos</mark>	11
GET	/api/v1/photos/id/{id}	11
GET	/api/v1/photos/name/{name}	12
GET	/api/v1/photos/shared	12
GET	/api/v1/photos /tags/any/{tags}	13
GET	/api/v1/photos /tags/all/{tags}	13
GET	/api/v1/photos /categories/any/{tags}	14
GET	/api/v1/photos / categories /all/{tags}	14
IMAGES	5	15
POST	/api/v1/images/{photoId}	15
GET	/api/v1/images/{photoid}	15
CATEGO	DDIEC	16

			4.0
P	OST	/api/v1/categories	
G	ET	/api/v1/categories/{parentid}	16
G	ET	/api/v1/categories	17
P	UT	/api/v1/categories/{id}	17
D	ELETE	/api/v1/categories/{id}	17
P	OST	/api/v1/category	18
PI	UT	/api/v1/category/{photoId} /{categoryId}	18
D	ELETE	/api/v1/category/{photoid}	18
TAG	S		19
P	OST	/api/v1/tags	19
P	OST	/api/v1/tags/	19
G	ET	/api/v1/tags	19
G	ET	/api/v1/tags/{letters}	20
G	ET	/api/v1/tags/public	20
G	ET	/api/v1/tags/public/{letters}	20
PI	UT	/api/v1/tag/{id}/{name}	
D	ELETE	/api/v1/tag/{id}	21
SHA	RES		
P	OST	/api/v1/shares	22
	ELETE		
RAT			
		/api/v1/rates/{photoId}	
<u> </u>	OST		
D	ELETE	/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
 - 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
 - password type string
 - o role type string

OR

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

PUT /api/v1/users

Description:

Edit user.

Parameters:

- In body:
 - o password type string (not required)
 - o 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
 - id type long
- 400 Bad request

PUT /api/v1/photos/{id}

Description:

Edit photo info.

Parameters:

- In body:
 - 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
 - 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
 - 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/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
 - 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
 - uploadTime type timestamp
 - 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
 - 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

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

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:
 - 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:

○ id – type long (required)

- 200 OK
- 400 Bad request

l) (required) (required)

(required) equired)
(required)

Parameters:

• In path:

photoid – type long (required)

- 200 OK
- 400 Bad request

TAGS

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

204 No content

user – type longname – 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
 - o 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
 - o letters type string (required)

Respon	nses:
•	200 OK tagID – type long photo – type long user – type long name – type string 204 No content
PUT	/api/v1/tag/{id}/{name}
Descrip	otion:
	Edit tag.
Parame	eters:
•	In path o id – type long (required) name – type string (required)
Respon	nses:
•	200 OK 400 Bad request
DELETE Descrip	
	Remove tag

Remove tag.

Parameters:

- In path:
 - (required) ○ id – type long

- 200 OK
- 400 Bad request

SHARES

POST /a	api/v	/1/sł	nares
---------	-------	-------	-------

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)
 - o photo type long(required)

Responses:

- 200 OK
- 400 Bad request

DELETE /api/v1/shares/{id}

Description:

Remove share.

Parameters:

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

- 200 OK
- 400 Bad request

RATES

2051	/api/v1/rates/{pnotoid}
Descrip	tion:
	Create new rate.
Parame	iters:
•	In path: o photoId – type long (required)
Respor	nses:
•	201 Created 400 Bad request
	/api/v1/rates/{photoid}
Descrip	tion:
	Remove rate.
Parame	ters:
•	In path: o photoId – type long (required)
Respon	ses:
•	200 OK 400 Bad request