

Contents

USERS	3
POST /api/v1/login	3
POST /api/v1/users	3
GET /api/v1/users	4
GET /api/v1/users/{email}	4
PUT /api/v1/users	5
DELETE /api/v1/users	5
PHOTOS	6
POST /api/v1/photos	6
GET /api/v1/photos	6
GET /api/v1/photos/id/{id}	6
GET /api/v1/photos/name/{name}	7
GET /api/v1/photos/public	8
GET /api/v1/photos/shared	8
PUT /api/v1/photos/{id}	9
DELETE /api/v1/photos/{id}	9
IMAGES	10
POST /api/v1/images	10
GET /api/v1/images/{photoid}	10
CATEGORIES	11
POST /api/v1/categories	11
GET /api/v1/categories/{parentid}	11
GET /api/v1/categories	12
PUT /api/v1/categories/{id}	12
DELETE /api/v1/categories/{id}	12
TAGS	13
POST /api/v1/tags	13
GET /api/v1/tags	13
GET /api/v1/tags/{letters}	14
GET /api/v1/tags/public	14
GET /api/v1/tags/public/{letters}	14
PUT /api/v1/tag/{id}/{name}	15

DELETE	/api/v1/tag/{id}	15
SHARES		16
POST	/api/v1/shares	16
GET	/api/v1/shares	16
DELETE	/api/v1/shares/{id}	16
RATES		17
POST	/api/v1/rates	17
GET	/api/v1/rates	17
PUT	/api/v1/rates/{photoid}/{rate}	17
DELETE	/api/v1/rates/{photoid}	18

USERS

POST /api/v1/login

Description:

User authorization.

Parameters:

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

Responses:

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

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:
 - email – string (required)
 - password – string (required)
 - firstName – string (required)
 - lastName – string (required)

Responses:

- 200 OK
- 400 Bad request

GET /api/v1/users

Description:

Load list of app users data.

Responses:

- **200 OK**
 - userID – long
 - email – string
 - firstName – string
 - lastName – string
 - password – string
 - role – string
 - **401 Unauthorized**
-

GET /api/v1/users/{email}

Description:

Load user data.

Parameters:

- **In path:**
 - email – string (required)

Responses:

- **200 OK**
 - userID – long
 - email – string
 - firstName – string
 - lastName – string
 - password – string
 - role – string
- **401 Unauthorized**

PUT /api/v1/users

Description:

Edit user.

Parameters:

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

Responses:

- 200 OK
 - 401 Unauthorized
-

DELETE /api/v1/users

Description:

Remove user and all related data.

Responses:

- 200 OK
- 400 Bad request

PHOTOS

POST /api/v1/photos

Description:

Create photo info.

Parameters:

- In body:
 - name – string (required)
 - description – string (not required – default NULL)
 - shareState – string (not required – default “PRIVATE”)
 - photoState – string (not required – default “ACTIVE”)

Responses:

- 200 OK
 - 400 Bad request
-

GET /api/v1/photos

Description:

Load list of user's photos info sorted by upload time from the latest.

Responses:

- 200 OK
 - photoID – long
 - name – string
 - user – string
 - userID – long
 - path – string
 - uploadTime – timestamp
 - description – string
 - shareState – string
 - photoState – string
 - rate – byte
 - tags – Array < TAG >
 - category - Category
 - 204 No content
-

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

Description:

Load user's photo by id.

Parameters:

- **In path:**
 - **id – long** (required)

Responses:

- **200 OK**
 - **photoID – long**
 - **name – string**
 - **user – string**
 - **userID – long**
 - **path – string**
 - **uploadTime – timestamp**
 - **description – string**
 - **shareState – string**
 - **photoState – string**
 - **rate – byte**
 - **tags – Array < TAG>**
 - **category - Category**
 - **204 No content**
-

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

Description:

Load user's photo by name.

Parameters:

- **In path:**
 - **name – string** (required)

Responses:

- **200 OK**
 - **photoID – long**
 - **name – string**
 - **user – string**
 - **userID – long**
 - **path – string**
 - **uploadTime – timestamp**
 - **description – string**
 - **shareState – string**
 - **photoState – string**
 - **rate – byte**
 - **tags – Array < TAG>**
 - **category - Category**
- **204 No content**

GET /api/v1/photos/public

Description:

Load list of public photos info sorted by rate descending.

Responses:

- **200 OK**
 - photoID – long
 - name – string
 - user – string
 - userID – long
 - path – string
 - uploadTime – timestamp
 - description – string
 - shareState – string
 - photoState – string
 - rate – byte
 - tags – Array < TAG>
 - category - Category
 - **204 No content**
-

GET /api/v1/photos/shared

Description:

Load list of public photos info sorted by upload time from the latest.

Responses:

- **200 OK**
 - photoID – long
 - name – string
 - user – string
 - userID – long
 - path – string
 - uploadTime – timestamp
 - description – string
 - shareState – string
 - photoState – string
 - rate – byte
 - tags – Array < TAG>
 - category - Category
- **204 No content**

PUT /api/v1/photos/{id}

Description:

Edit photo info.

Parameters:

- **In body:**
 - **description** – string (not required)
 - **shareState** – string (not required)
 - **photoState** – string (not required)

Responses:

- **200 OK**
 - **400 Bad request**
-

DELETE /api/v1/photos/{id}

Description:

Remove photo and all related data, include image.

Parameters:

- **In path:**
 - **id** – long(required)

Responses:

- **200 OK**
- **400 Bad request**

IMAGES

POST /api/v1/images

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 with the same name must be sent to server.

Parameters:

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

Responses:

- 200 OK
 - 400 Bad request
-

GET /api/v1/images/{photoid}

Description:

Load image by id.

Parameters:

- In path:
 - photoid – long (required)

Responses:

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

CATEGORIES

POST /api/v1/categories

Description:

Create new category.

Parameters:

- In body:
 - name – string (required)
 - parentCategory - 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 – long (required)

Responses:

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

GET /api/v1/categories

Description:

Load list of user's root categories.

Responses:

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

PUT /api/v1/categories/{id}

Description:

Edit category.

Parameters:

- In path
 - id – long (required)
- In body:
 - name – string (required)
 - parentCategory - 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 – long (required)

Responses:

- 200 OK
- 400 Bad request

TAGS

POST /api/v1/tags

Description:

Create new tag/tags.

Parameters:

- In body:
 - List:
 - name – string (required)
 - photo – long (required)

Responses:

- 200 OK
 - 400 Bad request
-

GET /api/v1/tags

Description:

Load list of user's tags.

Responses:

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

GET /api/v1/tags/{letters}

Description:

Load list of user's tags started with letters.

Parameters:

- In path
 - letters – string (required)

Responses:

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

GET /api/v1/tags/public

Description:

Load list of public tags.

Responses:

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

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

Description:

Load list of public tags started with letters.

Parameters:

- In path
 - letters – string (required)

Responses:

- **200 OK**
 - tagID – long
 - photo – long
 - user – long
 - name – string
 - **204 No content**
-

PUT /api/v1/tag/{id}/{name}

Description:

Edit tag.

Parameters:

- **In path**
 - id – long (required)
 - name – string (required)

Responses:

- **200 OK**
 - **400 Bad request**
-

DELETE /api/v1/tag/{id}

Description:

Remove tag.

Parameters:

- **In path:**
 - id – long (required)

Responses:

- **200 OK**
- **400 Bad request**

SHARES

POST /api/v1/shares

Description:

Create new share.

Parameters:

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

Responses:

- 200 OK
 - 400 Bad request
-

GET /api/v1/shares

Description:

Load list of all shares.

Responses:

- 200 OK
 - shareID – long
 - user – long
 - photo – long
 - owner – long
 - 204 No content
-

DELETE /api/v1/shares/{id}

Description:

Remove share.

Parameters:

- In path:
 - id – long (required)

Responses:

- 200 OK
- 400 Bad request

RATES

POST /api/v1/rates

Description:

Create new rate.

Parameters:

- In body:
 - Rate - byte (required)
 - photo - long (required)

Responses:

- 200 OK
 - 400 Bad request
-

GET /api/v1/rates

Description:

Load list of user's rates.

Responses:

- 200 OK
 - rateID – long
 - user – long
 - photo – long
 - date – timestamp
 - rate - byte
 - 204 No content
-

PUT /api/v1/rates/{photoid}/{rate}

Description:

Edit rate.

Parameters:

- In path
 - photoid – long (required)
 - rate – byte (required)

Responses:

- 200 OK
- 400 Bad request

DELETE **/api/v1/rates/{photoid}**

Description:

Remove rate.

Parameters:

- **In path:**
 - **id – long** **(required)**

Responses:

- **200 OK**
- **400 Bad request**