Contents

[USERS 3](#_Toc515609930)

[**POST** **/api/v1/login** 3](#_Toc515609931)

[**POST** **/api/v1/account/logout** 3](#_Toc515609932)

[**POST** **/api/v1/users** 3](#_Toc515609933)

[**GET** **/api/v1/users** 4](#_Toc515609934)

[**GET** **/api/v1/users/{email}** 4](#_Toc515609935)

[**PUT** **/api/v1/users** 5](#_Toc515609936)

[**DELETE** **/api/v1/users** 5](#_Toc515609937)

[PHOTOS 6](#_Toc515609938)

[**POST**  **/api/v1/photos** 6](#_Toc515609939)

[**PUT** **/api/v1/photos/{id}** 6](#_Toc515609940)

[**DELETE** **/api/v1/photos/{id}** 7](#_Toc515609941)

[**GET** **/api/v1/photos/count/{state}** 7](#_Toc515609942)

[PHOTOS PUBLIC 8](#_Toc515609943)

[**GET** **/api/v1/photos/public/hot/{beg}/{end}** 8](#_Toc515609944)

[**GET** **/api/v1/photos/public/trending/{beg}/{end}** 8](#_Toc515609945)

[**GET** **/api/v1/photos/public/fresh/{beg}/{end}** 8](#_Toc515609946)

[**GET** **/api/v1/photos/public/tags/any/{tags}** 9](#_Toc515609947)

[**GET** **/api/v1/photos/public/tags/all/{tags}** 9](#_Toc515609948)

[PHOTOS PRIVATE 10](#_Toc515609949)

[**GET** **/api/v1/photos** 10](#_Toc515609950)

[**GET** **/api/v1/photos/id/{id}** 10](#_Toc515609951)

[**GET** **/api/v1/photos/name/{name}** 11](#_Toc515609952)

[**GET** **/api/v1/photos/shared** 11](#_Toc515609953)

[**GET** **/api/v1/photos /tags/any/{tags}** 12](#_Toc515609954)

[**GET** **/api/v1/photos /tags/all/{tags}** 12](#_Toc515609955)

[**GET** **/api/v1/photos /categories/any/{tags}** 13](#_Toc515609956)

[**GET** **/api/v1/photos / categories /all/{tags}** 13](#_Toc515609957)

[**GET** **/api/v1/photos /nocat** 14](#_Toc515609958)

[**GET** **/api/v1/photos /archived** 14](#_Toc515609959)

[IMAGES 15](#_Toc515609960)

[**POST** **/api/v1/images/{photoId}** 15](#_Toc515609961)

[**GET** **/api/v1/images/{photoid}** 15](#_Toc515609962)

[CATEGORIES 16](#_Toc515609963)

[**POST** **/api/v1/categories** 16](#_Toc515609964)

[**GET**  **/api/v1/categories/{parentid}** 16](#_Toc515609965)

[**GET** **/api/v1/categories** 17](#_Toc515609966)

[**PUT**  **/api/v1/categories/{id}** 17](#_Toc515609967)

[**DELETE** **/api/v1/categories/{id}** 17](#_Toc515609968)

[**POST** **/api/v1/category** 18](#_Toc515609969)

[**PUT**  **/api/v1/category/{photoId} /{categoryId}** 18](#_Toc515609970)

[**DELETE** **/api/v1/category/{photoid}** 18](#_Toc515609971)

[TAGS 19](#_Toc515609972)

[**POST** **/api/v1/tags** 19](#_Toc515609973)

[**POST** **/api/v1/tags/** 19](#_Toc515609974)

[**GET** **/api/v1/tags** 19](#_Toc515609975)

[**GET**  **/api/v1/tags/{letters}** 20](#_Toc515609976)

[**GET** **/api/v1/tags/public** 20](#_Toc515609977)

[**GET**  **/api/v1/tags/public/{letters}** 20](#_Toc515609978)

[**PUT** **/api/v1/tag/{id}/{name}** 21](#_Toc515609979)

[**DELETE** **/api/v1/tag/{id}** 21](#_Toc515609980)

[SHARES 22](#_Toc515609981)

[**POST** **/api/v1/shares** 22](#_Toc515609982)

[**DELETE** **/api/v1/shares/{id}** 22](#_Toc515609983)

[RATES 23](#_Toc515609984)

[**POST** **/api/v1/rates/{photoId}** 23](#_Toc515609985)

[**DELETE** **/api/v1/rates/{photoid}** 23](#_Toc515609986)

# USERS

## **POST /api/v1/login**

**Description:**

**User authorization.**

**Parameters:**

* **In body:**
  + **email – type string (required)**
  + **password – type string (required)**

**Responses:**

* **200 OK**
  + **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:**
  + **email – type string (required)**
  + **password – type string (required)**
  + **firstName – type string (required)**
  + **lastName – type string (required)**

**Responses:**

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

## **GET /api/v1/users**

**Description:**

**Load list of app users data.**

**Responses:**

* **200 OK**
  + **userID – type long**
  + **email – type string**
  + **firstName – type string**
  + **lastName – type string**
  + **password – type string**
  + **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:**
  + **email – type string (required)**

**Responses:**

* **200 OK**
  + **userID – type long**
  + **email – type string**
  + **firstName – type string**
  + **lastName – type string**
  + **password – type string**
  + **role – type string**

**OR**

* + **userID – type long**
  + **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.**

**Responses:**

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

# PHOTOS

## **POST /api/v1/photos**

**Description:**

**Create photo info.**

**Parameters:**

* **In body:**
  + **name – type string (required)**
  + **description – type string (not required – default NULL)**
  + **shareState – type string (not required – default “PRIVATE”)**
  + **photoState – type string (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)**

**Responses:**

* **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:**
  + **state – PUBLIC | PRIVATE (required)**

**Responses:**

* **200 OK**
  + **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.**

**Responses:**

* **200 OK**
  + **photoID – type long**
  + **name – type string**
  + **user – type string**
  + **uploadTime – type timestamp**
  + **description – type string**
  + **rate – type int**
  + **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:**
  + **tags – type string (required) example “ …/all/1,2,3… ”**

**Responses:**

* **200 OK**
  + **photoID – type long**
  + **name – type string**
  + **user – type string**
  + **uploadTime – type timestamp**
  + **description – type string**
  + **rate – type int**
  + **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**
  + **photoID – type long**
  + **name – type string**
  + **user – type string**
  + **uploadTime – type timestamp**
  + **description – type string**
  + **rate – type int**
  + **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)**

**Responses:**

* **200 OK**
  + **photoID – type long**
  + **name – type string**
  + **user – type string**
  + **uploadTime – type timestamp**
  + **description – type string**
  + **rate – type int**
  + **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**
  + **photoID – type long**
  + **name – type string**
  + **user – type string**
  + **uploadTime – type timestamp**
  + **description – type string**
  + **rate – type int**
  + **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.**

**Responses:**

* **200 OK**
  + **photoID – type long**
  + **name – type string**
  + **user – type string**
  + **uploadTime – type timestamp**
  + **description – type string**
  + **rate – type int**
  + **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:**
  + **tags – type string (required) example “ …/all|any/1,2,3… ”**

**Responses:**

* **200 OK**
  + **photoID – type long**
  + **name – type string**
  + **user – type string**
  + **uploadTime – type timestamp**
  + **description – type string**
  + **rate – type int**
  + **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:**
  + **categories – type string (required) example “ …/all|any/1,2,3… ”**

**Responses:**

* **200 OK**
  + **photoID – type long**
  + **name – type string**
  + **user – type string**
  + **uploadTime – type timestamp**
  + **description – type string**
  + **rate – type int**
  + **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.**

**Responses:**

* **200 OK**
  + **photoID – type long**
  + **name – type string**
  + **user – type string**
  + **uploadTime – type timestamp**
  + **description – type string**
  + **rate – type int**
  + **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)**

**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 – type string (required)**
  + **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)**

**Responses:**

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

## **GET /api/v1/categories**

**Description:**

**Load list of user’s root categories.**

**Responses:**

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

**--------------------------------------------------­----------------------------------------------------------------------------------------**

## **PUT /api/v1/categories/{id}**

**Description:**

**Edit category.**

**Parameters:**

* **In path**
  + **Id – type long (required)**
* **In body:**
  + **name – type string (required)**
  + **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)**

**Responses:**

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

## **POST /api/v1/category**

**Description:**

**Assigns category to the photo.**

**Parameters:**

* **In body:**
  + **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)**
  + **categoryId – 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)**

**Responses:**

* **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:**
  + - **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**
  + **user – type long**
  + **name – type string**
* **204 No content**

## **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**
  + **photo – type long**
  + **user – type long**
  + **name – type string**
* **204 No content**

**--------------------------------------------------­----------------------------------------------------------------------------------------**

## **GET /api/v1/tags/public**

**Description:**

**Load list of public tags.**

**Responses:**

* **200 OK**
  + **tagID – type long**
  + **photo – type long**
  + **user – type long**
  + **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)**

**Responses:**

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

**--------------------------------------------------­----------------------------------------------------------------------------------------**

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

**Description:**

**Edit tag.**

**Parameters:**

* **In path**
  + **id – type long (required)**
  + **name – type string (required)**

**Responses:**

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

**--------------------------------------------------­----------------------------------------------------------------------------------------**

## **DELETE /api/v1/tag/{id}**

**Description:**

**Remove tag.**

**Parameters:**

* **In path:**
  + **id – type long (required)**

**Responses:**

* **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:**
  + **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)**

**Responses:**

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

# RATES

## **POST /api/v1/rates/{photoId}**

**Description:**

**Create new rate.**

**Parameters:**

* **In path:**
  + **photoId – type long (required)**

**Responses:**

* **201 Created**
* **400 Bad request**

**--------------------------------------------------­----------------------------------------------------------------------------------------**

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

**Description:**

**Remove rate.**

**Parameters:**

* **In path:**
  + **photoId – type long (required)**

**Responses:**

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