**Все запросы отправляются с заголовком Content-Type: application/json, если в запросе не указан другой вариант.**

**Все передаваемые и получаемые параметры должны строго соответствовать тому, что описано ниже!**

**Регистрация reg.php**

URL: {service}/signup

Method: POST

Data params:

* first\_name - обязательное
* surname - обязательное
* phone - обязательное, уникальное, ровно 11 цифр, может быть с ведущими нулями
* password - обязательное

Success response:

* Code: 201 Created
* Content:
  + id - идентификатор созданного пользователя

Error Response:

* Code: 422 Unprocessable entity
* Content - объект, где ключи - это поля, которые не прошли валидацию, а их значения - текст ошибки валидации

**Авторизация auth.php**

URL: {service}/login

Method: POST

Data params:

* phone - обязательное
* password - обязательное

Success response

* Code: 200 OK
* Content:
  + token - сгенерированный токен для дальнейшего доступа к странице

Error Response (ошибка валидации):

* Code: 422 Unprocessable entity
* Content - объект, где ключи - это поля, которые не прошли валидацию, а их значения - текст ошибки валидации

Error Response (неверный логин или пароль):

* Code: 404 Not found
* Content:
  + login: “Incorrect login or password”

Все последующие запросы требуют авторизации с использованием Bearer-токена. Токен должен быть отправлен в заголовке Authorization.

При отправке запроса без заголовка авторизации сервер должен вернуть следующий ответ:

Error Response:

* Code: 403 Forbidden
* Content:
  + message: You need authorization

**Выход exit.php**

URL: {service}/logout

Method: POST

Success response

* Code: 200 OK

**Загрузка фотографии photo\_load.php**

URL: {service}/photo

Method: POST

Content-Type: **FormData**

Data params:

* photo – обязательное, файл с изображением, только jpg, jpeg или png.

Success response

* Code: 201 Created
* Content:
  + id – уникальный идентификатор фотографии
  + name – название фотографии, по умолчанию “Untitled”
  + url – **абсолютная** ссылка на фотографию (с http://)

Error Response:

* Code: 422 Unprocessable entity
* Content - объект, где ключи - это поля, которые не прошли валидацию, а их значения - текст ошибки валидации

**Изменение фотографии photo\_change.php**

URL: {service}/photo/{ID}

Method: POST

Data params:

* **\_method: обязательное поле, со значением “patch”, без кавычек**
* name – необязательное
* photo – необязательное, измененная фотография в формате base64

Success response

* Code: 200 OK
* Content:
  + id – уникальный идентификатор фотографии
  + name – название фотографии
  + url – **абсолютная** ссылка на фотографию (с http://)

Error Response (ошибка валидации):

* Code: 422 Unprocessable entity
* Content - объект, где ключи - это поля, которые не прошли валидацию, а их значения - текст ошибки валидации

Error Response (Ошибка доступа) - В случае, если пользователь пытается изменить не свою фотографию, ему возвращаются следующие параметры:

* Code: 403 Forbidden

**Получение фотографий unload\_photo.php**

URL: {service}/photo

Method: GET

Success response

* Code: 200 OK
* Content: массив объектов, где каждый объект имеет следующие свойства:
  + id
  + name
  + url
  + owner\_id – id пользователя, которому принадлежит фотография
  + users – массив с id пользователей, которые имеют доступ к этой фотографии

**Получение одной фотографии find\_photo.php**

URL: {service}/photo/{ID}

Method: GET

Success response

* Code: 200 OK
* Content:
  + id
  + name
  + url
  + owner\_id – id пользователя, которому принадлежит фотография
  + users – массив с id пользователей, которые имеют доступ к этой фотографии

**Удаление фотографии photo\_delete.php**

URL: {service}/photo/{ID}

Method: DELETE

Success response:

* Code: 204 Deleted

Error Response (Ошибка доступа) - В случае, если пользователь пытается изменить не свою фотографию, ему возвращаются следующие параметры:

* Code: 403 Forbidden

**Шаринг фотографий**

URL: {service}/user/{ID}/share

Method: POST

Data params:

* photos – массив с идентификаторами фотографий. В случае, если в массиве будет id фотографии которая уже была расшарена, то повторно она расшарена не будет.

Success response:

* Code: 201 Created
* Content:
  + existing\_photos - массив с идентификаторами фотографий, которые уже есть на сервере.

**Поиск пользователей**

URL: {service}/user

Method: GET

Query parameters:

* search – строка запроса, в которой указывается имя (или часть имени) и (или) фамилия (или часть фамилии) и (или) номер телефона (или часть номера телефона). Например: Иван Иванов 7951, И Иван 7.

Success response:

* Code: 200 OK
* Content - массив объектов, которые содержат идентификатор, имя, фамилию и телефон пользователя:
  + id
  + first\_name
  + surname
  + phone