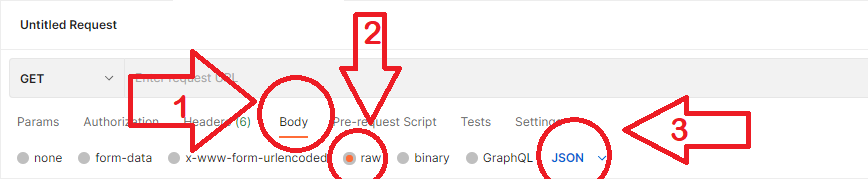
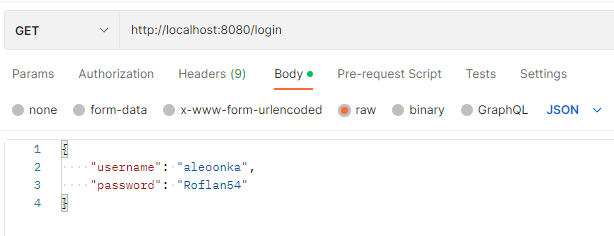
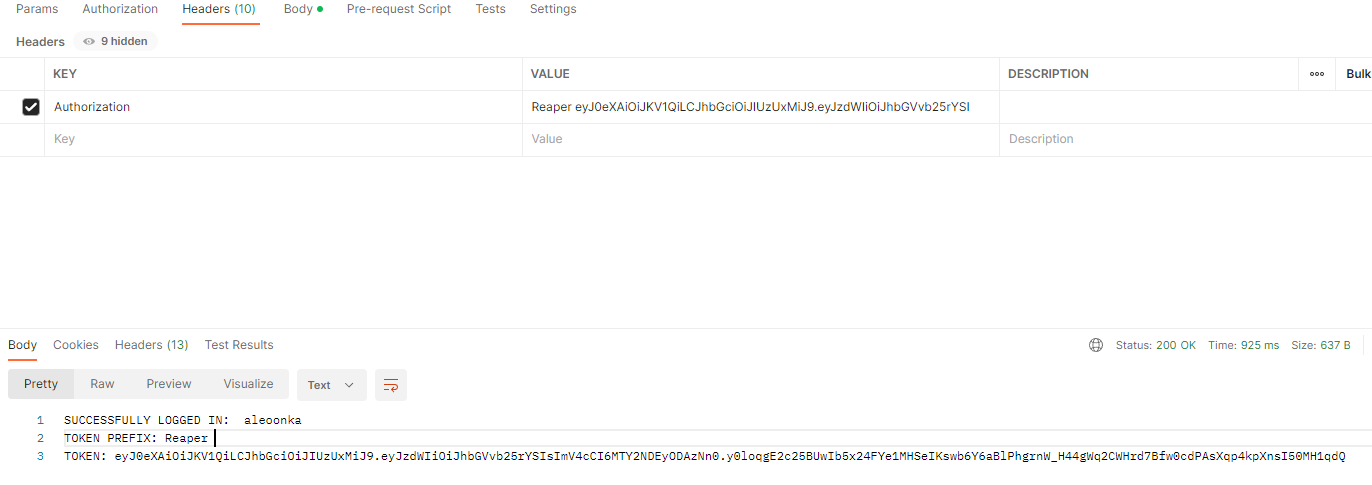
**Личная информация пользователя**

* Регистрация пользователя: /sign\_up, тип запроса – POST, получает на вход JSON информацию о пользователе, а также идентификатор роли (roleId равен 1 – админ или 2 – пользователь). Возвращает статус операции. JSON информация о пользователе состоит из:
  + username – логин пользователя (никнейм);
  + password – пароль;
  + email – адрес электронной почты;
  + firstName – имя;
  + lastName – фамилия;
  + genderType – пол;
  + dateOfBirth – дата рождения;
  + city – город;
  + openProfile – является ли профиль открытым;



* Вход в систему: /login, тип запроса GET, получает на вход JSON информацию о пользователе (username и пароль). Возвращает токен авторизации и его префикс. Токен и префикс необходимо поместить в Header с ключом Authorization.



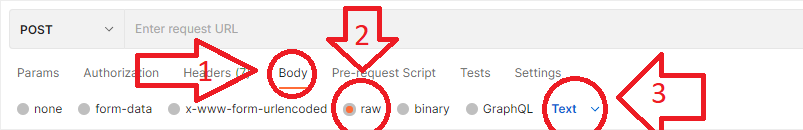


* Запрос на отправку письма с восстановлением пароля: /forgot\_password, тип запроса – PATCH, получает на вход никнейм пользователя (username). Возвращает статус данного действия.
* Запрос сброса пароля: /reset\_password/{код восстановления}, тип запроса – PATCH, получает на вход новый пароль (newPassword). Возвращает статус данного действия.
* Активация аккаунта: /activation/{код активации}, тип запроса – PATCH, (Вместо фигурных скобок вставляется код активации аккаунта). Возвращает статус данного действия.
* Обновление пользователя: /edit, тип запроса – PUT, получает на вход JSON информацию о пользователе (введите только необходимые параметры), а также текущий пароль пользователя (currentPassword). Возвращает обновленные данные, а также новый токен пользователя (только в случае изменения никнейма). Для выполнения данного действия должна быть выполнена авторизация уровня ‘WRITING’.
* Получение личной информации пользователя: /id{идентефикатор пользователя}, тип запроса – GET. Возвращает информацию о пользователе.
* Поиск пользователя: /search, тип запроса – GET, получает на вход JSON параметры для поиска (не обязательно искать сразу по всем параметрам), а возвращает список найденных пользователей. Параметры для поиска состоят из:
  + username – логин пользователя (никнейм);
  + email – адрес электронной почты;
  + firstName – имя;
  + lastName – фамилия;
  + genderType – пол;
  + fromDate – дата рождения, от которой искать;
  + toDate – дата рождения, до которой искать;
  + city – город;
* Бан (блокировка) пользователя: /ban, тип запроса – PATCH, получает на вход идентификатор (id) пользователя, которого необходимо забанить. Возвращает статус данного действия. Для выполнения данного действия должна быть выполнена авторизация уровня ‘ADMIN\_ACTIONS’.
* Разблокировка пользователя: /unban, тип запроса – PATCH, получает на вход идентификатор (id) пользователя, которого необходимо разбанить. Возвращает статус данного действия. Для выполнения данного действия должна быть выполнена авторизация уровня ‘ADMIN\_ACTIONS’.

**Взаимодействие с чатами**

Важно, что все действия в данной секции требуют авторизацию уровня ‘WRITING’, кроме некоторых, требующих прав админа.

* Получение информации о членстве пользователя в чатах: /chats, тип запроса – GET, возвращает список чатов, в которых состоит пользователь.
* Получение информации о чате, а также сообщений из него: /chat{идентификатор чата}, тип запроса – GET, возвращает всю информацию о чате, а также сообщения из него.
* Отправка сообщения в чат: /chat, тип запроса – POST, получает на вход либо идентификатор группового чата (chatId) или идентификатор пользователя (recipientId), для которого предназначено сообщение, а также текст самого сообщения (текст сообщения отправляется в качестве текстового тела запроса). Возвращает информацию о созданном сообщении.

****

* Проверка участия текущего пользователя в чате: /chat{ идентификатор чата}/participations, тип запроса – GET, возвращает информацию об участии текущего пользователя в чате.
* Создание группового чата: /public\_chat, тип запроса – POST, получает на вход идентификатор пользователя (userToAddId), которого добавят в чат при создании, а также имя чата (chatName). Возвращает информацию об участниках группового чата.
* Добавление пользователя в групповой чат: /public\_chat/add\_user, тип запроса – POST, получает на вход идентификатор чата (chatId), а также идентификатор пользователя, которого необходимо добавить (userToAddId). Возвращает информацию о добавлении пользователя в чат.
* Переименование чата: /chat{идентификатор чата}, тип запроса – PATCH, принимает на вход новое имя чата (chatName). Возвращает всю информацию о чате, а также сообщения из него.
* Удаление пользователя из чата: /public\_chat{идентификатор чата}/remove\_user, тип запроса – DELETE, принимает на вход идентификатор пользователя, которого необходимо удалить (userToRemoveId). Возвращает статус данного действия.
* Получение идентификатора владельца чата: /public\_chat{идентификатор чата}/owner, тип запроса – GET, возвращает информацию о владельце чата.
* Изменение владельца чата: /public\_chat{идентификатор чата}/owner, тип запроса – PATCH, получает на вход идентификатор нового владельца чата (newOwnerId). Возвращает статус данного действия.
* Зарос в поддержку: /support, тип запроса – POST, получает на вход текст запроса в поддержку (текстовое тело запроса), а возвращает информацию о созданном запросе в поддержку.
* Получение запросов в поддержку: /support\_requests, тип запроса – GET, возвращает информацию о полученных запросах в поддержку. Для выполнения данного действия должна быть выполнена авторизация уровня ‘ADMIN\_ACTIONS’.
* Ответ пользователю на запрос в поддержку: /support, тип запроса – DELETE, получает на вход идентификатор обращения в поддержку (messageId), а также ответ пользователю (текстовое тело запроса). Возвращает статус данного действия. Для выполнения данного действия должна быть выполнена авторизация уровня ‘ADMIN\_ACTIONS’.

**Управление списком друзей**

Важно, что все действия в данной секции требуют авторизацию уровня ‘WRITING’.

* Получение информации о друзьях текущего пользователя: /friends, тип запроса – GET, возвращает информацию о друзьях текущего пользователя.
* Получение списка друзей конкретного пользователя: /friends{идентификатор пользователя}/friends, тип запроса – GET, возвращает информацию о друзьях пользователя.
* Получение информации об отправленных запросах дружбы текущего пользователя: /friend\_requests/sent, тип запроса – GET, возвращает информацию об отправленных запросах дружбы текущего пользователя.
* Получение информации о полученных запросах дружбы текущего пользователя: /friend\_requests/received, тип запроса – GET, возвращает информацию о полученных запросах дружбы текущего пользователя.
* Отправка запроса дружбы: /friend, тип запроса – POST, получает на вход идентификатор пользователя (userId), которому будет послан запрос дружбы. Возвращает информацию о созданном запросе дружбы.
* Принять запрос дружбы: /friend\_requests/accept, тип запроса – PATCH, получает на вход идентификатор запроса дружбы (requestId). Возвращает информацию о данном запросе.
* Отклонить запрос дружбы: /friend\_requests/decline, тип запроса – DELETE, получает на вход идентификатор запроса дружбы (requestId). Возвращает статус данного действия.
* Удалить пользователя из друзей: /unfriend, тип запроса – DELETE, получает на вход идентификатор пользователя (userId), которого необходимо удалить из друзей. Возвращает статус данного действия.

**Взаимодействие с постами и группами**

Важно, что все действия в данной секции требуют авторизацию уровня ‘WRITING’.

* Получение информации о группе: /group{идентификатор группы}, тип запроса – GET, возвращает информацию о группе.
* Создание группы: /group, тип запроса – POST, получает на вход JSON информацию о группе, возвращает информацию о созданной группе. JSON информация о группе состоит из:
  + id – идентификатор группы;
  + name – имя группы;
  + info – информация о группе;
  + wallType – тип доступа к стене;
  + openToJoin – статус открытость группы;
  + wall – стена группы с постами (при создании она пустая);
  + groupMemberships – участники группы;
* Обновление информации о группе: /group{идентификатор группы}, тип запроса – PATCH, получает на вход обновленную JSON информацию о группе (редактировать можно только нужные параметры). Возвращает обновленную информацию о группе.
* Подписка на группу: /group{идентификатор группы}/subscribe, тип запроса – POST, возвращает информацию о совершенной подписке.
* Отписка от группы: group{идентификатор группы}/unsubscribe, тип запроса – DELETE, возвращает статус действия.
* Выдать роль подписчику: /group{идентификатор группы}/grant\_role, тип запроса – PATCH, получает на вход идентификатор подписчика (subscriberId), которому необходимо выдать роль, а также выдаваемую роль (groupRole на выбор из “OWNER”, “ADMIN”, “SUBSCRIBER”, “AWAITING\_CHECK”). Возвращает обновленную информацию о членстве в группе.
* Удаление группы: /group{идентификатор группы}, тип запроса – DELETE. Возвращает статус данного действия.
* Получение информации о запросах на вступление в группу: /group{идентификатор группы}/sub\_requests, тип запроса – GET. Возвращает информацию о полученных запросах на подписку.
* Принять запрос на вступление в группу: /group{идентификатор группы}/sub\_requests/accept, тип запроса – PATCH, принимает на в ход идентификатор запроса на вступление в группу. Возвращает информацию о созданном членстве в группе.
* Принять запрос на вступление в группу: /group{идентификатор группы}/sub\_requests/decline, тип запроса – DELETE, принимает на в ход идентификатор запроса на вступление в группу. Возвращает статус данного действия.
* Получение идентификатора владельца группы: /group{идентификатор группы}/owner, тип запроса – GET, возвращает информацию о членстве владельца группы.
* Получить информацию о постах в группе: /group{идентификатор чата}/wall, тип запроса – GET, возвращает информацию о постах в группе.
* Создать пост: /group{идентификатор группы}/wall, тип запроса – POST, принимает на вход текст поста (текстовое тело запроса).
* Получение поста из группы: /group{идентификатор группы}/wall/post, тип запроса – GET, получает на вход идентификатор поста (postId). Возвращает информацию о посте.
* Удаление поста: /group{идентификатор группы}/wall, тип запроса – DELETE, получает на вход идентификатор поста (postId). Возвращает информацию о посте.
* Получение постов из списка предложенных постов: /group{идентификатор группы}/wall/offered\_posts, тип запроса – GET. Возвращает посты, ожидающие проверки.
* Принять пост из списка предложенных постов: group{идентификатор группы}/wall/offered\_posts/accept, тип запроса – PATCH, получает на вход идентификатор поста (postId). Возвращает информацию о посте.
* Отклонить пост из списка предложенных постов: group{идентификатор группы}/wall/offered\_posts/decline, тип запроса – DELETE, получает на вход идентификатор поста (postId). Возвращает информацию о посте.
* Получение информации о лайках поста: /group{идентификатор группы}/wall/post{идентификатор поста}/likes, тип запроса – GET. Возвращает список поставленных посту лайков.
* Поставить лайк на пост: /group{идентификатор группы}/wall/post{идентификатор поста}/likes, тип запроса – POST. Возвращает информацию о поставленном лайке. Если второй раз подать такой же запрос лайк отменится.
* Получить комментарии: /group{идентификатор группы}/wall/{идентификатор поста}/comments, тип запроса – GET. Возвращает информацию о комментариях к посту.
* Написать комментарий: /group{идентификатор группы}/wall/post{идентификатор поста}/comments, тип запроса – POST, получает на вход текст комментария (текстовое тело запроса). Возвращает информацию о созданном комментарии.
* Удаление комментария: /group{идентификатор группы}/wall/post{идентификатор поста}/comments, тип запроса – DELETE, получает на вход идентификатор комментария (coomentId). Возвращает статус данного действия.
* Получение лайков на комментарии: /group{идентификатор группы}/wall/post{идентификатор поста}/comment{идентификатор комментария}/likes, тип запроса – GET. Возвращает список лайков на комментарии.
* Поставить лайк на комментарий: /group{идентификатор группы}/wall/post{идентификатор поста}/comment{идентификатор комментария}/likes, тип запроса – POST. Возвращает информацию о поставленном лайке. Если второй раз подать такой же запрос лайк отменится.
* Получить информацию о группах, в которых состоит пользователь: /id{идентификатор пользователя}/groups, тип запроса – GET. Возвращает список групп пользователя.

**Взаимодействие c личными постами пользователя**

Важно, что все действия в данной секции требуют авторизацию уровня ‘WRITING’.

* Получение постов текущего пользователя: /wall, тип запроса – GET. Возвращает информацию о постах на стене текущего пользователя.
* Написать пост на своей стене: /wall, тип запроса – POST, получает на вход текст поста (текстовое тело запроса).
* Изменить настройки доступа к стене пользователя: /wall/settings, тип запроса – PUT, получает на вход тип разрешения постов на стене (postPermission равный ALL, FRIENDS или OWNER) и тип разрешения на комментирование (commentPermission). Возвращает статус данного действия.
* Получить стену пользователя: /id{идентификатор пользователя}/wall, тип запроса – GET. Возвращает список постов со стены пользователя.
* Добавить пост на стену стену пользователя: /id{идентификатор пользователя}/wall, тип запроса – POST, получает на вход текст поста (текстовое тело запроса). Возвращает информацию о созданном посте.
* Получить пост со стены пользователя: /id{идентификатор пользователя}/wall/post, тип запроса – GET, получает на вход идентификатор поста (postId). Возвращает информацию о посте.
* Удаление поста со стены пользователя: /id{идентификатор пользователя}/wall/post, тип запроса – DELETE, получает на вход идентификатор поста, который необходимо удалить (postId). Возвращает статус действия.
* Получение информации о лайках поста: /id{идентификатор пользователя}/wall/post{идентификатор поста}/likes, тип запроса – GET. Возвращает список поставленных посту лайков.
* Поставить лайк на пост: /id{идентификатор пользователя}/wall/post{идентификатор поста}/likes, тип запроса – POST. Возвращает информацию о поставленном лайке. Если второй раз подать такой же запрос лайк отменится.
* Получить комментарии: /id{идентификатор пользователя}/wall/{идентификатор поста}/comments, тип запроса – GET. Возвращает информацию о комментариях к посту.
* Написать комментарий: /id{идентификатор пользователя}/wall/post{идентификатор поста}/comments, тип запроса – POST, получает на вход текст комментария (текстовое тело запроса). Возвращает информацию о созданном комментарии.
* Удаление комментария: /id{идентификатор пользователя}/wall/post{идентификатор поста}/comments, тип запроса – DELETE, получает на вход идентификатор комментария (coomentId). Возвращает статус данного действия.
* Получение лайков на комментарии: /id{идентификатор пользователя}/wall/post{идентификатор поста}/comment{идентификатор комментария}/likes, тип запроса – GET. Возвращает список лайков на комментарии.
* Поставить лайк на комментарий: /id{идентификатор пользователя}/wall/post{идентификатор поста}/comment{идентификатор комментария}/likes, тип запроса – POST. Возвращает информацию о поставленном лайке. Если второй раз подать такой же запрос лайк отменится.