

API Spec

2024-03-13

OneOnOne_Api

POST /auth/login/

Takes a set of user credentials and returns an access and refresh JSON web token pair to prove the authentication of those credentials.

Request Body:

Content: `application/json` | Type: `TokenObtainPair`

```
{
  username: string;
  password: string;
  access: string;
  refresh: string;
}
```

Content: `application/x-www-form-urlencoded` | Type: `TokenObtainPair`

```
{
  username: string;
  password: string;
  access: string;
  refresh: string;
}
```

Content: `multipart/form-data` | Type: `TokenObtainPair`

```
{
  username: string;
  password: string;
  access: string;
  refresh: string;
}
```

Response 200:

Content: `application/json` | Type: `TokenObtainPair`

```
{
  username: string;
  password: string;
  access: string;
  refresh: string;
}
```

POST /auth/refresh/

Takes a refresh type JSON web token and returns an access type JSON web token if the refresh token is valid.

Request Body:

Content: `application/json` | Type: `TokenRefresh`

```
{
  access: string;
  refresh: string;
}
```

Content: `application/x-www-form-urlencoded` | Type: `TokenRefresh`

```
{
```

```

    access: string;
    refresh: string;
  }
  Content: multipart/form-data | Type: TokenRefresh
  {
    access: string;
    refresh: string;
  }

```

Response 200:

```

  Content: application/json | Type: TokenRefresh
  {
    access: string;
    refresh: string;
  }

```

POST /auth/register/

Register a new user.

Request Body:

```

  Content: application/json | Type: User
  {
    username: string; // Required. 150 characters or fewer. Letters, digits and @/./+/_ only.
    email?: string;
    password: string;
    password2: string;
  }
  Content: application/x-www-form-urlencoded | Type: User
  {
    username: string; // Required. 150 characters or fewer. Letters, digits and @/./+/_ only.
    email?: string;
    password: string;
    password2: string;
  }
  Content: multipart/form-data | Type: User
  {
    username: string; // Required. 150 characters or fewer. Letters, digits and @/./+/_ only.
    email?: string;
    password: string;
    password2: string;
  }

```

Response 201:

User created successfully.

GET /calendars/

Request Parameters:

page?: integer; // A page number within the paginated result set.

Response 200:

Content: application/json | Type: PaginatedCalendarList

```
{
  count: integer;
  next: string;
  previous: string;
  results: Array<Calendar>;
}
```

POST /calendars/**Request Body:**

Content: application/json | Type: CalendarCreate

```
{
  id: integer;
  title: string;
  description: string;
  days: Array<Day>;
}
```

Content: application/x-www-form-urlencoded | Type: CalendarCreate

```
{
  id: integer;
  title: string;
  description: string;
  days: Array<Day>;
}
```

Content: multipart/form-data | Type: CalendarCreate

```
{
  id: integer;
  title: string;
  description: string;
  days: Array<Day>;
}
```

Response 201:

Content: application/json | Type: CalendarCreate

```
{
  id: integer;
  title: string;
  description: string;
  days: Array<Day>;
}
```

GET /calendars/{calendar_id}/day/{day_id}/timeslot/**Request Parameters:**

```
calendar_id: integer;
day_id: integer;
page?: integer; // A page number within the paginated result set.
```

Response 200:Content: `application/json` | Type: `PaginatedTimeSlotList`

```
{
  count: integer;
  next: string;
  previous: string;
  results: Array<TimeSlot>;
}
```

POST `/calendars/{calendar_id}/day/{day_id}/timeslot/`**Request Parameters:**

`calendar_id`: integer;
`day_id`: integer;

Request Body:Content: `application/json` | Type: `TimeSlot`

```
{
  id: integer;
  owner_username: string;
  start_time: string;
  end_time: string;
}
```

Content: `application/x-www-form-urlencoded` | Type: `TimeSlot`

```
{
  id: integer;
  owner_username: string;
  start_time: string;
  end_time: string;
}
```

Content: `multipart/form-data` | Type: `TimeSlot`

```
{
  id: integer;
  owner_username: string;
  start_time: string;
  end_time: string;
}
```

Response 201:Content: `application/json` | Type: `TimeSlot`

```
{
  id: integer;
  owner_username: string;
  start_time: string;
  end_time: string;
}
```

GET `/calendars/{calendar_id}/day/{day_id}/timeslot/{id}/`

Retrieve a time slot.

Request Parameters:

calendar_id: integer;
 day_id: integer;
 id: integer;

Response 200:

Content: application/json | Type: TimeSlot

```
{
  id: integer;
  owner_username: string;
  start_time: string;
  end_time: string;
}
```

PUT /calendars/{calendar_id}/day/{day_id}/timeslot/{id}/**Request Parameters:**

calendar_id: integer;
 day_id: integer;
 id: integer;

Request Body:

Content: application/json | Type: TimeSlot

```
{
  id: integer;
  owner_username: string;
  start_time: string;
  end_time: string;
}
```

Content: application/x-www-form-urlencoded | Type: TimeSlot

```
{
  id: integer;
  owner_username: string;
  start_time: string;
  end_time: string;
}
```

Content: multipart/form-data | Type: TimeSlot

```
{
  id: integer;
  owner_username: string;
  start_time: string;
  end_time: string;
}
```

Response 200:

Content: application/json | Type: TimeSlot

```
{
  id: integer;
  owner_username: string;
  start_time: string;
  end_time: string;
}
```

}

DELETE /calendars/{calendar_id}/day/{day_id}/timeslot/{id}/

Delete a time slot.

Request Parameters:

calendar_id: integer;
day_id: integer;
id: integer;

Response 204:

Content: application/json | Type: TimeSlot

```
{
  id: integer;
  owner_username: string;
  start_time: string;
  end_time: string;
}
```

GET /calendars/{calendar_id}/invitations/

Retrieve all invitations for a calendar.

Request Parameters:

calendar_id: integer;
page?: integer; // A page number within the paginated result set.

Response 200:

Content: application/json | Type: PaginatedInvitationList

```
{
  count: integer;
  next: string;
  previous: string;
  results: Array<Invitation>;
}
```

POST /calendars/{calendar_id}/invitations/

Create an invitation.

Request Parameters:

calendar_id: integer;

Request Body:

Content: application/json | Type: InvitationCreate

```
{
  id: integer;
  invitee_username: string;
}
```

```

}
Content: application/x-www-form-urlencoded | Type: InvitationCreate
{
  id: integer;
  invitee_username: string;
}
Content: multipart/form-data | Type: InvitationCreate
{
  id: integer;
  invitee_username: string;
}

```

Response 201:

```

Content: application/json | Type: Invitation
{
  id: integer;
  calendar_id: integer;
  invitee_username: string;
  inviter_username: string;
  status?: undefined;
}

```

GET /calendars/{calendar_id}/invitations/{id}/

Retrieve an invitation.

Request Parameters:

```

calendar_id: integer;
id: integer;

```

Response 200:

```

Content: application/json | Type: Invitation
{
  id: integer;
  calendar_id: integer;
  invitee_username: string;
  inviter_username: string;
  status?: undefined;
}

```

PUT /calendars/{calendar_id}/invitations/{id}/

Update an invitation.

Request Parameters:

```

calendar_id: integer;
id: integer;

```

Request Body:

```

Content: application/json | Type: InvitationEdit

```



```

{
  id: integer;
  status?: InvitationEditStatusEnum;
}
Content: application/x-www-form-urlencoded | Type: InvitationEdit

{
  id: integer;
  status?: InvitationEditStatusEnum;
}
Content: multipart/form-data | Type: InvitationEdit

{
  id: integer;
  status?: InvitationEditStatusEnum;
}

```

Response 204:

Content: application/json | Type: InvitationEdit

```

{
  id: integer;
  status?: InvitationEditStatusEnum;
}

```

DELETE /calendars/{calendar_id}/invitations/{id}/

Delete an invitation.

Request Parameters:

```

calendar_id: integer;
id: integer;

```

Response 204:

Content: application/json | Type: Invitation

```

{
  id: integer;
  calendar_id: integer;
  invitee_username: string;
  inviter_username: string;
  status?: undefined;
}

```

GET /calendars/{id}/**Request Parameters:**

```

id: integer;

```

Response 200:

Content: application/json | Type: Calendar

```

{
  id: integer;
}

```

```

creator_username: string;
title: string;
description: string;
days: Array<Day>;
participants?: Array<integer>;
}

```

PUT /calendars/{id}/

Request Parameters:

id: integer;

Request Body:

Content: application/json | Type: CalendarEdit

```

{
  id: integer;
  title: string;
  description: string;
  days: Array<Day>;
}

```

Content: application/x-www-form-urlencoded | Type: CalendarEdit

```

{
  id: integer;
  title: string;
  description: string;
  days: Array<Day>;
}

```

Content: multipart/form-data | Type: CalendarEdit

```

{
  id: integer;
  title: string;
  description: string;
  days: Array<Day>;
}

```

Response 200:

Content: application/json | Type: CalendarEdit

```

{
  id: integer;
  title: string;
  description: string;
  days: Array<Day>;
}

```

DELETE /calendars/{id}/

Request Parameters:

id: integer;

Response 204:

No response body.

GET /contacts/

Request Parameters:

page?: **integer**; // A page number within the paginated result set.

Response 200:

Content: **application/json** | Type: **PaginatedContactList**

```
{
  count: integer;
  next: string;
  previous: string;
  results: Array<Contact>;
}
```

POST /contacts/

Request Body:

Content: **application/json** | Type: **Contact**

```
{
  id: integer;
  userA: integer;
  userB: integer;
  status: undefined;
}
```

Content: **application/x-www-form-urlencoded** | Type: **Contact**

```
{
  id: integer;
  userA: integer;
  userB: integer;
  status: undefined;
}
```

Content: **multipart/form-data** | Type: **Contact**

```
{
  id: integer;
  userA: integer;
  userB: integer;
  status: undefined;
}
```

Response 201:

Content: **application/json** | Type: **Contact**

```
{
  id: integer;
  userA: integer;
  userB: integer;
  status: undefined;
}
```

GET /contacts/{id}/**Request Parameters:**

id: **integer**; // A unique integer value identifying this contact.

Response 200:

Content: **application/json** | Type: **Contact**

```
{
  id: integer;
  userA: integer;
  userB: integer;
  status: undefined;
}
```

PUT /contacts/{id}/**Request Parameters:**

id: **integer**; // A unique integer value identifying this contact.

Request Body:

Content: **application/json** | Type: **Contact**

```
{
  id: integer;
  userA: integer;
  userB: integer;
  status: undefined;
}
```

Content: **application/x-www-form-urlencoded** | Type: **Contact**

```
{
  id: integer;
  userA: integer;
  userB: integer;
  status: undefined;
}
```

Content: **multipart/form-data** | Type: **Contact**

```
{
  id: integer;
  userA: integer;
  userB: integer;
  status: undefined;
}
```

Response 200:

Content: **application/json** | Type: **Contact**

```
{
  id: integer;
  userA: integer;
  userB: integer;
  status: undefined;
}
```

PATCH /contacts/{id}/**Request Parameters:**

id: **integer**; // A unique integer value identifying this contact.

Request Body:

Content: **application/json** | Type: **PatchedContact**

```
{
  id: integer;
  userA: integer;
  userB: integer;
  status: undefined;
}
```

Content: **application/x-www-form-urlencoded** | Type: **PatchedContact**

```
{
  id: integer;
  userA: integer;
  userB: integer;
  status: undefined;
}
```

Content: **multipart/form-data** | Type: **PatchedContact**

```
{
  id: integer;
  userA: integer;
  userB: integer;
  status: undefined;
}
```

Response 200:

Content: **application/json** | Type: **Contact**

```
{
  id: integer;
  userA: integer;
  userB: integer;
  status: undefined;
}
```

DELETE /contacts/{id}/**Request Parameters:**

id: **integer**; // A unique integer value identifying this contact.

Response 204:

No response body.

POST /contacts/accept/

Accept an incoming pending friend request from another user.

Request Body:Content: `application/json` | Type: `Contact`

```
{
  id: integer;
  userA: integer;
  userB: integer;
  status: undefined;
}
```

Content: `application/x-www-form-urlencoded` | Type: `Contact`

```
{
  id: integer;
  userA: integer;
  userB: integer;
  status: undefined;
}
```

Content: `multipart/form-data` | Type: `Contact`

```
{
  id: integer;
  userA: integer;
  userB: integer;
  status: undefined;
}
```

Response 200:Content: `application/json` | Type: `Contact`

```
{
  id: integer;
  userA: integer;
  userB: integer;
  status: undefined;
}
```

POST /contacts/add/

Add a new contact to the user's contact list. The user to be added is specified by their username.
The user cannot add themselves.

If the contact already exists, the request will be rejected.

If the contact is blocked, the request will be rejected.

If the user is not found, the request will be rejected.

If the user is found, the request will be accepted and the contact will be added to the user's contact list with a status of pending.

Request Body:Content: `application/json` | Type: `Contact`

```
{
  id: integer;
  userA: integer;
  userB: integer;
  status: undefined;
}
```

Content: `application/x-www-form-urlencoded` | Type: `Contact`

```
{
  id: integer;
```

```

    userA: integer;
    userB: integer;
    status: undefined;
  }
  Content: multipart/form-data | Type: Contact
  {
    id: integer;
    userA: integer;
    userB: integer;
    status: undefined;
  }

```

Response 200:

```

  Content: application/json | Type: Contact
  {
    id: integer;
    userA: integer;
    userB: integer;
    status: undefined;
  }

```

POST /contacts/block/

Current user blocks another user.

Request Body:

```

  Content: application/json | Type: Contact
  {
    id: integer;
    userA: integer;
    userB: integer;
    status: undefined;
  }
  Content: application/x-www-form-urlencoded | Type: Contact
  {
    id: integer;
    userA: integer;
    userB: integer;
    status: undefined;
  }
  Content: multipart/form-data | Type: Contact
  {
    id: integer;
    userA: integer;
    userB: integer;
    status: undefined;
  }

```

Response 200:

```

  Content: application/json | Type: Contact
  {
    id: integer;
    userA: integer;
  }

```

```

    userB: integer;
    status: undefined;
  }

```

GET /contacts/friends/

Get a list of all friends of the current user.

Response 200:

Content: `application/json` | Type: `Contact`

```

{
  id: integer;
  userA: integer;
  userB: integer;
  status: undefined;
}

```

GET /contacts/incoming/

Get a list of all incoming pending friend requests for the current user.

Response 200:

Content: `application/json` | Type: `Contact`

```

{
  id: integer;
  userA: integer;
  userB: integer;
  status: undefined;
}

```

GET /contacts/outgoing/

Get a list of all outgoing pending friend requests for the current user.

Response 200:

Content: `application/json` | Type: `Contact`

```

{
  id: integer;
  userA: integer;
  userB: integer;
  status: undefined;
}

```

POST /contacts/reject/

Reject an incoming pending friend request from another user.

Request Body:

Content: `application/json` | Type: `Contact`

```
{
  id: integer;
  userA: integer;
  userB: integer;
  status: undefined;
}
```

Content: `application/x-www-form-urlencoded` | Type: `Contact`

```
{
  id: integer;
  userA: integer;
  userB: integer;
  status: undefined;
}
```

Content: `multipart/form-data` | Type: `Contact`

```
{
  id: integer;
  userA: integer;
  userB: integer;
  status: undefined;
}
```

Response 200:

Content: `application/json` | Type: `Contact`

```
{
  id: integer;
  userA: integer;
  userB: integer;
  status: undefined;
}
```

GET `/contacts/search/`

Get list of all people other than the current user.

Response 200:

Content: `application/json` | Type: `Contact`

```
{
  id: integer;
  userA: integer;
  userB: integer;
  status: undefined;
}
```

POST `/contacts/unadd/`

Current user unadds an existing blocked user.

Request Body:

Content: `application/json` | Type: `Contact`

```
{
```

```

    id: integer;
    userA: integer;
    userB: integer;
    status: undefined;
  }
Content: application/x-www-form-urlencoded | Type: Contact
{
  id: integer;
  userA: integer;
  userB: integer;
  status: undefined;
}
Content: multipart/form-data | Type: Contact
{
  id: integer;
  userA: integer;
  userB: integer;
  status: undefined;
}

```

Response 200:

```

Content: application/json | Type: Contact
{
  id: integer;
  userA: integer;
  userB: integer;
  status: undefined;
}

```

POST /contacts/unblock/

Current user unblocks an existing blocked user.

Request Body:

```

Content: application/json | Type: Contact
{
  id: integer;
  userA: integer;
  userB: integer;
  status: undefined;
}
Content: application/x-www-form-urlencoded | Type: Contact
{
  id: integer;
  userA: integer;
  userB: integer;
  status: undefined;
}
Content: multipart/form-data | Type: Contact
{
  id: integer;
  userA: integer;
  userB: integer;
}

```

```
    status: undefined;  
  }
```

Response 200:

Content: application/json | Type: Contact

```
{  
  id: integer;  
  userA: integer;  
  userB: integer;  
  status: undefined;  
}
```

Schemas

Calendar

```
{
  id: integer;
  creator_username: string;
  title: string;
  description: string;
  days: Array<Day>;
  participants?: Array<integer>;
}
```

CalendarCreate

```
{
  id: integer;
  title: string;
  description: string;
  days: Array<Day>;
}
```

CalendarEdit

```
{
  id: integer;
  title: string;
  description: string;
  days: Array<Day>;
}
```

Contact

```
{
  id: integer;
  userA: integer;
  userB: integer;
  status: undefined;
}
```

ContactStatusEnum

Type: integer

Values: 1, 2, 3, 4

Day

```
{
  id: integer;
  date: string;
  ranking: integer;
  timeslots: Array<TimeSlot>;
}
```

Invitation

```
{
  id: integer;
  calendar_id: integer;
  invitee_username: string;
  inviter_username: string;
  status?: undefined;
}
```

InvitationCreate

```
{
  id: integer;
  invitee_username: string;
}
```

InvitationEdit

```
{
  id: integer;
  status?: InvitationEditStatusEnum;
}
```

InvitationEditStatusEnum

Type: string

Values: pending, accepted, rejected

InvitationStatusEnum

Type: string

Values: pending, accepted, rejected

PaginatedCalendarList

```
{
  count: integer;
  next: string;
  previous: string;
  results: Array<Calendar>;
}
```

PaginatedContactList

```
{
  count: integer;
  next: string;
  previous: string;
  results: Array<Contact>;
}
```

PaginatedInvitationList

```
{  
  count: integer;  
  next: string;  
  previous: string;  
  results: Array<Invitation>;  
}
```

PaginatedTimeSlotList

```
{  
  count: integer;  
  next: string;  
  previous: string;  
  results: Array<TimeSlot>;  
}
```

PatchedContact

```
{  
  id: integer;  
  userA: integer;  
  userB: integer;  
  status: undefined;  
}
```

TimeSlot

```
{  
  id: integer;  
  owner_username: string;  
  start_time: string;  
  end_time: string;  
}
```

TokenObtainPair

```
{  
  username: string;  
  password: string;  
  access: string;  
  refresh: string;  
}
```

TokenRefresh

```
{  
  access: string;  
  refresh: string;  
}
```

User

```
{  
  username: string; // Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.  
  email?: string;  
  password: string;  
  password2: string;  
}
```