

# Assistance Platform API Documentation

## Table of Contents

- [Overview](#)
- [Authentication](#)
- [REST API Endpoints](#)
  - [Authentication Endpoints](#)
  - [Profile Endpoints](#)
  - [Assistance Endpoints](#)
- [WebSocket API](#)
  - [Assistance WebSocket](#)
  - [Assistant WebSocket](#)
- [Data Models](#)

## Overview

This API provides functionality for an assistance platform where clients can request assistance and assistants can accept and fulfill these requests. The platform includes user authentication, profile management, real-time location tracking, and assistance status updates.

## Authentication

The API uses JWT (JSON Web Token) for authentication. Most endpoints require an Authorization header with a valid token.

### Token Format

```
Authorization: Bearer <access_token>
```

### Token Refresh

When an access token expires, use the refresh endpoint to get a new access token.

## REST API Endpoints

### Authentication Endpoints

#### Register User

- **URL:** `/api/auth/register/`
- **Method:** `POST`

- **Authentication:** No
- **Description:** Register a new user (client or assistant)
- **Request Body:**

json

```
{
  "email": "user@example.com",
  "phone": "+1234567890",
  "password": "securepassword",
  "name": "User Name",
  "user_type": "client" // or "assistant"
}
```

- **Response:**

json

```
{
  "user": {
    "id": 1,
    "email": "user@example.com",
    "name": "User Name",
    "user_type": "client"
  },
  "tokens": {
    "access": "access_token_string",
    "refresh": "refresh_token_string"
  }
}
```

## Email Login

- **URL:** `/api/auth/login/email/`
- **Method:** `POST`
- **Authentication:** No
- **Description:** Log in with email and password
- **Request Body:**

json

```
{
  "email": "user@example.com",
  "password": "securepassword"
}
```

- **Response:**

json

```
{
  "user": {
    "id": 1,
    "email": "user@example.com",
    "name": "User Name",
    "user_type": "client"
  },
  "tokens": {
    "access": "access_token_string",
    "refresh": "refresh_token_string"
  }
}
```

## Phone Login

- **URL:** `/api/auth/login/phone/`
- **Method:** `POST`
- **Authentication:** No
- **Description:** Log in with phone number and password
- **Request Body:**

json

```
{
  "phone": "+1234567890",
  "password": "securepassword"
}
```

- **Response:** Same as email login

## Google Login

- **URL:** `/api/auth/login/google/`
- **Method:** `POST`
- **Authentication:** No
- **Description:** Log in with Google OAuth
- **Request Body:**

json

```
{
  "token": "google_oauth_token"
}
```

- **Response:** Same as email login

## Email Verification

- **URL:** `/api/auth/verify-email/`
- **Method:** `POST`
- **Authentication:** No
- **Description:** Verify user's email address
- **Request Body:**

json

```
{  
  "email": "user@example.com",  
  "code": "verification_code"  
}
```

## Phone Verification

- **URL:** `/api/auth/verify-phone/`
- **Method:** `POST`
- **Authentication:** No
- **Description:** Verify user's phone number
- **Request Body:**

json

```
{  
  "phone": "+1234567890",  
  "code": "verification_code"  
}
```

## Password Reset Request

- **URL:** `/api/auth/password-reset/`
- **Method:** `POST`
- **Authentication:** No
- **Description:** Request a password reset
- **Request Body:**

json

```
{  
  "email": "user@example.com"  
}
```

## Password Reset Confirm

- **URL:** `/api/auth/password-reset/confirm/`
- **Method:** `POST`
- **Authentication:** No
- **Description:** Confirm password reset with code
- **Request Body:**

json

```
{  
  "email": "user@example.com",  
  "code": "reset_code",  
  "new_password": "new_secure_password"  
}
```

## Resend Verification

- **URL:** `/api/auth/resend-verification/`
- **Method:** `POST`
- **Authentication:** No
- **Description:** Resend verification code
- **Request Body:**

json

```
{  
  "email": "user@example.com"  
  // or  
  "phone": "+1234567890"  
}
```

## Token Refresh

- **URL:** `/api/auth/token/refresh/`
- **Method:** `POST`
- **Authentication:** No

- **Description:** Get a new access token using a refresh token

- **Request Body:**

```
json
{
  "refresh": "refresh_token_string"
}
```

- **Response:**

```
json
{
  "access": "new_access_token_string"
}
```

## Profile Endpoints

### User Profile

- **URL:** `/api/profile/user/`
- **Method:** `GET`
- **Authentication:** Required
- **Description:** Get user profile information
- **Response:**

```
json
{
  "id": 1,
  "email": "user@example.com",
  "phone": "+1234567890",
  "name": "User Name",
  "user_type": "client",
  "current_lat": 40.7128,
  "current_lng": -74.0060
}
```

- **Method:** `PUT`/`PATCH`
- **Description:** Update user profile
- **Request Body:**

json

```
{
  "name": "Updated Name",
  "email": "updated@example.com",
  "phone": "+1987654321"
}
```

## Assistant Profile

- **URL:** `/api/profile/assistant/`
- **Method:** `GET`
- **Authentication:** Required (must be assistant)
- **Description:** Get assistant profile information
- **Response:**

json

```
{
  "id": 2,
  "email": "assistant@example.com",
  "phone": "+1234567890",
  "name": "Assistant Name",
  "user_type": "assistant",
  "current_lat": 40.7128,
  "current_lng": -74.0060,
  "is_active_assistant": true
}
```

- **Method:** `PUT`/`PATCH`
- **Description:** Update assistant profile
- **Request Body:**

json

```
{
  "name": "Updated Assistant Name",
  "email": "updated_assistant@example.com",
  "phone": "+1987654321"
}
```

## Update Location

- **URL:** `/api/profile/update-location/`
- **Method:** `POST`

- **Authentication:** Required
- **Description:** Update user's current location
- **Request Body:**

```
json
{
  "lat": 40.7128,
  "lng": -74.0060
}
```

## Assistant Status

- **URL:** `/api/profile/assistant-status/`
- **Method:** `POST`
- **Authentication:** Required (must be assistant)
- **Description:** Update assistant active status
- **Request Body:**

```
json
{
  "is_active": true
}
```

## Assistant List

- **URL:** `/api/assistants/`
- **Method:** `GET`
- **Authentication:** Required
- **Description:** Get list of active assistants
- **Response:**

```
json
[
  {
    "id": 2,
    "name": "Assistant Name",
    "current_lat": 40.7128,
    "current_lng": -74.0060,
    "distance": 1.5 // km from requesting user
  }
]
```



# Assistance Endpoints

## Request Assistance

- **URL:** `/api/assistance/request/`
- **Method:** `POST`
- **Authentication:** Required (must be client)
- **Description:** Request assistance
- **Request Body:**

json

```
{
  "pickup": {
    "lat": 40.7128,
    "lng": -74.0060
  },
  "dropoff": { // Optional
    "lat": 40.7308,
    "lng": -73.9973
  }
}
```

- **Response:**

json

```
{
  "assistance": {
    "id": 1,
    "client": {
      "id": 1,
      "name": "Client Name"
    },
    "assistant": null,
    "pickup_lat": 40.7128,
    "pickup_lng": -74.0060,
    "dropoff_lat": 40.7308,
    "dropoff_lng": -73.9973,
    "status": "requested",
    "created_at": "2025-05-02T10:00:00Z"
  },
  "nearby_assistants": [
    {
      "assistant_id": 2,
      "distance": 1.5
    }
  ]
}
```

## View Assistance

- **URL:** `/api/assistance/view/`
- **Method:** `GET`
- **Authentication:** Required
- **Description:** Get list of user's assistances (as client or assistant)
- **Response:**

json

```
{
  "message": "success",
  "Assistance": [
    {
      "id": 1,
      "client": {
        "id": 1,
        "name": "Client Name"
      },
      "assistant": {
        "id": 2,
        "name": "Assistant Name"
      },
      "pickup_lat": 40.7128,
      "pickup_lng": -74.0060,
      "dropoff_lat": 40.7308,
      "dropoff_lng": -73.9973,
      "status": "requested",
      "created_at": "2025-05-02T10:00:00Z"
    }
  ]
}
```

## View Specific Assistance

- **URL:** `/api/assistance/view/{assistance_id}/`
- **Method:** `GET`
- **Authentication:** Required
- **Description:** Get details of a specific assistance
- **Response:** Same as above but for a single assistance

## Accept Assistance

- **URL:** `/api/assistance/accept/{assistance_id}/`
- **Method:** `PATCH`
- **Authentication:** Required (must be assistant)
- **Description:** Accept an assistance request
- **Response:**

json

```
{
  "id": 1,
  "client": {
    "id": 1,
    "name": "Client Name"
  },
  "assistant": {
    "id": 2,
    "name": "Assistant Name"
  },
  "pickup_lat": 40.7128,
  "pickup_lng": -74.0060,
  "dropoff_lat": 40.7308,
  "dropoff_lng": -73.9973,
  "status": "accepted",
  "created_at": "2025-05-02T10:00:00Z"
}
```

## Update Assistance Status

- **URL:** `/api/assistance/update/{assistance_id}/`
- **Method:** `PATCH`
- **Authentication:** Required (must be client or assistant)
- **Description:** Update assistance status
- **Request Body:**

json

```
{
  "status": "ongoing" // Options: ongoing, completed, canceled
}
```

- **Response:**

json

```
{
  "id": 1,
  "client": {
    "id": 1,
    "name": "Client Name"
  },
  "assistant": {
    "id": 2,
    "name": "Assistant Name"
  },
  "pickup_lat": 40.7128,
  "pickup_lng": -74.0060,
  "dropoff_lat": 40.7308,
  "dropoff_lng": -73.9973,
  "status": "ongoing",
  "created_at": "2025-05-02T10:00:00Z"
}
```

## WebSocket API

The application uses WebSockets for real-time communication between clients and assistants.

### Assistance WebSocket

Connect to this WebSocket to receive real-time updates about a specific assistance.

- **URL:** `ws://your-domain/ws/assistance/{assistance_id}/`

### Messages from Server

#### 1. Connection Established

json

```
{
  "type": "connection_established",
  "message": "Connected to channel_1"
}
```

#### 2. Status Update

json

```
{
  "type": "status",
  "status": "accepted",
  "assistant": {
    "id": 2,
    "name": "Assistant Name"
  },
  "client": {
    "id": 1,
    "name": "Client Name"
  }
}
```

### 3. Location Update

json

```
{
  "type": "location",
  "lat": 40.7128,
  "lng": -74.0060,
  "user_id": 2
}
```

### 4. Chat Message

json

```
{
  "type": "chat",
  "message": "I'll be there in 5 minutes",
  "sender": 2
}
```

### 5. Error

json

```
{
  "type": "error",
  "message": "Error message"
}
```

## Messages to Server

### 1. Location Update

json

```
{
  "type": "location_update",
  "user_id": 2,
  "lat": 40.7128,
  "lng": -74.0060
}
```

## Assistant WebSocket

Connect to this WebSocket to receive real-time assistance requests as an assistant.

- **URL:** `ws://your-domain/ws/assistant/{assistant_id}/`

## Messages from Server

### 1. Connection Established

json

```
{
  "type": "connection_established",
  "message": "Connected as assistant_2"
}
```

### 2. New Assistance Request

json

```
{
  "type": "new_assistance_request",
  "assistance_id": 1,
  "pickup_location": {
    "lat": 40.7128,
    "lng": -74.0060
  },
  "client": {
    "id": 1,
    "name": "Client Name"
  }
}
```

## Data Models

### User

- `id`: Integer
- `email`: String

- `phone`: String
- `name`: String
- `user_type`: String ('client' or 'assistant')
- `current_lat`: Float
- `current_lng`: Float
- `is_active_assistant`: Boolean (for assistants only)

## Assistance

- `id`: Integer
- `client_id`: Integer (foreign key to User)
- `assistant_id`: Integer (foreign key to User, nullable)
- `pickup_lat`: Float
- `pickup_lng`: Float
- `dropoff_lat`: Float (optional)
- `dropoff_lng`: Float (optional)
- `status`: String ('requested', 'accepted', 'ongoing', 'completed', 'canceled')
- `created_at`: DateTime