# Health Connect REST API Documentation

## Introduction

The Next app API is a RESTful API built using Django REST Framework. It serves as the backend for Next app providing a scalable and modular system for handling data, authentication, and business logic.

## Technologies Used

* Python
* Django
* Django REST API (DRF)
* Message Central OTP verification
* Razor pay payment gateway
* Postgres DB

## Authentication Flow

The API implements token-based authentication with OTP verification for both users and partners.

### Authentication Process

* Registration or login request (phone number provided)
* OTP sent to user/partner phone
* OTP verification
* Token issuance (access + refresh tokens)
* Token usage for API access

## API Routes

## Auth Routes (Partner and User)

## Registration end points

## auth/register/user/

## This route is used to register a user where it will take phone number, email and full name to give out OTP and a verification id(part of OTP verification) which needs to be used in the route auth/verify/user. There is max retry count of 3 every minute. Full name and email is stored in cache so it can be accessed by auth/verify/user to add to db.

## auth/verify/user/

## This route is used to verify the user’s OTP and verification ID and they are verified by Message central service If verification is ok then email and full name is taken from cache and a user object is created then the cache is cleared. The user will receive their appropriate JWT access and refresh tokens.

## auth/register/partner/

## This route is used to register a partner where it will take phone number, email, education. Experience, medical certificate and full name to give out OTP and a verification id(part of OTP verification) which needs to be used in the route auth/verify/user. There is max retry count of 3 every minute. Full name and email and others are stored in cache so it can be accessed by auth/verify/user to add to db.

## auth/verify/partner/

## This route is used to verify the partner’s OTP and verification ID and they are verified by Message central service If verification is ok then email and full name and others are taken from cache and a user and its partner object is created then the cache is cleared. If a already registered user registers as partner then the user’s partner object is created. The user will receive their appropriate JWT access and refresh tokens for partner views.

## Login end points

## auth/login/user/

## This route is used for sending OTP and verification id so they can login to the app. Phone number is taken here. Max retry of 3 is set and phone number and verification is stored in cache so it can be accessed by auth/verify/login/user to verify and login.

## auth/verify/login/user/

## This route is used to verify the user’s OTP and verification ID and they are verified by Message central service and user will receive their appropriate JWT access and refresh tokens.

## auth/login/partner/

## This route is used for sending OTP and verification id so they can login to the app. Phone number is taken here. Max retry of 3 is set and phone number and verification is stored in cache so it can be accessed by auth/verify/login/user to verify and login.

## auth/verify/login/partner/

## This route is used to verify the partner’s OTP and verification ID and they are verified by Message central service and user will receive their appropriate JWT access and refresh tokens.

## JWT Token end points

## auth/token/refresh/

## This route is used to create access tokens for auth by providing refresh if expired. This app needs access token in header to access each protected routes.

## auth/token/verify/

## This route is used to check if a access token has expired so the app can use auth/token/refresh to create another from refresh token

## auth/logout/

## This route is used to logout for both user and partner. In context of JWT logging out means abandoning both refresh and access by the API consumer (web-app , mobile app etc) if stored so the user can’t use it anymore to login. Here a blacklisting mechanism is implemented where if logged out the old refresh token is stored in blacklisted table in database so it can’t be used anymore.

## User Routes

## User end points

## user/home/

## Home page route for user to access after logging in. This route returns user details.

## user/services/

## This route gives out the available services details which are Hospital care , check-up companion, adult care, baby sitting. This route also gives hourly rate of the above services.

## user/bookings/history/

## This route gives out booking history of users with status either completed or cancelled.

## user/bookings/<int:booking\_id>/

## This route gives out detailed view of a specific booking.

## user/bookings/create/

## This route is used to create a booking for either book now or book later options.This route has validations for avoiding same day bookings if another booking exists and time overlap for book laters.

## Instant booking is where partners can accept work and start work immediately.

## {

## "service\_type": 1,

## "partner\_type": "trained",

## "is\_instant": true,

## "hours": 4,

## "user\_location": "123 Main Street, Anytown",

## "hospital\_location": "General Hospital, Anytown",

## "notes": "Bring n95 Mask",

## "long": -74.0060,

## "lang": 40.7128

## }

## Booking later is where partners can accept work and start work at given time and date.

## {

## "service\_type": 2,

## "partner\_type": "normal",

## "is\_instant": false,

## "scheduled\_date": "2025-04-30",

## "scheduled\_time": "14:30",

## "hours": 3,

## "user\_location": "456 Elm Street, Anytown",

## "hospital\_location": "City Health Clinic, Anytown",

## "notes": "Patient needs wheelchair assistance",

## "long": -74.0059,

## "lang": 40.7130

## }

## In Service type checkup companion there needs to give a additional field called hospital location.

## {

## "service\_type": 3,

## "partner\_type": "trained",

## "is\_instant": false,

## "scheduled\_date": "2025-05-02",

## "scheduled\_time": "10:00",

## "hours": 2,

## "user\_location": "789 Oak Avenue, Anytown",

## "hospital\_location": "Downtown Medical Center, Anytown",

## "notes": "Elderly patient, please assist with forms",

## "long": -74.0070,

## "lang": 40.7110

## }

## Data Models

[Key data structures and their relationships]

## Error Handling

[Standard error responses and codes]

## Rate Limiting

[Information about API rate limits]

## Versioning

[API versioning information]