**APNESATHI BACKEND DOCUMENTATION FOR MOBILE APP**

In the backend we have created 14 APIS for the respective functionalities in the mobile app.

Please find the API list below which we have created till date.

* **/seniorcitizenDetails**

In this API we will pass callid in the request body to fetch the details of Senior citizen

* **/VolunteerorAdminData**

In this API we will pass phoneNo in the request body to fetch the details of Volunteer or Admin

* **/saveForm**

In this API we will pass VolunteerAssignment object in the request body to save the details of call and grievances raised by particular senior citizen

* **/loadDashboard**

In this API we will pass id and filterBy in the request body to get the details of volunteer or admin and their corresponding calls, grievances, grievance status details

filterBy can be any of the following

1- volunteer

2- staff member

4- master admin

Since district admin doesn’t have dashboard so we are not using 3 for filter by value

* **/registerNewSrCitizen**

**In this API we will pass senior citizen details along with role and call statuscode to register new senior citizen**

**Callstatuscode will be 1**

**Role can be any of the following integer**

1- volunteer

2- staff member

4- master admin

Since district admin doesn’t have permission to register new senior citizen so we are not using 3 for role value

* **/getGreivanceDetails**

In this API we will pass id , filterBy or id ,districtId to get all grievance details.

Id can be either volunteer or admin id, filterby is the role as mentioned below, districtId is the id of the district to which district admin or master admin assigned to

**Volunteer - 1**

**Staff Member - 2**

**District Admin - 3**

**Master Admin - 4**

* **/getSrCitizenDetails**

In this API we will pass idvolunteer and loggeddateTime to fetch all senior citizen details from DB <=loggeddateTime

* **/updateGreivanceDetails**

In this API we will pass trackingId,status,description, grievance type, reviewedby or resolvedby or raisedby based on the status to which we are updating

* **/updateProfile**

In this API we will pass adminId or idvolunteer for whom we need to update address or firstName or LastName or email

* **/verifyPassword**

This API is applicable only for admins where we will pass adminId and password in request body and we convert the password to encrypted format and compare it with password stored in DB.If it matches we will send success response else we will send failure response

* **/getVolunteerList**

In this API we will pass adminId in the request body to fetch all volunteer details assigned to this admin from DB

* **/loginVolunteerOrAdmin**

In this api we pass phoneNo in the request body to validate whether the user exists in DB or not.

* **/volunteerRating**

In this API we will pass idvolunteer, adminId,First name of volunteer and admin,rating given by admin to volunteer in the request body and will send successfu l response if details get stored in the DB else will send a failure response

* **/getEmergencyContactDetails**

In this API we will pass districtId or districtName to fetch emergency contact details of that particular district from DB

**Postman link for localhost:**

<https://www.getpostman.com/collections/ee64fbecb46f4567d518>

**Postman link for Dev Server: 15.207.42.209**

<https://www.getpostman.com/collections/8376c150946c2b5aabf8>

**Postman link for Production Server : 3.7.39.123**

<https://www.getpostman.com/collections/cc985f49252ef06e12e6>

**Solution to solve the string fields to accept multiple languages and store them in that particular language in DB and return as it is in response object:**

**1.Execute the below statement for particular column where it needs to accept UTF-8 characters**

ALTER TABLE `KEFUSER`.`greivance\_tracking` MODIFY underreviewremarks VARCHAR(255) CHARACTER SET UTF8;

**2.Add the below configuration steps in application.properties file for that particular DB details**

spring.jpa.properties.hibernate.connection.characterEncoding=utf-8

spring.jpa.properties.hibernate.connection.CharSet=utf-8

spring.jpa.properties.hibernate.connection.useUnicode=true

**Process to load Volunteer and SrCitizen details into DB**

**1.**We will get an excel of data listed with senior citizen details assigned to volunteers.

2.Separate the volunteers list and give idvolunteer starting with last idvolunteer from DB + 1 and fill the rest of the ids in series

3.Add rest of the columns existing in volunteer table from DB which are not provided in excel

4.Give CURRENT\_TIMESTAMP for all date time fields

5.Copy All senior citizen details into another sheet along with corresponding volunteer first name and last name

6.Assign callid for each senior citizen as mentioned in step 2

7.Do a vlookup to get the idvolunteer from volunteer sheet into senior citizen list sheet to get corresponding idvolunteer for the particular senior citizen based on first name of volunteer and then paste them as values

8.Remove volunteer first name, last name and then add rest of the columns from volunteer assignment table to the sheet and make sure you add callstatuscode as 1 for all the senior citizens and use NULL for rest of the fields

9.Save both excels as CSV(Comma separated Values) and open them in notepad++

10.Replace ‘CURRENT\_TIMESTAMP’ with CURRENT\_TIMESTAMP

11.Open <https://sqlizer.io/> where we can convert csv files to queries

12.Execute queries in MySQL Workbench to upload data into volunteer and volunteerassignment tables respectively

We are using MD5 algorithm to encrypt Admin password and store it in DB. If we are entering admin details manually so we need to encrypt the password before storing it in DB

**CallStatusCodes we are using for DB purpose**

1.Pending

2.Not Picked

3.Not Reachable

4.Number Busy

5.Call Later

6.Call Dropped

7.Wrong Number

8.Number doesn't exist

9.Disconnected

10.Connected

**Quarantinestatus codes we are using for DB purpose**

0.Not quarantine

1. Home Quarantine

2. Govt Quarantine

3. Hospitalized

**For all grievances such as food shortage, medicine shortage we are using below codes for DB purpose**

1. Raised

2. Under Review

3. Resolved

4. No issue

**Roles codes we are using for DB purpose**

1.Volunteer

2.Staff member

3.District Admin

4.Master Admin

**Enum Values we are using to store the corresponding english text for few fields are mentioned below:**

**TalkedWith enum values**

SENIOR\_CITIZEN = "1";

FAMILY\_MEMBER\_OF\_SR\_CITIZEN = "2";

COMMUNITY\_MEMBER = "3";

**InputEnum values**

Y = "1";

N = "2";

**related\_info\_talked\_about enum values**

PREVENTION = "1";

ACCESS = "2";

DETECION = "3";

**behavioural\_change\_noticed enum values**

YES = "1";

NO = "2";

MAY\_BE = "3";

NOT\_APPLICABLE = "4";

**Package Structure**

