

## Doctors

models -> DoctorSchema holds the schema for the Doctor records.

- A brief overview of the schema and the corresponding pre/post middleware methods is as follows:
  - Patient slug creation - pre
  - Creation of a location field and geocode - pre
  - Password encryption - pre
  - Sign and return JWT token - getSignedJwtToken
  - Password matching between user input and hashed password in database - match password
  - Generate and hash password token - getResetPasswordToken

controllers -> The controller methods used on patients are stored here.

- The controller methods are as follows:
  - getDoctors
    - This method attempts to retrieve the data of all of the doctors from the database in the form of a JSON.
  - getDoctor
    - This method attempts to retrieve the data of a specific doctor from the database in the form of a JSON.
  - getPatientsWorked
    - This method attempts to retrieve the data of all of the patients that have been worked by a specific doctor from the database in the form of a JSON.
  - getDoctorAppointments
    - This method attempts to retrieve the appointments for a specific doctor from the database in the form of a JSON.
  - postdoctorAppointment
    - This method attempts to update all of the data in the database to reflect the changes in doctor status following the scheduling of an appointment, such as if a new patient needs to be added to the list of patients worked by a doctor. This method also attempts to send an email to the patient informing them of the scheduled appointment.
  - getDoctorReview
    - This method attempts to retrieve all of the reviews of a specific doctor from the database in the form of a JSON.
  - createDoctor
    - This method attempts to create new data for a new doctor in the database.

- loginDoctor
  - This method attempts to authenticate the login for a doctor, and sends an error message if the credentials are invalid, and sends an authentication token response if the login is correct.
- updateDoctor
  - This method attempts to update the data for a specific doctor within the database
- addDoctorReview
  - This method allows the addition of a review of a specific doctor to the database by a specific patient. Only one review is allowed for a specific doctor from the same patient.
- deleteDoctor
  - This method attempts to delete the data of a specific doctor from the database.
- getDoctorsInRadius
  - This method gets a list of the doctors within a specific radius of a location.
- doctorPhotoUpload
  - This method accepts a file for uploading, and ensures that the file fits the upload criteria such as being a photo and adhering to a file size limit. The method also provides functionality to allow the user to create a filename for the photo file.
- sendTokenResponse
  - This method gets a token from the model and creates a cookie and a send response.

routes -> The API calls are all stored here.

- We have the below routes for Doctor:
  - getDoctors
    - This route is unprotected, and anyone may access it.
  - getPatientsWorked
    - This route is protected, and users must be a doctor to access it.
  - postdoctorAppointments
    - This route is unprotected, and anyone may access it.
  - createDoctor
    - This route is unprotected, and anyone may access it.
  - loginDoctor
    - This route is unprotected, and anyone may access it.
  - getDoctor
    - This route is unprotected, and anyone may access it.
  - updateDoctor
    - This route is protected, and users must be a doctor to access it
  - deleteDoctor

- This route is protected, and users must be a patient to access it
- getDoctorReview
  - This route is unprotected, and anyone may access it.
- addDoctorReview
  - This route is protected, and users must be a patient to access it
- getDoctorsInRadius
  - This route is unprotected, and anyone may access it.
- doctorPhotoUpload
  - This route is protected, and users must be a doctor to access it.