

MedicController
+ updateMedic(Integer, Medic): Medic + getAllMedics(): Iterable<Medic> + createMedic(Medic, UriComponentsBuilder): ResponseEntity<Medic> + deleteMedic(Integer): Medic + getMedicById(Integer): ResponseEntity<Medic>

TherapyController
+ updateTherapy(Integer, Therapy, UriComponentsBuilder): Therapy + getAllTherapies(): Iterable<Therapy> + createTherapy(Therapy, UriComponentsBuilder): ResponseEntity<Therapy> + getTherapyById(Integer): ResponseEntity<Therapy> + deleteTherapy(Integer): Therapy

PatientController
+ createPatient(Patient, UriComponentsBuilder): ResponseEntity<Patient> + deletePatient(Integer): Patient + getAllPatients(): Iterable<Patient> + getPatientById(Integer): ResponseEntity<Patient> + updatePatient(Integer, Patient): Patient

ReportController
+ getReportsByUserId(Integer): Iterable<Report> + deleteReportById(Integer): Report + updateReport(Report, Integer): Report + updateUserReport(Report, Integer): Report + getUserReportById(Integer, Integer): Optional<Report> + getReportById(Integer): Optional<Report> + createReport(Report, Integer, UriComponentsBuilder): ResponseEntity<Report> + deleteUserReportById(Integer, Integer): Report + getAllReports(): Iterable<Report> + createUserReport(Report, Integer, UriComponentsBuilder): ResponseEntity<Report>

UserController
+ login(User, UriComponentsBuilder): ResponseEntity<?> + getAllUsers(): Iterable<User> + getUserById(Integer): ResponseEntity<User> + register(User, UriComponentsBuilder): ResponseEntity<?> + getPendingUsers(): Iterable<User> + updateUser(Integer, User): User + updatePendingUser(Integer): User + deleteUser(Integer): User

NotificationController
+ createUserNotification(Notification, Integer, UriComponentsBuilder): ResponseEntity<Notific + getAllNotifications(): Iterable<Notification> + updateUserNotification(Notification, Integer): Notification + getNotificationByUser(Integer, Integer): Optional<Notification> + updateNotification(Notification, Integer): Notification + createUserNotification(Notification, Integer, Integer, UriComponentsBuilder): Notification + deleteNotificationById(Integer): Notification + deleteUserNotificationByUserId(Integer): Notification + getAllNotificationByUser(Integer): Iterable<Notification> + getNotificationById(Integer): Optional<Notification>

MedicChangeLogController
+ getChangeLogByPatientId(Integer): MedicChangeLog + getChangeLogById(Integer): ResponseEntity<MedicChangeLog> + getChangeLogByMedicId(Integer): Iterable<MedicChangeLog> + getAllChangeLogs(): Iterable<MedicChangeLog> + createMedicChangeLog(MedicChangeLog, UriComponentsBuilder): ResponseEntity<MedicChangeLog> + updatePatient(Integer, MedicChangeLog): MedicChangeLog + delete(Integer): MedicChangeLog