(هرجا با آیدی کار داشتیم قبلش چک میکنیم که objectId صحیح است یا خیر)

Group

getAll

۱. تمام groupها از دیتابیس دریافت میشوند

۲. تمام groupها برگشت داده میشوند

Create

۱. وضعیت ادمین بودن کاربر چک میشود

۲. وجود ticketing در roleهایی ادمین چک میشود

۳. نام group از request body دریافت میشود

۴. نام group چک میشود که خالی نباشد و بین ۳ تا ۴۰ کاراکتر باشد

۵. شی group جدید ساخته میشود و در دیتابیس قرار میگیرد

۶. شی message با مقادیر message و data (شی group جدید) برگشت داده میشود

Provider

getAll

۱. تمام providerها از دیتابیس دریافت میشود

۲. تمام providerها برگشت داده میشوند

Create

۱. وضعیت ادمین بودن کاربر چک میشود

۲. وجود ticketing در roleهایی ادمین چک میشود

۳. نام provider از request body دریافت میشود

۴. نام provider چک میشود که خالی نباشد و بین ۳ تا ۴۰ کاراکتر باشد

۵. شی provider جدید ساخته میشود و در دیتابیس قرار میگیرد

۶. شی message با مقادیر message و data (شی provider جدید) برگشت داده میشود

Ticket

getAll

۱. مقادیر page و limit از request query برای pagination دریافت میشوند

۲. وجود مقادیر page و limit چک میشود و در صورت وجود نباید مقادیر غیر عددی باشند و از صفر کمتر باشند

۳. اگر مقدار page وارد نشده بود دیفالت ۱ و مقدار limit در صورت عدم وجود دیفالت ۱۰ در نظر گرفته میشوند

۴. مقدار userId از request.user.sub دریافت میشود (آیدی یوزر وارد شده در سیستم لاگین مرکزی)

۵. از دیتابیس تمام ticketها که userId آنها برابر با userId مرحله قبل باشد با مقدار page و limit بصورت paginate دریافت میشود و در نتیجه این دریافت از propertyهای provider و group هر ticket فقط name آنها دریافت میشود

۶. نتیجه برگشت داده میشود

getAllNewTicketsAsGroup

۱. وضعیت ادمین بودن کاربر چک میشود

۲. وجود ticketing در roleهایی ادمین چک میشود

۳. مقادیر page و limit از request query برای pagination دریافت میشوند

۴. وجود مقادیر page و limit چک میشود و در صورت وجود نباید مقادیر غیر عددی باشند و از صفر کمتر باشند

۵. اگر مقدار page وارد نشده بود دیفالت ۱ و مقدار limit در صورت عدم وجود دیفالت ۱۰ در نظر گرفته میشوند

۶. مقدار groupName از request query دریافت میشود

۷. از دیتابیس یک group با groupName در دسترس دریافت میشود

۸. اگر group با groupName داده شده یافت نشود و groupName برابر با all نباشد ارور نام گروه معتبر نیست برگشت داده میشود

۹.در صورت معتبر بودن groupName از دیتابیس تمام ticketها که status=0 (یعنی همان NEW) و group آنها برابر با آیدی group دریافت شده در مرحله ۵ باشد دریافت میشوند و در نتیجه این دریافت از propertyهای provider و group هر ticket فقط name آنها دریافت میشود

۱۰. نتایج برگشت داده میشوند

getOne

۱. از request params مقدار id دریافت میشود

۲. اگر id معتبر نبود ارور برگشت داده میشود

create

(فایل در اولین قدم آپلود میشود و req.file ساخته میشود)

(یک property به اسم attachment به شی request body اضافه میکنیم و مقدار آن در صورت وجود req.file برابر با req.file.path است و در صورات عدم وجود undefined است)

(در هر مرحله که به ارور برخورد کردیم و ساخت تیکت موفقیت آمیز نبود فایل رو پاک میکنیم)

۱. تمام propertyهای آبجکت request body بررسی میشوند:

-آیتم group خالی نباشد، چون از نوع objectId است دقیقا ۲۴

کاراکتر از نوع string باشد

-آیتم title خالی نباشد، کمتر از ۵ و بیشتر از ۷۰ کاراکتر نباشد،

کاراکتری بجز حرف و عدد نداشته باشد

-آیتم issue خالی نباشد، کمتر از ۱۵ و بیشتر از ۴۰۰ کاراکتر نباشد،

کاراکتری بحز حرف و عدد نداشته باشد

-آیتم priority خالی نباشد و فقط ۱،۲یا ۳ باشد

-آیتم provider میتواند ارسال نشود اما در صورت ارسال نمیتواند

خالی باشد،‌ چون از نوع objectId است دقیقا ۲۴ کاراکتر از نوع

string باشد

نوع string

-آیتم workspaceId میتواند ارسال نشود اما درصورت ارسال باید

عددی صحیح باشد

-آیتم attachment میتواند ارسال نشود اما در صورت ارسال

نمیتواند خالی باشد و از باید از نوع string باشد

۲. اگر req.fileError وجود داشت ارور ۴۰۰ همراه خود req.fileError برگشت داده شود (این property در زمان چک کردن نوع و پسوند فایل چک میشود)

۳. با استفاده از فیلد group ارسالی، از دیتابیس group مربوط دریافت میشود و در صورت عدم وجود group ارور میدهد

۴. برای فیلد provider مثل مرحله قبل پیش میرویم

۵. اگر workspaceId ارسال شده بود، یک get request با هدر authorization که برابر با authorization موجود در request header کاربر فعلی است به workspace api زده میشود. اگر workspaceای پیدا نشد ارور داده میشود. اگر workspaceهایی پیدا شد، درصورتی که id هرکدام از آنها با workspaceId ارسالی برابر باشد ادامه میدهیم در غیر اینصورت ارور شناسه میزکار معتبر نیست برمیگردانیم

۶. یک object از ticket با تمام آیتمهای ارزیابی شده میسازیم و numId که یک رشته رندوم است را به آن اضافه میکنیم

۷. تیکت رو داخل دیتابیس ذخیره میکنیم

۸. مجددا از دیتابیس یک ticket رو با \_id تیکت جدید ساخته شده دریافت میکنیم (از فیلدهای group و provider فقط name)

۹. یک object به اسم message با فیلدهای message و data ساخته و به کاربر برمیگردانیم

۱۰. فایل ارسالی رو با استفاده از multer ذخیره میکنیم

close

۱. مقدار id از request params دریافت میشود

۲. اگر id از نوع objectId نبود، ارور میدهد

۳. از دیتایس ticket با مقدار id دریافتی دریافت میشود

۴. اگر status در ticket مربوطه برابر با ۲ (closed) بود تیکت از قبل بسته شده

۵. اگر کاربر فعلی (req.user.sub) با کاربر سازنده تیکت (ticket.userId) برابر نبود و همچنین کاربر ادمین نبود یا ادمینی بود که ticketing role را نداشت ارور بده

۶. وضعیت status رو به ۲ تغییر بده و به closedAt مقدار زمان حال رو بده

۷. یک object به اسم message با فیلدهای message و data ساخته و به کاربر برمیگردانیم

download  
۱. مقدار id از request params دریافت میشود

۲. اگر id از نوع objectId نبود، ارور میدهد

۳. از دیتایس ticket با مقدار id دریافتی دریافت میشود

۴. اگر کاربر فعلی (req.user.sub) با کاربر سازنده تیکت (ticket.userId) برابر نبود و همچنین کاربر ادمین نبود یا ادمینی بود که ticketing role را نداشت ارور بده

۵. اگر ticket مقدار attachment را نداشت ارور بده هیچ فایلی مربوط به این تیکت نیست

۶. فایلی که مسیر آن توی ticket.attachment ذخیره شده رو برای کاربر ارسال کن

comment

create

(فایل در اولین قدم آپلود میشود و req.file ساخته میشود)

(یک property به اسم attachment به شی request body اضافه میکنیم و مقدار آن در صورت وجود req.file برابر با req.file.path است و در صورات عدم وجود undefined است)

(در هر مرحله که به ارور برخورد کردیم و ساخت کامنت موفقیت آمیز نبود فایل رو پاک میکنیم)

۱. تمام properyهای request body بررسی میشوند:

-مقدار ticket خالی نباشد و چون از نوع objectId است دقیقا ۲۴

کاراکتر باشد

-مقدار content خالی نباشد حداقل ۱۵ و حداکثر ۴۰۰ کاراکتر باشد

و چیزی بجز حرف و عدد نباشد

-مقدار attachment میتواند خالی باشد و اگر نبود باید string باشد

۲.اگر req.fileError وجود داشت ارور ۴۰۰ همراه خود req.fileError برگشت داده شود (این property در زمان چک کردن نوع و پسوند فایل چک میشود)

۳. صحیح بودن ticket که از نوع objectId است و همینطور وجود ticket بررسی شود

۴. اگر status برای ticket برابر با ۲ بود ارور تیکت بسته شده است بده

۵. اگر کاربر فعلی (req.user.sub) با کاربر سازنده تیکت (ticket.userId) برابر نبود و همچنین کاربر ادمین نبود یا ادمینی بود که ticketing role را نداشت ارور بده

۶. یک شی comment جدید بساز، مقدار userId رو برابر آیدی یوزر فعلی (req.user.sub) قرار بده

۷. تمام propertyهای بررسی شده رو به comment جدید اضافه کن

۸. اگر کاربر فعلی ادمینی است که ticketing role هم دارد، مقدار isAdminComment رو true قرار بده، اگر ticketing role را ندارد یا اصلا ادمین نیست false قرار بده

۹. اگر کاربر فعلی ادمینی است و ticketing role رو دارد مقدار seenByAdmin رو true قرار بده، اگر ticketing role را ندارد یا اصلا ادمین نیست false قرار میدیم

۱۰. اگر کاربر ادمین نیست یا ادمینی است که ticketing role را ندارد مقدار seenByUser رو true قرار بده و اگر ادمینی است که ticketing role هم دارد false قرار بده

۱۱. در دیتابیس comment رو ذخیره کن

۱۲. یک object به اسم message با فیلدهای message و data ساخته و به کاربر برمیگردانیم

seenByUser

۱. مقدار id از request params دریافت میشود

۲. مقدار id بررسی میشود که از نوع objectId باشد

۳. از دیتابیس comment به همراه ticket مرتبط با آن دریافت میشود

۴. مقدار seenByUser در comment دریافتی به true تغییر میکند

۵. فقط statusCode برگشت داده میشود

seenByAdmin

۱. اگر کاربر ادمین نبود یا ادمینی بود که ticketing role را نداشت ارور بده

۲. مقدار id از request params دریافت میشود

۳. مقدار id بررسی میشود که از نوع objectId باشد

۴. از دیتابیس comment به همراه ticket مرتبط با آن دریافت میشود

۵. مقدار seenByAdmin در comment دریافتی به true تغییر میکند

۶. فقط statusCode برگشت داده میشود

download

۱. مقدار id از request body دریافت میشود

۲. صحیح بودن objectId چک میشود

۳. از دیتابیس comment مربوطه دریافت میشود

۴. اگر کاربر فعلی (req.user.sub) با کاربر سازنده تیکت (ticket.userId) برابر نبود و همچنین کاربر ادمین نبود یا ادمینی بود که ticketing role را نداشت ارور بده

۵.. اگر comment مقدار attachment را نداشت ارور بده هیچ فایلی مربوط به این تیکت نیست

۶. فایلی که مسیر آن توی comment.attachment ذخیره شده رو برای کاربر ارسال کن