Olex-backnend Documentaion

🕓	Created	@July 12, 2022 6:38 PM
🕓	Last Edited Time	@July 12, 2022 8:10 PM
⊙	Type	Project description
⊙	Status	Completed 🏁
👤	Created By	
👤	Last Edited By	

Description

this is a documantation for each API in this project

$SignUpAPI$

- Method : POST

- req.body

    1. firstName (String , required)

    2. lastName (String , required)

    3. email ((String , , email format (.com or .net only) , unique , required)

    4. password (String, must met pattern (/^(?=.\d)(?=.[a-z])(?=.[A-Z])(?=.[a-zA-Z]).{8,}$/) , required)

    5. cpass(String , must match password , in pattern (/^(?=.\d)(?=.[a-z])(?=.[A-Z])(?=.[a-zA-Z]).{8,}$/) , required)

- Goal of this API : store user information in database , after sign up must check your email to confirm it and it is a neccessary step

$SignInAPI$

- Method : POST

- req.body

- Goal of this API : make user to be signed in by his account email and his password , must signup first

SignOutAPI

- Method : POST

- req.body

- Goal of this API : make user to be signed out by his account email and his password

UpdatePasswordAPI

- Method: PATCH

- req.body

  1. old_password (String, must met pattern (/^(?=.\d)(?=.[a-z])(?=.[A-Z])(?=.[a-zA-Z]).{8,}$/) , required)

  2. new_password(String, must met pattern (/^(?=.\d)(?=.[a-z])(?=.[A-Z])(?=.[a-zA-Z]).{8,}$/) , required)

  3. cpass (String, must met pattern (/^(?=.\d)(?=.[a-z])(?=.[A-Z])(?=.[a-zA-Z]).{8,}$/) , required , must match new_password)

- Goal of this API : make user ability to update his pasword in case of he know his old password

- Who can access this API: the account owner only

UpdateProfileAPI

- Method: PATCH

- req.body

  1. new_email (String , , email format (.com or .net only) , unique , required)

  2. firstName((String, optional)

  3. lastName (String, optional)

- Goal of this API : make user ability to update his email , firstName and lastName , user can update all of three fields or one of them or two of them , but if he update his email , he must confirm it again

- Who can access this API: the account owner only

ForgetPasswordAPI

- Method: PATCH

- req.body

  1. email (String , , email format (.com or .net only) , unique , required)

- Goal of this API : send code in user email to use it in another api called reset api , so these two apis used to make user able to reset his password again after forgetting it

- Who can access this API: the account owner only

$ResetPasswordAPI$

- Method: POST

- req.body

  1. email (String , , email format (.com or .net only) , unique , required)

  2. code (String , required , must be the same code that recieved from forget password API)

  3. new_password (String, must met pattern (/^(?=.\d)(?=.[a-z])(?=.[A-Z])(?=.[a-zA-Z]).{8,}$/) , required)

  4. cNewPass (String, must met pattern (/^(?=.\d)(?=.[a-z])(?=.[A-Z])(?=.[a-zA-Z]).{8,}$/) , required , must match new_password)

- Goal of this API : update user's password by the new password which had been reset

- Who can access this API: the account owner only

$DeleteUserAPI$

- Method: DELETE

- req.body

  1. email (String , , email format (.com or .net only) , unique , optional)

- Goal of this API :

  1. in case of you logged in by admin email $\Rightarrow$ this api check if there is an email send in body, so it delete this user's email from database , and if there is no email send in body so it delete the email of admin that logged in now (you want to delete account , or delete another user account , you cannot delete another admin account)

  2. in case of you logged in by user email $\Rightarrow$ this api delete user's email that logged in now from the database (you want to delete your own account , you cannot delete another user account)

- Who can access this API: the account owner only and Admin

$SoftDeleteUserAPI$

- Method: PATCH

- req.body

  1. email (String , email format (.com or .net only) , unique , required)

- Goal of this API : this api mark user account as deleted but not actually delete it

- Who can access this API: Admin only

AddProfilePictureAPI

- Method: PATCH

- req.form : image (image extension must br in png-jpg-jpeg)

- Goal of this API : this api will add a profile picture for user that logged in now with preserving tha old profile pictures

- Who can access this API: Account owner only

AddCoverPictureAPI

- Method: PATCH

- req.form : image (image extension must br in png-jpg-jpeg)

- Goal of this API : this api will add a cover picture for user that logged in now with preserving tha old cover pictures

- Who can access this API: Account owner only

GetAllUsersAPI

- Method: GET

- Goal of this API : this api will get all users with their informations and with products

- Who can access this API: User OR Admin

AddProductAPI

- Method: POST

- req.body:

  1. Product_title (String , required)

  2. Product_desc (String , required)

  3. Product_price (number , required)

- Goal of this API : this api will add a product and make it created by logged in user now (his _id) , when prouduct created it reflected in all browsers that open the site excpet the user that create the product because he didnot need to be notified by this , he is the owner also this api generated a QR code for each product that anyone can scan to get product info

- Who can access this API: User OR Admin

UpdateProductAPI

- Method: PATCH

- req.body:

    1. Product_title (String , optional)

    2. Product_desc (String , optional)

    3. Product_price (number , optional)

    4. _id (String , must be 24 digit , required)

- Goal of this API : this api search about product with this _id and created by the _id of user that logged in now and update product's fields by req.body

- Who can access this API: product owner only

DeleteProductAPI

- Method: DELETE

- req.body:

    1. _id (String , must be 24 digit , required)

- Goal of this API :

    1. incase of Admin logged in : this api search about product with this _id and delete it

    2. in case of user logged in : this api search about product with this _id and created by the _id of user that logged in now and delete it

- Who can access this API: product owner only and Admin

Like/UnlikeProductAPI

- Method: PATCH

- req.body:

    1. _id (String , must be 24 digit , required)

- Goal of this API : this api search about product by _id and check if its likes array contain the _id of user that logged in now or not , if contain it willpull it form the array , if not contain it will push it into the array , the product owner cannot like his product

- Who can access this API: User OR Admin

SoftDeleteProductAPI

- Method: PATCH

- req.body

    1. _id (String , must be 24 digit , required)

- Goal of this API : this api mark product as deleted but not actually delete it from DB

- Who can access this API: Admin only

HideProductAPI

- Method: PATCH

- req.body

    1. _id (String , must be 24 digit , required)

- Goal of this API : this api mark product as hidden

- Who can access this API: User OR Admin

AddToWishlistAPI

- Method: PATCH

- req.body:

    1. _id (String , must be 24 digit , required)

- Goal of this API : this api search about product by _id and check if its wishlists array contain the _id of user that logged in now or not , if contain it will pull it form the array , if not contain it will push it into the array , Then it search about user by _id of logged in user now and check if its wishlist array contain the _id of the product or not , if contain it will pull it form the array , if not contain it will push it into the array

- Who can access this API: User OR Admin

AddCommentAPI

- Method: POST

- req.body:

    1. comment_body (String , required)

    2. Product_id (String , required , must be 24 digit)

- Goal of this API : this api will add a comment in product with _id equal to Product_id   and make it created by logged in user now (his _id) , when comment created it reflected in all browsers that open the site excpet the user that create the comment because he didnot need to be notified by this , he is the owner also this api generated a QR code for each comment that anyone can scan to get comment info

- Who can access this API: User OR Admin

AddReplyAPI

- Method: POST

- req.body:

    1. comment_body (String , required)

    2. Comment_id (String , required , must be 24 digit)

- Goal of this API : this api will add a Reply in Comment with _id equal to Comment_id and make it created by logged in user now (his _id) , when reply created it reflected in all browsers that open the site excpet the user that create the reply because he didnot need to be notified by this , he is the owner also this api generated a QR code for each reply that anyone can scan to get comment info

- Who can access this API: User OR Admin

UpdateCommentAPI

- Method: PATCH

- req.body:

    1. comment_body (String , required)

    2. _id (String , required , must be 24 digit)

- Goal of this API : this api search about comment with this _id and created by the _id of user that logged in now and update comment's fields by req.body

- Who can access this API: comment owner only

DeleteCommentAPI

- Method: DELETE

- req.body:

    1. _id (String , must be 24 digit , required)

- Goal of this API :

    1. in case of Admin logged in : this api search about comment with this _id and delete it

    2. in case of user logged in : this api search about comment with this _id and created by the _id of user that logged in now and delete it

- Who can access this API: product owner only and Admin

CronJob PDF

this function create pdf contain the products that cretaed today and send it for all admins at 11:59:59 PM every day

All pictures of pdfs uploaded to the uploads file