

GraphQL_Project Documentation

🕒 Created	@July 26, 2022 4:02 PM
🕒 Last Edited Time	@July 26, 2022 7:49 PM
▼ Type	
▼ Status	Completed 🏁
👤 Created By	
👤 Last Edited By	

User APIs Documentation:

- the route link of any user api "localhost:3000/userAPI"



Add User API

- in mutation not query
- must send these data for each user :
 1. name (must be string , required)
 2. email (must be string , must be in domain(.com or .net) only , must be valid format of email , required , must be unique)
 3. password(must be string , match the pattern like 'Amira112233@@@' , required)
 4. cPassword (must be string , match thepassword , required)
- Goal of this API : will add a new user for your database , and confirm it , after add user , the user must check his email ti verify it.



Sign In API

- in mutation not query
- must send these data :
 1. email (must be string , must be in domain(.com or .net) only , must be valid format of email , required , must be unique)
 2. password(must be string , match the pattern like 'Amira112233@@@' , required)
- Goal of this API : user can log in to his account by his email and his password (his email must be confirmed before), and return the token this user that hold his information.



Update User API

- in mutation not query
- must send these data for each user :
 1. name (must be string , optional)
 2. email (must be string , must be in domain(.com or .net) only , must be valid format of email , optional, must be unique)
 3. token (must be string , valid token , required)
- Goal of this API : will take the email of the user that logged in now and update his email by this email or update his name or update both of them. , if the user update his email , he must verify it also
- Access to this API: only the account owner(user) can access this api.



delete User API

- in mutation not query
- must send these data for each user :
 1. email (must be string , must be in domain(.com or .net) only , must be valid format of email , optional)
 2. token (must be string , valid token , required)
- Goal of this API :
 - if the role of the token is Admin \Rightarrow there are two cases : first case, if there is email send and the role of this email is user , so this email will be deleted by the admin. seconde case , if there is no email send , so the account of this admin that logged in now will be deleted
 - if the role of the token is user , so the user account that logged in now will be deleted
- Access to this API: the account owner(user) and the admin can access this api.



Forget Password API

- this api divided into two parts :

1- Send Code API:

- in mutation not query
- must send these data for each user :
 1. email (must be string , must be in domain(.com or .net) only , must be valid format of email ,required)
- Goal of this API : will send a code to this email to use it in reset password api

2- Reset Password API :

- in mutation not query
- must send these data for each user :
 1. email (must be string , must be in domain(.com or .net) only , must be valid format of email ,required)
 2. code (must be string , required)
 3. newPassword(must be string , match the pattern like 'Amira112233@@@' , required)
 4. cPassword (mist be string , match the newpassword , required)
- Goal of this API : will take the email and the code then get the user that has the two fields which match their values **together** and update this password by the newPassword after hashing it



Get All User API

- in query to in mutation
- must send these data for each user \Rightarrow require no data
- Goal of this API : get all users in the database

Blog API Documentation

- the route link of any user api "localhost:3000/blogAPI"



Add Blog API :

- in mutation not in query
- must send these data for each blog:
 1. title (must be string , required)
 2. desc (must bt string , required)
 3. price (must be number , requird)
 4. token (must be string , requirde, valid token)
- Goal of this API : add a blog in database createdBy the id included in the token.



Update Blog API :

- in mutation not in query
- must send these data for each blog:
 1. title (must be string , optional)
 2. desc (must be string , optional)
 3. price (must be number , optional)
 4. token (must be string , required, valid token)
 5. id (must be string , must contain 24 digit , required)
- Goal of this API : get the blog of this id and check if the logged in user is the owner of the blog or not , if he is the owner will update the title or the price or the desc or all of them , if not the owner ,will not update any thing
- Access to this API: only the blog owner(user) can access this api.



Delete Blog API :

- in mutation not in query
- must send these data for each blog:
 1. token (must be string , required, valid token)
 2. id (must be string , must contain 24 digit , required)
- Goal of this API : check if :
 - the role of the token is admin \Rightarrow so delete the blog of the id that send
 - the role of the token is user \Rightarrow check if this user is the owner of the blog or not , if the owner \Rightarrow will delete the blog of the id that send , else \Rightarrow will not delete
- Access to this API: the blog owner(user) and the Admin can access this api.



Get all Blogs API :

- in mutation not in query
- must send these data for each blog \Rightarrow no data required
- Goal of this API : get all blogs with the name and the email of user that create each blog
- Access to this API: any user can access this api (whether logged in or not)



Get Blog By its ID API :

- in mutation not in query
- must send these data for each blog:
 1. id (must be string , must contain 24 digit , required)
- Goal of this API : get the blog information of this id with the name and the email of the creator user
- Access to this API: any user can access this api (whether logged in or not)