

API Documentation

We used GraphQL to handle queries between Express and React. The advantage of using GraphQL is if you send a GraphQL query to your API and get exactly what you need, nothing more and nothing less. GraphQL queries always return predictable results. Apps using GraphQL are fast and stable because they control the data they get, not the server. So In Backend we used GraphQL for handling requests except login and registration.

Login

For login the API used is `'/api/auth/login'` and type is post request. This API returns the user data and Json Web Token to the frontend. This JWT is used to protect the requests of information of other users. for verifying JWT we are using PassportJs. PassportJs decodes the JWT and checks the user in the database. if there then it is allowed to make requests.

Register

For Register the API used is `'/api/auth/register'` and type is post request. This API returns the user data and Json Web Token to the frontend. This post request register the user with his info in database.

For handling GraphQL queries we have to do schema and rootValues, Schema is a GraphQL Schema where all the funtions all declared and return values of them are also declared. rootValues are resolver functions of functions which are declared in schema. The API for all this requests is `'/graphql'`, Here the requests are divided into two types 1) Queries 2) Mutations. Queries are similar to get requests and Mutations are similar to post requests.

Queries and their functionalities are in the below table

Query functions	Functionality
getUser	Returns the user data like email,name,phonenumner etc.
getMyNewspapers:	Returns an array of information of Subscribed newspapers for the logged in User like newspaperName,selected avatar,selected language etc.,
getMyNewspaper(newspaperid)	Returns the information of Subscribed newspaper with id=newspaperid like newspaperName,selected avatar,selected language etc.,
allNewspapers	Returns an array of information of Newspapers like newspaper avatar, newspaper name, newspaper description etc. except the subscribed newspapers by logged in user
allAvatars	Returns an array of all avatars and its information like number of subscriptions of this avatar(no of times the avatar has been taken as avatar for subscribed newspaper)

Query functions	Functionality
popularNewspapers	Returns an array of Eight Newspapers, Which have more no of subscriptions(these are increased when a user subscribes the paper)
popularAvartars	Returns an array of Eight Avatars, Which are have no of subscriptions(these are increased when a avatar has been taken as avatar for subscribed newspaper)
getNewspaper(newspaperid)	Returns information of Newspaper with id = newspaperid like newspaper avatar, newspaper name, newspaper description
getAvatar(avatarid)	Returns avatar and its information with id = avatarid like number of subscriptions of this avatar(no of times the avatar has been taken as avatar for subscribed newspaper)
getProfilePic	Returns the logged in user's profile avatar url.

Mutations and their functionalities are in the below table

Mutatation functions	Functionalities
updateUser(UserInput)	It takes the UserInput arguments and updates the fields of the user with data in the fields of UserInput. And it returns updated user data like email , name etc.
subscribeNews(newsInput)	The logged in user subscribes the Newspaper with id = newsInput.newspaperid. Subscribing a Newspaper means it stores the information like avatar, language,voice,time,tags for a Newspaper. these fields are used to make a video of reading the headlines of the newspaper by the avatar with the language and voice given and sent it at the time mention
updateMyNewspaper(newspaperid,newsInput)	It updates the fields of a subscribed Newspaper with id = newspaperid , with data in the fields of newsInput and returns the updated subscribed information.
deleteMyNewspaper(newspaperid)	It deletes the subscription of the Newspaper or unsubscribes the Newspaper with id = newpsaperid, i.e it deletes the information of the subscription like avatar, language etc.
setAvatarforPapers(setAvatarInput)	It updates the avatars of subscribed newspapers with ids in setAvatarInput.newspaperids with the avatar of id = setAvatarInput.avatarid.

Mutatation functions	Functionalities
addAvatar(avatarInput)	It stores the avatar with information in avatarInput like it stores avatar url, and name of the avatar, if this function is called when upload the user id who has uploaded this avatar is also stored.
addNewspaper(newspaperInput)	it stores the newspaper with information in newspaperInput like newspaper avatar, newspaper name, description etc. it is used to just populate the database with newspapers it is not called by frontend
signS3(filename,filetype)	it gets the signedUrl by sending the filename to s3 with appropriate params, and also returns the url for the file that has to be uploaded.
profilepic(avatarUrl)	It changes the user's profile picture Url to the avatarUrl . so basically changes profile picture.
delProfilePic(filename,filetype)	it deletes the file with the passed filename in s3
delUploadedAvatar(avatarid)	it deletes the user's uploaded avatar which id =avatarid , which deletes the information of avatar also , like name, noof subscriptions.