Post API VS Get API thought process.

Date
Page

The thought process of Get API is different from Post API

Brief information.

whenever you look API we look it at the lense that this API is Post API, or Get API

Post API

-> what happens in Post API (think like you are the Guard of the database.

what is the job of the Guard

~~The Gov Guard of the Security~~

The job of the Guard is, not to let in any random people inside only authorized people can go inside. and if someone is going out side from inside it make sure that no data is leaking

Post API means user is trying to enter data to to database

Get API means user is trying to fetch the data from the database.

how can attackers can exploit a post API

- They can attack your post API by sending some random data into your api and we mistakenly put that data into database (if we put data that we dont wanted to. that the opportunity of attacker that is something we dont want to do as a Guard of the database.

Post threat → attacker can send malicious data to our database

get threat → attacker can leak or get unauthorize data.

Date
Page

~~Whenever~~

for POST API → Always think that attacker can put anything to our database, and ~~have OCD~~ we should have OCD that we will verify everything which is comming it to the request.

(sanitize) (verify each and everything we will make sure 100% whenever we are writing.
↓
(validate)
↓
(do all checks)
↓
(calculation)
↓
(.save())
↓
(send res)

connectionData.save(); ← it should be the second last line and we have to sanitize and ~~verify~~ validate everything before doing this

do all checks

Thought process change very different at get API

Here we are not ~~passing~~ saving any information to our Database, here we are getting information from database or we are fetching data for the user

In get APIs we will 100% make sure that we are sending only allowed data. (make sure user is verified user and whatever he is requesting is allowed for his scope

Always make sure what you are sending back to the date is 100% authorised and is in the user scope.

Date
Page

const hideUserFromFeed = new Set();

Set() ⇐ is like an array but if we put duplicate values it will ignore the duplicate values and always have unique values

① find all the rejected or saved request which user has done.

② we will find unique Id ⇐ create Set and push to all to Set.

const user = await User.find({
$or: [{fromUserId: {$in: Array.from(hideUserFromFeed)}},
{toUserId: {$in: loggedInUser._id}}]

$ and $ or $ in ⇐ not in $ne ⇐ not equal to.

.select(); ⇐ only select the data only data we want.

Date
Page

Pagination.

① we have created all the api now suppose we have 1000 user then our feed should not load 1000 user data. it should load in 10, 10, 10 &.... stack.

↑

for this we will use pagination.

best way is using query params.

/feed?page=1&limit=10 ⇒ first 10 user 1-10 ↗ skip(0) & limit(10)

/feed?page=2&limit=10 → 11-20 ↗ skip(10) & limit (10)

/feed?page=3&limit=10 → 21-30 ↘ skip(20) & limit(10)

in mongoDB we have two function

① . skip() ← How many doc you want to skip from the 1st (start)

② . limit() ← how many document do you want

~~and it~~ .skip(0) & .limit(10) → give first 10

skip = (page-1) * limit

.skip(skip)

let limit = parseInt(req.query.limit) || 5;

~~limit = Math.min~~

limit = Math.min(limit, 10)

↑ sanitized the data

max limit is 10

Completed All backend work for Trader Circle

↓

Now it's time to create frontend.

↓

All frontend notes are on frontEnd (2) book, in first book we have diffrent Project in (2) book we will have TradeCircle frontend Project.