



NAMASTE NODE.JS SEASON 2

Episode - 13

Hand Written Notes

-By Shanmuga Priya

www.linkedin.com/in/shanmuga-priya-e-tech2

Ref, Populate & Thought process of writing APIs

Thought process of writing API's

POST:

→ we should think what are all the possible ways that attackers can attack a POST API.

→ eg: sending some random data to API by attackers.

→ validate at Schema level, API level validation.

GET:

→ we should make ^{sure} ~~sure~~ the data we are sending back to client

→ check for authorization of the user and he has permission to access the protected routes

1) How to make connection between two collections?

→ In MongoDB, we make connection between 2 collections using "reference". Instead of embedding the entire document we store a reference (usually ID) of that doc in different collection.

→ This is a common way to represent one to many & many to many relationships.

2) what is ref?

→ It is used to define a relationship between 2 collections

→ It specifies which model/collection a field is referencing to

eg: Lets create 2 collection

// user Schema

const userSchema = new mongoose.Schema({

name: string,
email: string

3)

// post schema

```
const postSchema = new mongoose.Schema({  
  title: string,  
  content: string,  
  author: {  
    type: mongoose.Schema.Types.ObjectId,  
    ref: "User" }  
})
```

→ it made connection to user schema using Id as reference.

3)

3) what is populate?

→ It is used to retrieve the related document based on reference ID from the related collection.

eg. → "populate()" method accept 2 args (1st - field which has reference that we want to populate, 2nd - what are the data we want to retrieve from that reference document (optional))

→ we can give the 2nd args as [] separated by ,
or) as a single string separated by space.

eg.

```
const post = await Post.findOne({ title: "first post" })
```

```
  .populate('author', ["name"])
```

```
  console.log(post)
```

↓
it gives the post content along with author field & set with "_id" & "name" fields.