

A MODULE IS A PACKAGE

That encompasses all the files related to one particular entity inside NestJS application.

USERS MODULE
(`users.module.ts`)

`users.controller.ts`

`users.service.ts`

`user.entity.ts`

`users.controller.spec.ts`

FILES IN A MODULE

`users.controller.ts`

Contains routing logic API endpoints.

`users.service.ts`

Logic related to a particular functionality.

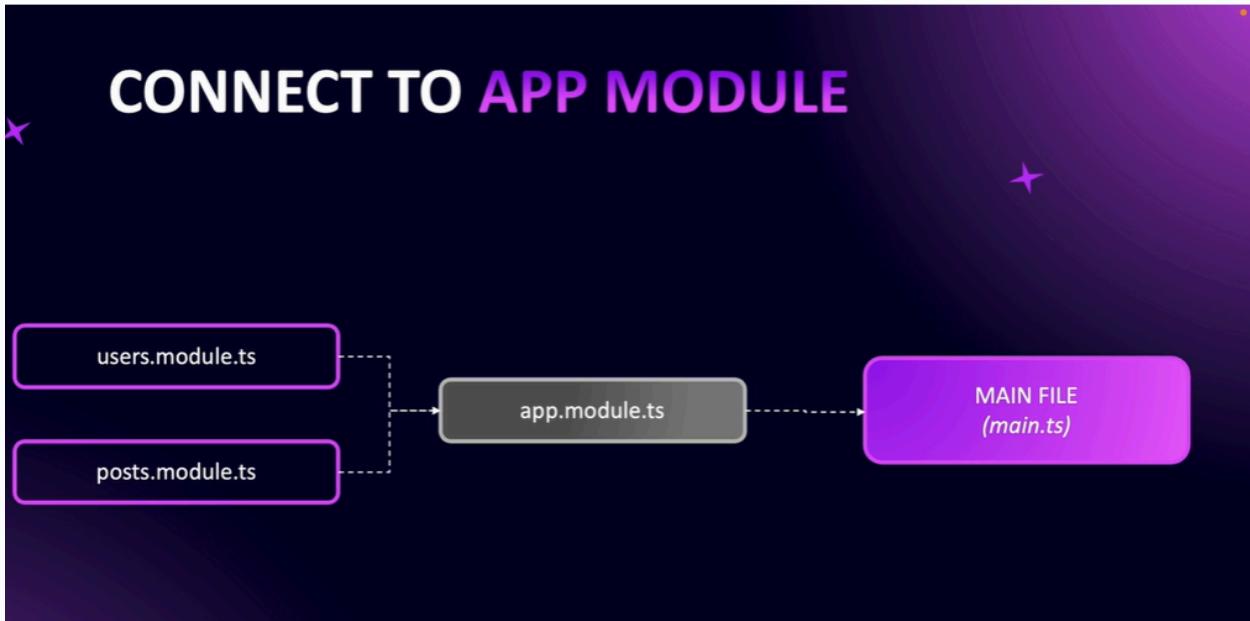
`user.entity.ts`

Database structure for users table.

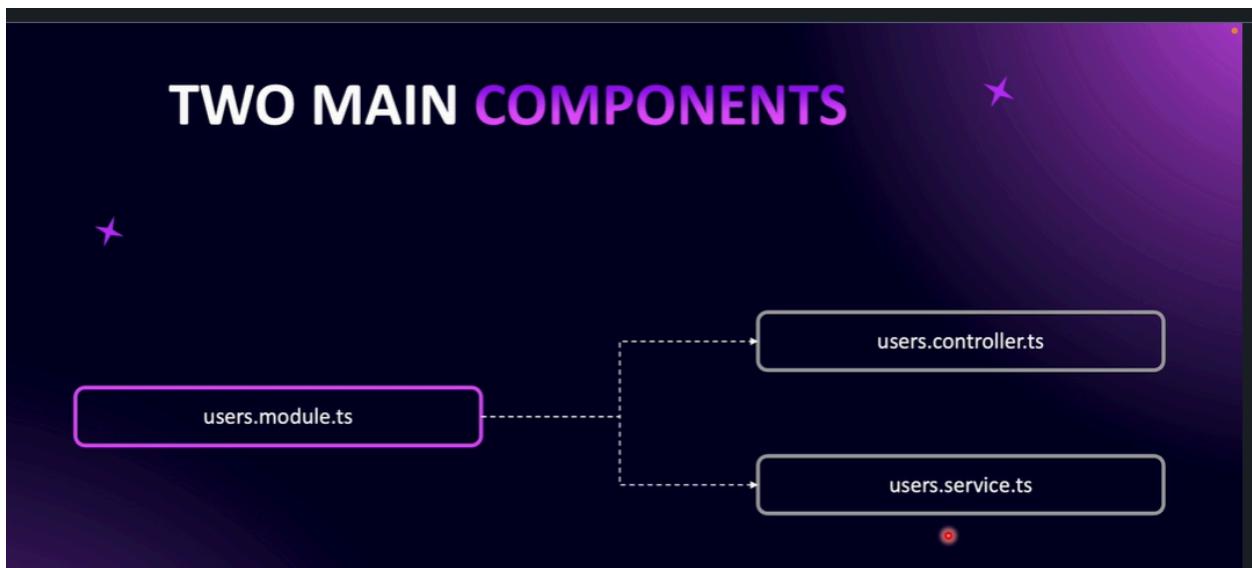
`users.controller.spec.ts`

File to write test cases for users controller.

Moudle based airtchture inside application a losts moulde like a losts app inside related file inside it



User moule also connect with post moulde two moulde connects





REST DESIGN PRINCIPLES

List of design principles followed by REST API Architecture

Client Server Decoupling

Statelessness

Cacheability

Uniform Interface

Layered System

Code On Demand (opt)

13. What is a REST API

CLIENT SERVER DECOUPLING

To exit full screen, press Esc



Client
(browser)



HTTP Verbs

Server



★ THE OFFICIAL DEFINITION

The REST API or RESTful API is an Application Programming Interface (API) that conforms to the design principles of Representational State Transfer (REST) architectural style of programming.

GET

► The GET method requests a representation of the specified resource. Requests using GET should only retrieve data.

POST

► The POST method submits an entity to the specified resource, often causing a change in state or side effects on the server.

DELETE

► The DELETE method deletes the specified resource.

PUT

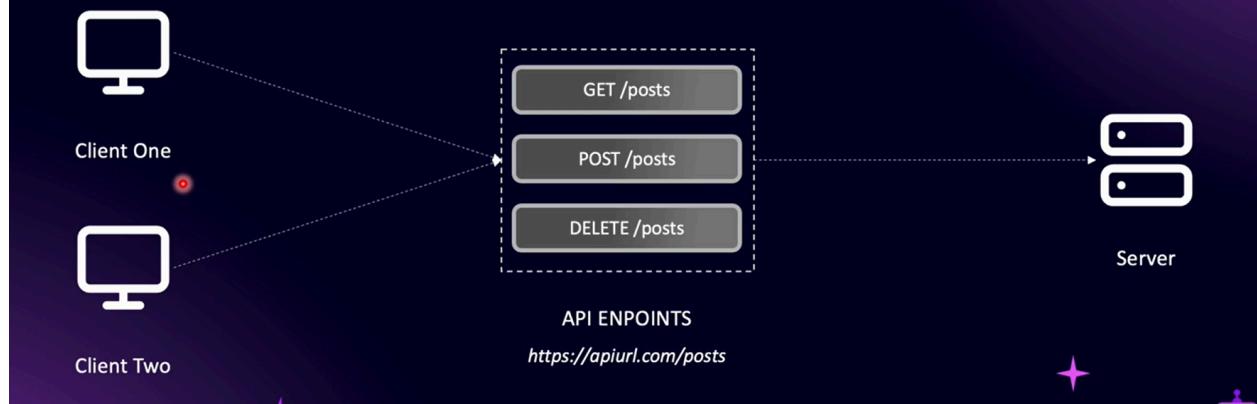
► The PUT method replaces all current representations of the target resource with the request payload.

PATCH

► The PATCH method applies partial modifications to a resource.

UNIFORM INTERFACE

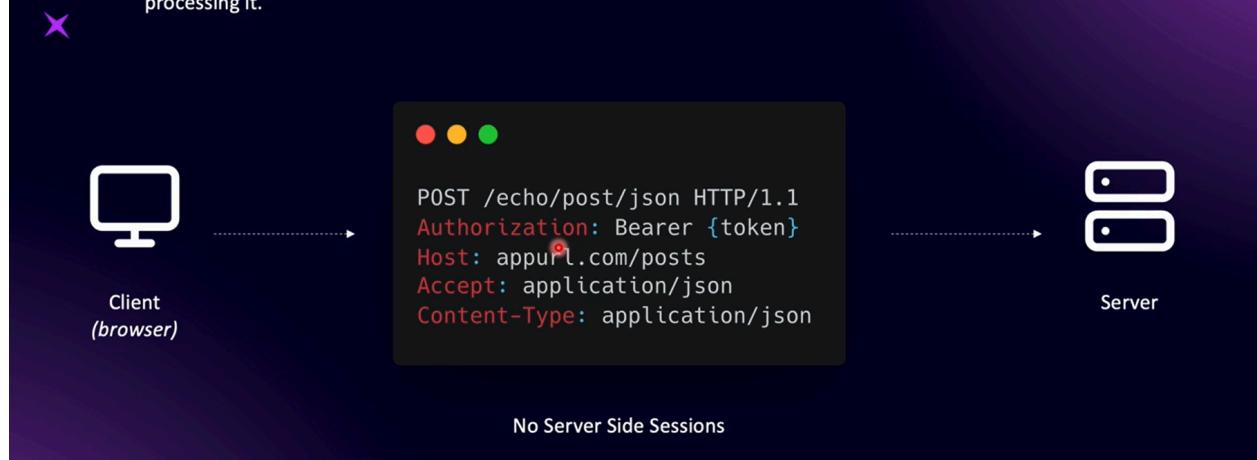
All API requests for the same resource should look the same, no matter where the request comes from.



3. What is a REST API

STATELESSNESS

REST APIs are stateless, meaning that each request needs to include all the information necessary for processing it.



CACHEABILITY

When possible, resources should be cacheable both on the client side and server side.

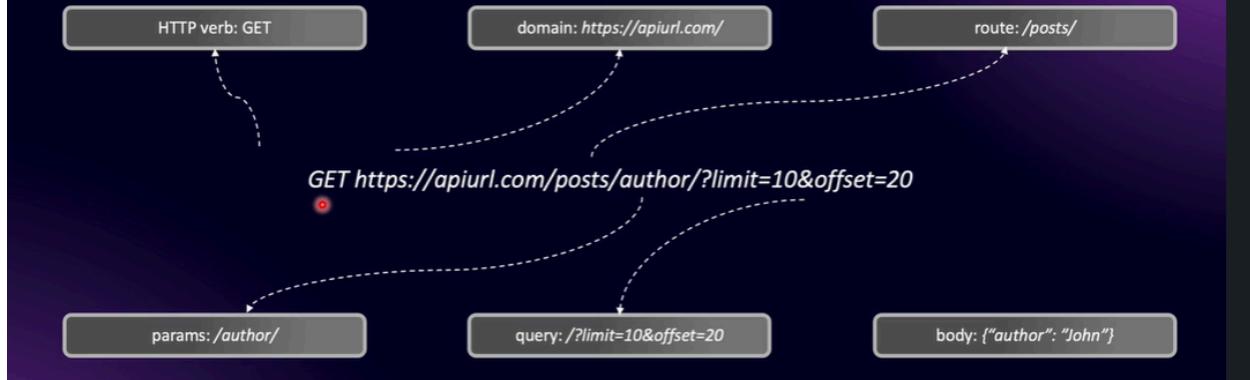


LAYERED ARCHITECTURE

REST APIs need to be designed so that neither the client nor the server can tell whether it communicates with the end application or an intermediary.



ANATOMY OF API ENDPOINT



PARAMS VS QUERY

GET https://apiurl.com/posts/author/?limit=10&offset=20

params: /author/

Params can be thought of as categories

query: /?limit=10&offset=20

And queries can be thought of as filters

EXAMPLE PROVIDERS

Many of the basic Nest classes may be treated as a provider – services, repositories, factories, helpers, and so on.



Users Controller
(routing only)

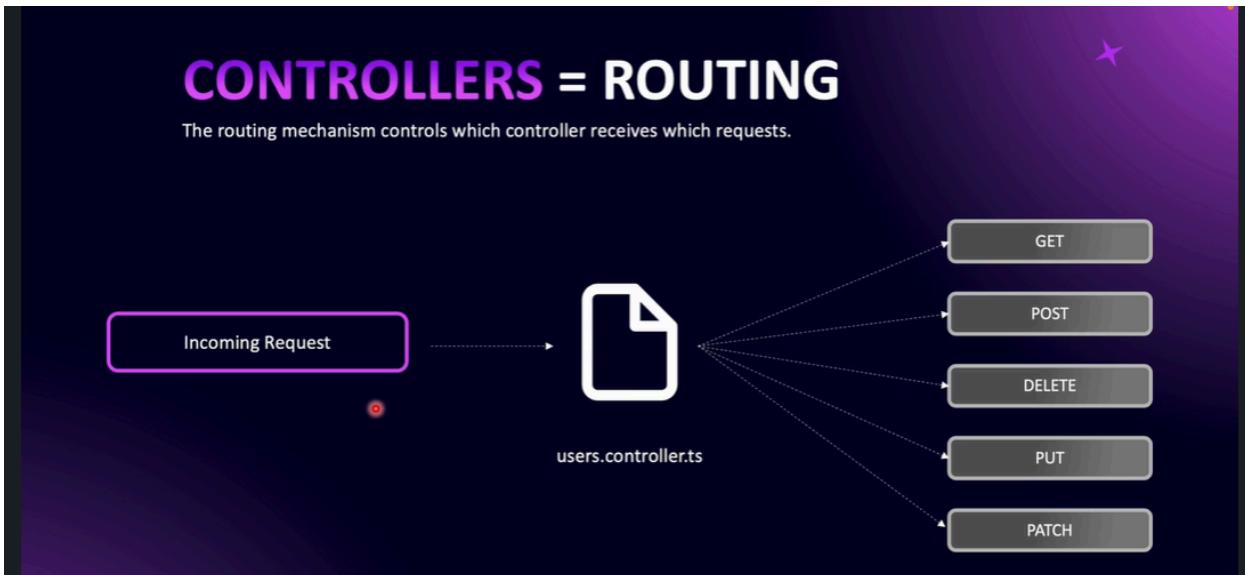
users-service.ts

users-check-blacklist.ts

users-check-garavatar.ts



Database



EXAMPLE PROVIDERS

Many of the basic Nest classes may be treated as a provider – services, repositories, factories, helpers, and so on.

