

Nest.js #1

- Es un framework inspirado en Angular.
- transpilar es pasar de un lenguaje a otro

Leer ([curso fernando herrera ude mi nest.js](#)).

<https://docs.nestjs.com/>

☐ En las 2 semanas vamos a leer toda la documentación.

cURL (hacerlo mediante eso es para alfas)

☐ api marvel, vamos a trabajar con esa api para estar obligados para trabajar con las api keys

Marvel Developer Portal

The Marvel developer portal gives Marvel fans, partners and other technologists access to an array of powerful APIs, documentation, and other tools to interact with Marvel's systems.

 <https://developer.marvel.com/>

Character {

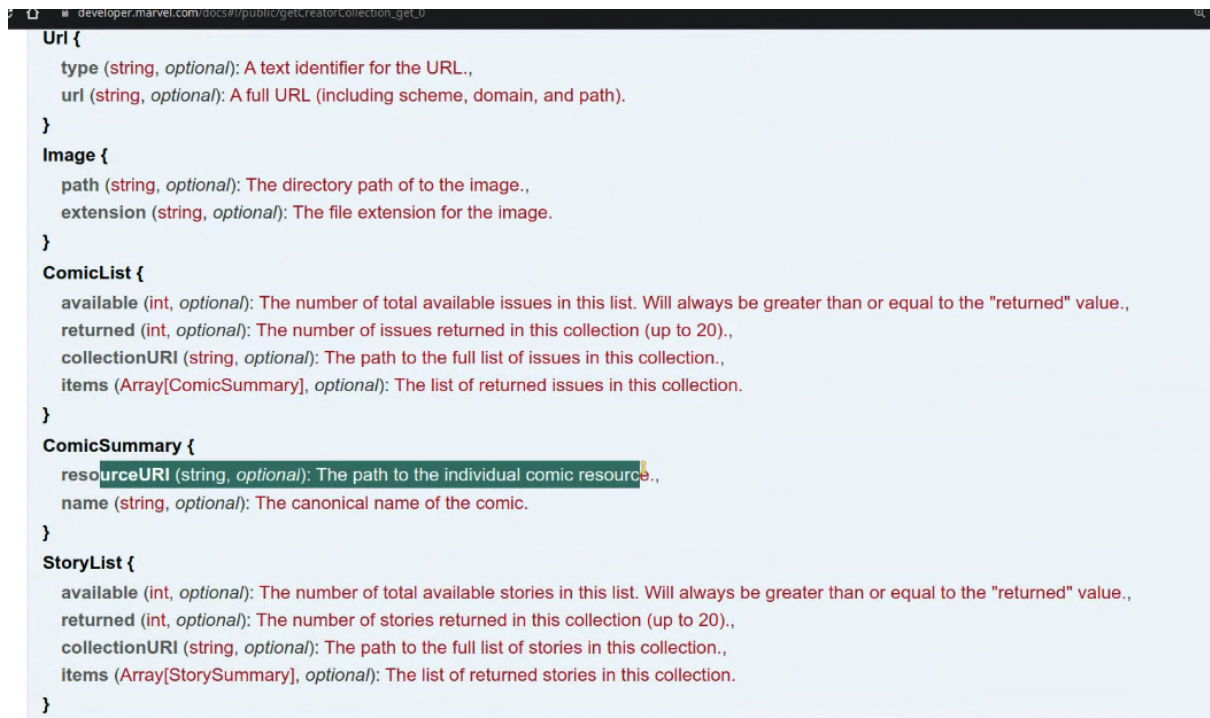
id (*int, optional*): The unique ID of the character resource.,
name (*string, optional*): The name of the character.,
description (*string, optional*): A short bio or description of the character.,
modified (*Date, optional*): The date the resource was most recently modified.,
resourceURI (*string, optional*): The canonical URL identifier for this resource.,
urls (*Array[Url], optional*): A set of public web site URLs for the resource.,
thumbnail (*Image, optional*): The representative image for this character.,
comics (*ComicList, optional*): A resource list containing comics which feature this character.,
stories (*StoryList, optional*): A resource list of stories in which this character appears.,
events (*EventList, optional*): A resource list of events in which this character appears.,
series (*SeriesList, optional*): A resource list of series in which this character appears.

}

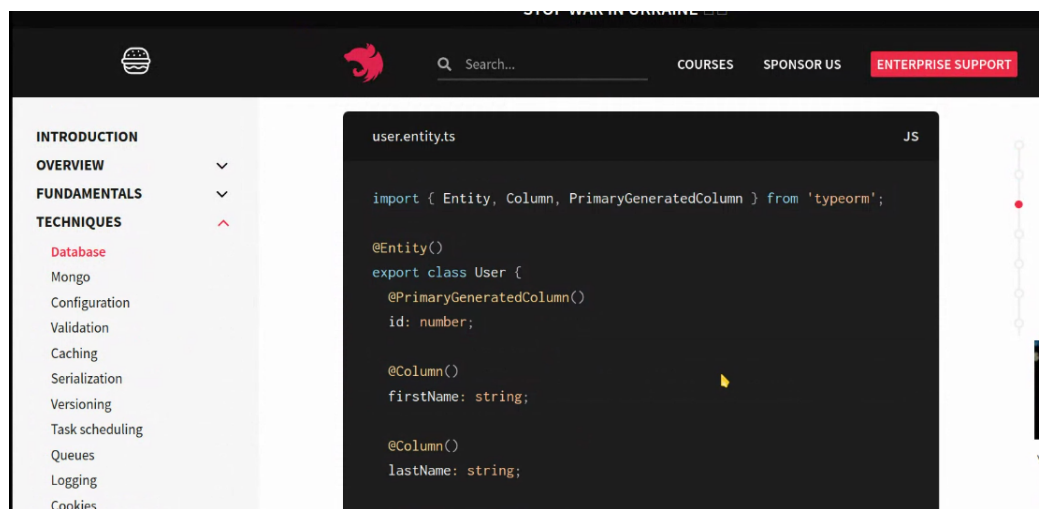
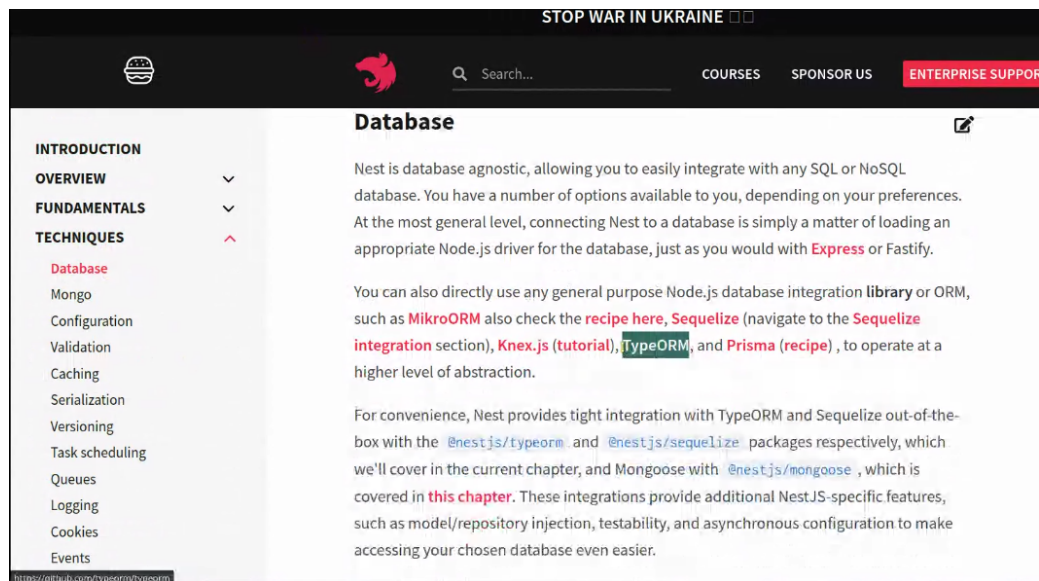
Url {

type (*string, optional*): A text identifier for the URL.,
url (*string, optional*): A full URL (including scheme, domain, and path).

}



- ☐ Crear la paginación y poder mostrar solamente información del personaje, foto, id, descripción, nombre
- ☐ hay que crear una base de datos llamada maestro detalle (factura, quien fue la persona que la compró, información de quién nos vendió y los productos, id),
- ☐ Necesitamos un sistema que pueda consultar la api de marvel heroes y que se pueda paginar, y que en la página pueda especificarse la cantidad de personajes que queremos visualizar. esto nos muestra la información del personaje.
- ☐ Existirá otro endpoint donde guardemos la información dentro de nuestra base de datos(con el id lo localizamos) comisc(tabla detalle)
- ☐ vamos a trabajar 2 base de datos una relacional MySQL, y otra noSQL(mongodb)
 - ☐ 3 tablas en total
 - ☐ personaje
 - ☐ comic
 - ☐ comic summary



- ☐ 3 end point
 - ☐ 1 consultas
 - ☐ y otras donde pasamos la id para hacer consultas en ambas base de datos
- ☐ EL UPDATE nos pedirá dos id, (nuestro sistema tiene que verificar si existe o no el personaje y si existiera tiene que avisarle al usuario que ya existe y de no existir se crearía el personaje)

INTRODUCTION

OVERVIEW

First steps

Controllers

Providers

Modules

Middleware

Exception filters

Pipes

Guards


Interceptors

Custom decorators

FUNDAMENTALS

Middleware

Middleware is a function which is called **before** the route handler. Middleware functions have access to the **request** and **response** objects, and the `next()` middleware function in the application's request-response cycle. The **next** middleware function is commonly denoted by a variable named `next`.



```
graph LR; ClientSide[Client Side] --> HTTPRequest[HTTP Request]; HTTPRequest --> Middleware[Middleware]; Middleware --> RouteHandler[Route Handler  
(Request Mapping)];
```

validadores

- ☐ la pagginación sea ver 1 o 5 o los que sean
- ☐ lo que haremos en nuestro sistema es pasarle 2 id, el id de nuestro sistema y el id del personaje
tendrán 2 endpoint para las dos base de datos y vamos a tener que validar su existencia
- ☐ el deleted será por la id del personaje (2 end point 2 base de datos sql y nosql)
- ☐ 2 tipos de borrado, lógico y físico - lo haremos físico que es literalmente es hacer un delete (va para las 2 base de datos)



12/10/22 DEADLINE