

Vistas con Nest JS

- Luis Ángel Serrano Catalá (A210490)
- Licenciatura en Ingeniería Desarrollo y Tecnologías de Software
- Taller de Desarrollo 4

Nest JS

NestJS en su núcleo, es una librería de enrutamiento HTTP para ambientes JavaScript, con potencial de irlo extendiendo para crear un framework completo.

Creando el patrón MVC

NestJS nos provee la funcionalidad de rutas por defecto, con sus controladores y un patrón de organización de servicios y módulos.

Esto cubre la "C" de Controlador, del modelo MVC. Que nos permite recibir y mandar respuestas al servidor de manera llana.

Para tener Vistas HTML, que son la manera en la que mostramos los datos a navegadores, necesitaremos configurar dos carpetas:

- **/views**: La carpeta donde almacenaremos nuestras vistas HTML para ser devueltas.
- **/public**: La carpeta donde almacenaremos cualquier otro archivo que necesite ser accedido por el cliente (CSS, Imágenes, etc).

Junto a esto, añadiremos un motor de plantillas llamado **hbs**, este nos permitirá pasar datos del controlador a la vista, y ejecutar lógica desde el BackEnd hacia el FrontEnd.

Esto se configura en el **src/main.ts**, nuestro punto de entrada de nuestra aplicación Nest:

```
async function bootstrap() {
  const app = await NestFactory.create<NestExpressApplication>(
    AppModule,
  );

  app.useStaticAssets(join(__dirname, '..', 'public'), { prefix: '/public'
}); // Asigna la dirección de la carpeta de recursos a `src/../public`

  app.setBaseViewsDir(join(__dirname, '..', 'views')); // Asigna la
dirección de la carpeta de vistas a `src/../views`
  app.setViewEngine('hbs'); // Asigna el motor de plantillas a hbs
```

```
    await app.listen(3000);  
  }  
  
  bootstrap();
```

Asignamos el motor de plantillas, pero no lo tenemos instalado aún, para ello corremos:

```
$ npm install hbs
```

Con esto, ya podremos crear nuestra vista.

Creando la ruta

Ya que tenemos listo lo necesario para construir la aplicación, empecemos a hacerlo.

Normalmente crearíamos un módulo primero y después un controller con `nest generate module` y `nest generate controller` respectivamente. Sin embargo, en este caso que solo queremos mostrar una vista, usaremos el controlador base `app.controller.ts`, que ya viene por defecto en un proyecto nuevo de Nest.

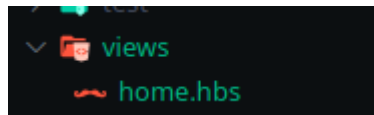
```
@Controller()  
export class AppController {  
  @Get()  
  @Render('home')  
  home() {  
  }  
}
```

En este caso vamos a crear una función con cuerpo vacío, ya que indicamos que lo que retornaremos es una vista indicándolo con el decorador `@Render`. Para esto también abremos creado una vista en la carpeta de `views/`.

Creando la vista

Ya que usamos el motor de plantillas `hbs`, nuestras plantillas deben ir con la extensión `.hbs`.

Por ejemplo `views/home.hbs`:



El nombre del archivo es lo que pasaremos al decorador `@Render` del que hablamos previamente.

Final

Codificando nuestro sitio en HTML desde `/views` e importando el CSS y JavaScript de `/public`, podremos empezar un sitio totalmente funcional.

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="utf-8" />
6   <title>Home</title>
7
8   <meta http-equiv="X-UA-Compatible" content="IE=edge">
9   <meta name="viewport" content="width=device-width, initial-scale=1">
10
11   <link href="/public/site.css" type="text/css" rel="stylesheet">
12 </head>
13
14 <body id="top" class="bg-beishe title-center title-h1h2 debug-screens">
15
16
17
18   <div x-data="hero">
19     <div class="flex isolate overflow-hidden flex-col min-w-screen"
20       x-bind="bindings">
21       <div class="flex inset-0 justify-center items-start w-screen h-screen">
22         
28         <button class="absolute bottom-8 font-serif italic cursor-pointer"
29           x-effect="showLogo && setTimeout(() => $el.classList.add('animate-pulse'), 1000);">
```

