**Instructivo para crear un módulo con angular-fullstack usando ES6**

1. Crear una nueva ruta dentro del proyecto, esto se hace con el fin del que generador cree los archivos necesarios para trabajar desde el cliente.El comando que se debe ingresar por consola para ejecutar sub-generador route es **[ruta proyecto] yo angular-fullstack:route [nombre ruta]**

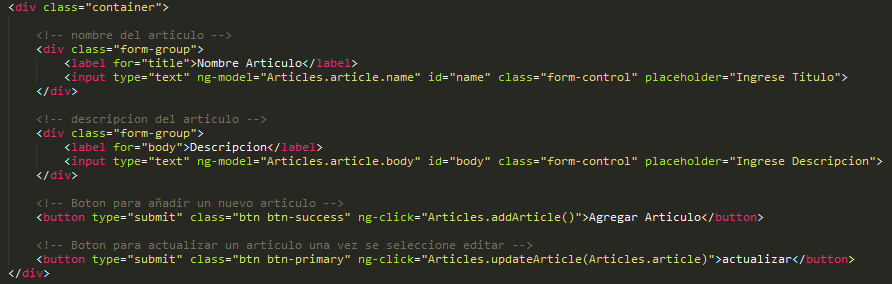
**Nota** el nombre de la ruta se ingresa en singular.

* **Ejemplo:**  c:\user\yording\ejemplo> yo angular-fullstack:route article
* **Archivos de salida:** 
  + **Client/app/article/article.controller.js=** Controlador angular para realizar la lógica del lado del cliente como eventos, peticiones al servidor, funcionalidades en el froend,etc.
  + **Client/app/article/article.controller.spec.js=** Pruebas unitarias para el articulo.controller.js.
  + **Client/app/article/article.css=** archivo css para el modulo.
  + **Client/app/article/article.html=** interfaz del módulo.
  + **Client/app/article/article.js=** Contiene toda la configuración del módulo, los estados de las rutas, los urls asociada al estados, permisos de autentificación para dicha rutas,controlador que los maneja,etc.

1. Definiremos un template para el modulo artículo, para ello nos dirigimos al módulo, que se encuentra en **Cliente/app/article/article.html,** también definiremosalgunos estilos css para nuestra interfaz **Cliente/app/article/article.css**

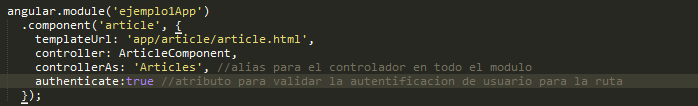
* En el archivo articulo.html borramos todo lo que se encuentra
* Agregarnos un formulario para registrar un artículo que contenga un nombre y una descripción y dos botones para registrar y actualizar (el botón actualizar no es bueno dejarlo ahí pero para fines didácticos se realiza el ejemplo).

Hasta ahora el código ha quedado así.

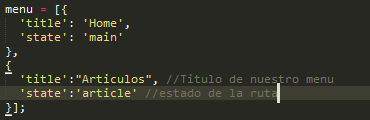


* Para que se conecte esta interfaz con nuestro controlador angular debemos crear un alias ‘Articles’ para el bind article que contendrá los datos desde el controlador, Para ello nos dirigimos a **Cliente/app/article/article.controller.js,** allí al component le agregamos un atributo llamado controllerAs:’Articles’ y otro llamado authenticate:true esto con el fin de evitar registrar artículos sin estar logueado, además para que se visualize artículos en nuestro navbar debemos agregarlo **Cliente/app/components/navbar/navbar.controller.js**  en array menu {title:’Articulos’,state:’article’}.

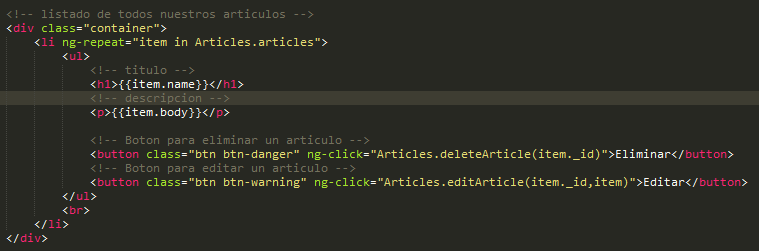
**Cliente/app/article/article.controller.js**



**Cliente/app/components/navbar/navbar.controller.js**

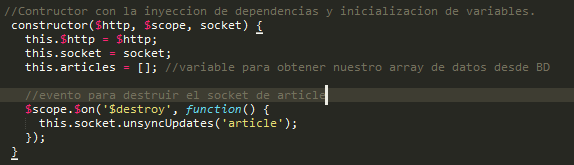


* Por ultimo agregamos a nuestro archivo.html una etiqueta para listar el array de datos de nuestros artículos.

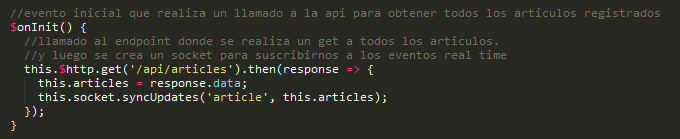


1. Teniendo terminada nuestra interfaz debemos ir a crear la lógica de negocio al article.controller.js.

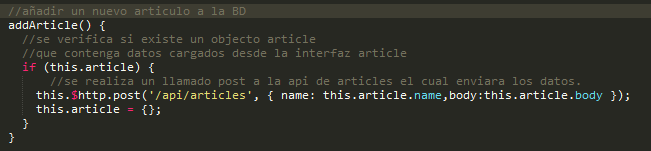
* Se debe crear un constructor el cual se le inyectara nuestras dependencias y además se inicializan variables.
* Se deben importar las dependencias para este ejemplo solo usaremos $http,$scope y socket.



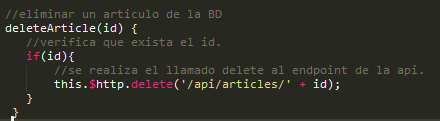
* creamos un evento inicial para obtener todos nuestros artículos de la base de datos y además añadirlos a un socket para cualquier cambio en la base de datos se vea reflejado inmediatamente en nuestra interfaz



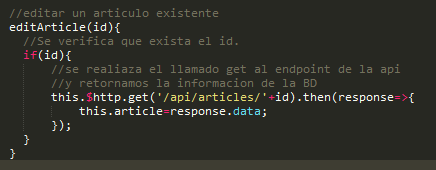
* creamos el evento pare registrar nuestros articles.



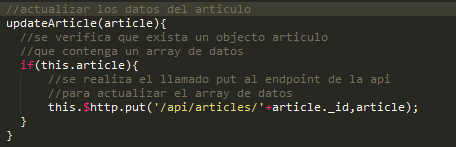
* creamos el evento delete pasándole como parámetro el id del article.



* creamos el evento editar pasando como parámetro id del article



* creamos el evento para actualizar enviamos id y le enviamos el array de datos



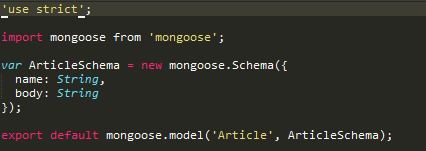
1. Ya teniendo todos los eventos en el controlador solo debemos crear un módulo artículo en el backend para ello utilizamos el comando del sub-generador endpoint es **[ruta proyecto] yo angular-fullstack:endpoint [nombre ruta]**

**Nota** el nombre de la ruta se ingresa en singular.

* **Ejemplo:**  c:\user\yording\ejemplo> yo angular-fullstack:endpoint article
* **Archivos de salida:** 
  + **Server/api/article/article.controller.js=**Controlador del lado del servidor para toda la lógica de negocioen backend
  + **Server/api/article/article.index.spec.js=** Pruebas unitarias para el articulo.controller.js. en el backend
  + **Server/api/article/index.js=** routeo de los endpoints.
  + **Server/api/article/article.events.js =** Son todos los eventos del modelo
  + **Server/api/article/article.model.js=** Esquemas de la base de datos en los modelos.
  + **Server/api/article/article.socket.js=** Broadcast updates para el cliente cuando en el modelo cambia.
  + **Server/api/article/article.integration.js**

1. Crear el esquema en Mongoose en **Server/api/article/article.model.js**.

* Se crea un esquema con nombre y descripción los nombres deben ser tal cual se nombraron en la interfaz article.html



Ya con eso tenemos un módulo funcionando en angular-fullstack