Mini-Front

Versión de angular: 14.0 **Versión Bootstrap:** 5.3.0

Descargar versión especifica de node:

- nvm install 16.10.0

Validar versión:

- node -v
- npm -v

Instalar dependencias:

- npm install

Inicializar proyecto:

- ng serve

Variables de entorno:

Las variables de entorno se podrán establecen en el archivo “environments.ts”, para llegar a este se accede a la carpeta “src/environments/”. Aquí se podrán configurar las variables de las apis.

Modelos:

La creación de los modelos la encontraremos en la siguiente carpeta “src/app/models”, para crear un modelo se utilizara el siguiente comando:

```
ng generate class models/“nombre del modelo”
```

De esta forma se creara una clase en la cual podremos establecer los atributos que contendrá nuestro modelo, este es un ejemplo de como puede lucir el modelo:

```
export class ServiceObject {
  constructor(public entity?: string, public id?: number, public data?: any, public attributes?: any) {
    this.entity = entity;
    this.id = id;
    this.data = data;
    this.attributes = attributes;
  }
}
```

Servicios:

- **Generales**:

  Para los servicios tenemos el archivo “api.service.ts” que se encontrara en “src/app/service/”, en este podremos encontrar cinco métodos generales, veremos el

método GetAction, PostAction, PutAction, PatchAction, DeleteAction, para cada uno de estos se debe de enviar un “serviceObject”, el cual es una clase con los siguientes atributos:

```
entity
id
data
attributes
```

Dependiendo el método se enviará uno u otro a excepción del entity, está siempre se enviará puesto que es la entidad que identificara la ruta a consumir.

- **Específicos:**

  En el proyecto se crean distintos servicios que servirán específicamente para nuestros componentes, esto se realizara en la carpeta “business-controller”, ubicada en “src/app/” en ella crearemos un servicio con el siguiente comando:

  ```
  ng generate service business-controller/“nombre del servicio”
  ```

  Después de crear el servicio procedemos a importar los siguientes elementos:

  ```
  import { ApiService } from '../service/api.service';
  import { ServiceObject } from '../models/main/service-object';
  constructor(private apiService: ApiService) {}
  ```

  Una vez importado lo siguiente es crear los métodos para llamar a ese servicio antes creado y consumirlo, este es un ejemplo de cómo se podría realizar este método:

  ```
  getComic(params: {}): Promise<Comics[]> {
     let servObj = new ServiceObject('v1/public/comics');
     return this.apiService.GetAction(servObj, params)
       .then(x => {
         servObj = <ServiceObject>x;
         this.comics = <Comics[]>servObj.data.results
         return Promise.resolve(this.comics);
       })
       .catch(x => {
         throw x.message;
       });
   }
  ```

  En este caso este método recibirá unos parámetros para posteriormente poder enviarlos, creamos un “ServiceObject”, que sería el modelo antes creado y en el enviamos con la entidad, llamamos a nuestro servicio enviándole los parámetros necesarios para después retornar una procesa con un modelo especifico que ya antes previamente creamos, dependiendo de lo que queremos consumir el método cambiara, este es solo un breve ejemplo.

Componentes:

Para crear un componente nos dirigimos a la carpeta “src/app/view”, en esta carpeta tendremos un “routing.module” que es en donde se ubicaran las primeras rutas de nuestro proyecto, estas rutas se crean a base de los componentes que estaremos creando a medida que realicemos el proyecto, para crear estos componentes utilizaremos los siguientes comandos:

ng generate module view/“nombre del componente” –routing

ng generate component view/“nombre del componente” --module view

Esto nos creara un modulo independiente con sus propias rutas junto con, estas también dependerán de los componentes que creemos a partir de este, para ello utilizaremos el siguiente comando:

ng generate component view/“nombre del componente”/”nombre subcomponente” --module example

De esta forma se creará un sub-componente que trabaja con el modulo independiente antes creado