

**Proyecto:** ACME Explorer (backend)

**Grupo:** ACME-Sandwich

**Autores:** Antonio Rodríguez Artacho, Jorge Gordo Aguilar y Rafael Fresno Aranda.

**Fecha:** 02-03-2020

Como A+ para la segunda entrega de la asignatura el día 2 de marzo de 2020, se ha implementado un sistema para la generación automática de datos mock a partir de una librería de npm propuesta por el profesor Carlos Müller: `mock-data-generator` (<https://www.npmjs.com/package/mock-data-generator/v/2.6.6>). Lo interesante de esta librería, que la diferencia de la herramienta vista en clase, es que en esta se pueden establecer relaciones entre los distintos objetos generados de forma automática, para que la inserción en la base de datos sea coherente sin necesidad de realizar relaciones manualmente mediante claves primarias y foráneas.

Para usar esta herramienta, es necesario definir un objeto JSON por cada modelo que tengamos en nuestro sistema. Estos objetos JSON tienen un atributo por cada atributo real del modelo. Gracias a bibliotecas como `faker.js` se pueden obtener datos aleatorios de texto, fechas entre rangos, números entre rangos, etc. En esta página, correspondiente con la documentación de la librería, se muestran todas las opciones de configuración que se admiten para la generación de los objetos: <https://github.com/danibram/mock-data-generator#readme>.

Lo más característico es la relación entre los distintos tipos de objetos; si tenemos una variable que moldea a los actores y otra a los trips, y queremos que un trip tenga un atributo de tipo `ObjectId` que se corresponda al de un actor, su creador (que debe ser un manager), sólo bastaría con crear una variable que configure las opciones de los managers, y a la hora de crear la configuración de los trips, bastaría con escribir lo siguiente sobre su atributo `creator` (un manager): `hasOne: 'managers', get: 'id'`. Esto indica que en este atributo se va a guardar la id aleatoria de un objeto de la variable managers.

Cuando los objetos y sus relaciones estén moldeados, la librería requiere que se escriba explícitamente la forma y el número de elementos a generar de cada tipo del modelo; esto se hace llamando a la biblioteca con `'mock()'` y posteriormente escribir `'.schema('trips', trip, 50)'`, donde el primer elemento es el nombre que le queremos dar a la colección a generar por la librería, el segundo la variable que almacena la configuración del modelo de la biblioteca y el tercer el número de objetos a generar en total.

Finalmente, cuando todos han sido generados, mediante bucles se recorren cada uno de ellos y se guardan en la base de datos llamando simplemente al método `.save` del modelo en cuestión.

Toda esta funcionalidad está implementada en el fichero `'api/controllers/mockDataGenerator.js'` del proyecto, y es accesible realizando una petición POST a la dirección `'/v1/mock-data-generator'`.