

Watch: 🎭 **Agilizar la creación de datos fake con Test Object Factories**

11:39 |



Puedes encontrar el código de este vídeo en [GitHub](#).

Hasta ahora hemos ido *hardcodeando* todos los datos que queríamos pasar a nuestros tests (props, valor que devuelven los mocks, etc). Esto es poco práctico, además que da poca robustez a nuestros tests, ya que siempre estamos testeando con los mismos valores, y si cambia la forma de cierto objeto tendremos que cambiarlo en todos nuestros tests.

Para agilizar nuestro proceso de testing y dar más robustez a nuestros tests con datos aleatorios, podemos hacer uso de librerías de test object factory como [Fishery](#) o [Rosie](#), y librerías de datos fake como [Faker](#).

Las librerías de factories nos ofrecen funciones para generar objetos con determinadas keys y valores:

```
import { Factory } from "fishery";
import { date, lorem } from "faker";

const commentFactory = Factory.define(({ sequence }) => ({
  id: sequence,
  date: date.recent().toISOString(),
  content: lorem.paragraph(),
}));
```

Podemos exportar la factory directamente para hacer uso de sus métodos, o bien exportar funciones propias que llamen los métodos de la librería internamente, por ejemplo en el caso que nos interese crear una lista de comentarios con un número de items aleatorio:

```
export function generateCommentList(min = 0, max = 10) {
  const length = Math.random() * (max - min) + min;
  return commentFactory.buildList(length);
}
```

De esta forma podremos escribir tests más ágilmente, serán más fáciles de leer y nos darán más confianza.