

¿Cómo llevar a la práctica todos los conocimientos que vimos con respecto a "JSON" "objetos" "promesas"? La mejor manera es consumir una API, los datos de una API, de una API, que van a venir hacia nosotros en formato JSON, tenemos que ir a buscarlos, tenemos que buscarlos de una manera utilizando "promesas", porque eso va a demorar un poco, cuando lleguen tenemos que convertirlos, porque vienen en formato JSON. Tenemos mucho para hacer pero,afortunadamente, ya hemos visto en clases pasadas cómo hacerlo. Ahora vamos a poner toda esa información en contexto, en una sola pieza de código y lo resolvemos para terminar así lo que sería nuestra unidad de Javascript.

Muy bien, aquí a la izquierda pueden ver una API, es un servicio muy conocido, muy utilizado de JSON, "place holder" que entrega distintos "posts", o sea, distintos puntos de acceso a recursos en formato JSON.

Esto es muy útil para utilizar cuando estamos diseñando algún sistema y queremos interactuar con una API y de tipo "rest" para obtener información. Bien, aquí tenemos, por ejemplo, un "end point" con posteos o "posts" de usuarios. Tenemos 100 recursos, comentarios, albums, etcétera. Vamos a usar el "end point" y "users" que nos da información sobre 10 usuarios para ver cuáles, entonces, que cuando hacemos, accedemos a este "end point" tenemos un formato JSON, es una suerte de registros, de base de datos en formato JSON. Okay, ¿cómo podemos trabajar esta información en nuestro, en nuestro sistema, en nuestro proyecto? Aquí tengo el mismo ejemplo pero que venimos utilizando hasta ahora y he agregado solamente una etiqueta semántica "section" y aquí dentro es donde vamos los datos que regresará desde esa API. Muy bien, vamos a ver cómo podríamos trabajar en este sentido, vamos a utilizar la interfaz "fetch". Ahora que ya han visto cómo funcionan las "promesas", recordarán en la clase pasada construimos un método "get data", que nos retornaba a una "promesa" y simulaba la búsqueda de información en una API y bueno ahora hacemos exactamente eso pero en el mundo real, no necesitamos construir una "get data", una función, porque "fetch" hace eso ya por naturaleza, por defecto lo hace, trabaja de manera asíncronica, por lo tanto recuerden, el primer parámetro aquí sería indicar la dirección, el "end point" de donde queremos extraer la información, es esta, bien, vamos a copiarlo y lo pegaremos aquí entre comillas muy bien a continuación recordando esto retorna una "promesa" por lo tanto debemos indicar "then", okay, para hacer algo cuando la "promesa" nos cuando "fetch" nos dé una primera respuesta bien entonces diremos que cuando está la respuesta aquí el nombre es convencional, podríamos ponerle "response", respuesta, o cualquier otro nombre. Bien, vamos a convertir esa respuesta, esa respuesta, entonces, que viene como está vendiéndola a la izquierda, en formato JSON, vamos a convertirla en un objeto, porque no nos serviría, no nos serviría de nada, de lo contrario, ¿verdad?, como ya hemos visto en la clase, en las clases pasadas. Entonces, a esta respuesta la convertiremos al formato JSON, ¿okay? Eso está hecho. Ahora, este procedimiento también tiene una leve demora, por lo tanto vamos otra "promesa" y aquí, concurrenciamos, por lo tanto, nuestra segunda "promesa" Aquí, "si", en esta instancia, ya deberíamos contar con nuestros datos, vamos a ponerle "data" y lo que hacemos con esa información, con esa "data", bueno, primero vamos a mostrarlo, solamente para que ustedes vean si esto funciona y aquí entonces como está está corriendo de inmediato, si abrimos la consola, deberíamos tener nuestra información, ¿correcto?, viene la información pero, ¿sí? que ya la hemos pensado