

Bien, este es un video que preparé como una demo para que ustedes puedan ver cómo se utiliza POSTMAN. Así que bueno, vamos ahora mismo a ver el video: Esto que están viendo acá es un JSON, esto es un JSON. Entonces acá tienen, ¿ven?, los campos, esto es un JSON de verdad, digamos, de producción. Acá tienen los nombres de los campos, ¿sí? Y ven, de repente aparece “datos de formulario” y adentro tiene más cosas, adentro tiene “renovación de nichos”. “Renovación de nichos” adentro tiene más cosas, ¿sí? Tienen un “Label” y adentro tiene un “Valor”, y “Valor” tiene adentro más cosas, bueno, y así. Es como que se va anidando, ¿no?

Y acá aparece un “ítem”, tiene un “Array”, y adentro, ¿sí?, tiene una estructura. Bueno, en realidad, si lo leen, digamos, se entiende, se entiende sobre todo si está acá este en Visual Studio Code. Si ustedes tienen un archivo “.JSON”, se los acomoda de una forma legible.

Si ustedes lo van a querer leer como si fuera un archivo de texto, y bueno, se les va a complicar un poco más.

Visual Studio Code es una herramienta que utilizamos para hacer desarrollo del software, entre otras cosas, también sirve para leer un JSON. Se la pueden descargar, la pueden instalar, es un programa liviano que tiene mucha funcionalidad, está bastante bien, pero en definitiva, es como un editor de texto pero para desarrolladores, bastante completa.

Porque no va a estar así, intentado. Pero bueno, nada, yo lo que quiero que sepan es que existe esto que se llama JSON, que es, digamos, un archivo que tiene texto adentro y que es el que sirve de intercambio para pasar la información a un este... a un servicio web. Y es también, digamos, la forma en que se va a recibir información de una API REST. Llamemos API REST. Servicio web a veces maneja también otros otro tipo de archivo, pero es más común usar API REST con JSON. Es lo más común.

Bueno, a ver. Y acá está la herramienta que se llama “POSTMAN”, esta es una herramienta llamada “POSTMAN”. Entonces “POSTMAN” les va a servir para testear web services. Si a ustedes le dicen tienen que testear un web service, lo que van a hacer es abrir “POSTMAN” y testearlo desde “POSTMAN”, o esa es una de las opciones, ¿no?, también puede haber otros programas, pero es lo más común.

Entonces, yo voy a ejecutar acá, tengo un servicio, hecho en Python, le voy a poner “ejecutar en mi máquina”, acá ejecuto y me dice que está corriendo en el puerto. Perdñon.

Esto es un servicio web, eso que están viendo ahí, es el código de un servicio web que tengo yo en mi máquina, que lo tengo levantado en mi computadora como servidor y lo estamos usando para este ejemplo.

8.501. Esto, les cuento, yo tengo, ustedes no tiene por qué saberlo, pero sí tienen que saber que me está corriendo un servicio en el puerto 8501 y algunas características le tienen que haber dado, porque sino ustedes cómo hacen para testear y le tienen que decir. Entonces, a ver, acá tengo, cuesta, fíjense tengo una IP, yo le dije que está corriendo en “local host”,

“local host” es mi máquina, ¿sí?, 8501 es el puerto y clientes, se llama, llamémosle al método que yo quiero invocar, que está dentro del servicio. O sea, está ésta API y tiene publicadas un montón de “end points”, ¿sí?, un montón de funcionalidades. Yo quiero llamarla “funcionalidad de clientes”. Entonces, yo pongo acá que voy a llamar a “funcionalidad de clientes”. Y tengo para elegir qué tipo de verbo voy a utilizar de “http”. En este caso, voy a usar el “GET”, ¿por qué el GET? Y, porque yo quiero traer información, ¿sí?, “clientes” es algo que me va a devolver a mí información acerca de los clientes que están en la base de datos, ¿está bien?.

Entonces, este... esto es un servicio, esta es una API que tiene una “end point” que se llama “clientes”, que me va a devolver el contenido de los clientes de la base de datos. No me devuelve todo el contenido, si no lo devuelve algunos campos, nada más, pero bueno, esa es la idea, ¿sí?

Entonces, si yo elijo acá, elijo acá, digamos, pongo la dirección correcta y pongo el verbo correcto. Pongo “SEND”, me dice que estaba buscando la respuesta y acá me traen la información. Voy a correr esto que me está molestando.

Este... acá me está trayendo la información que está adentro de la base de de datos y cuando devuelve la información, la devuelve en forma de JSON fíjense, abre, abre llave, cierra llave, abre llaves, cierra llaves, tiene clave, valor, clave, valor, me está devolviendo un JSON, ¿está bien?

Me está devolviendo un JSON, porque a parte yo acá le puse JSON, porque acá se puede elegir poder verlo de distinta manera, pero bueno, este... esta API devuelve JSON, y yo elijo acá JSON para poder verlo de una forma en formato JSON. Si no le pongo esa opción, por ahí no lo veo tan, tan claro.

Bien, esto sucede porque yo levanté el servicio web, ¿sí? O sea, puse en ejecución esta API REST y entonces, encuentra esto, ¿sí? y me devuelve la información y me dice “Status: 200”, ¿está bien?, estatus 200 era, ¿se acuerdan?, el que era el exitoso, estaba todo bien.

Si yo detengo esto, acá detuve el servicio y voy a volver a ejecutar, no lo encuentra. No encuentra algún error, no lo encuentra, entonces, voy a volver a ponerlo en ejecución.

Bien, entonces, ustedes quieren testear este servicio, está API, digamos, este “End point”, se llama “End point”, ¿no?, el esta dirección, digamos, quiero testear éste “End point”, de esta API que tengo funcionando, bueno, tengo que saber cómo invocarla, me tienen que decir “La tenés que invocar de esta manera” y me tienen que decir el método que tengo que usar, el verbo que tengo que usar, “Usá el verbo GET”, invocalo de esta manera, ¿sí?, esos son datos.

Entonces, ustedes si no les pasan los datos, no lo pueden testear. Si le pasan los datos, listo, prueban. Prueban, a ver... ¿funciona?, sí, perfecto, funciona. Listo. Esto está funcionando.

Ahora tienen que ver después si la información que vuelve es correcta, si la información que vuelve es no es correcta, ahí ya se hace otra cosa.

La idea es que, por lo que yo les comenté, esto devuelve el nombre y la fecha de última visita a cada uno de los clientes de la empresa, por ejemplo, ¿sí?. Está devolviendo todos los clientes, está devolviendo sólo un grupo de clientes, ¿qué está devolviendo?, ¿los clientes de una zona?, bueno, eso lo tendrán que ver, si funciona o no funciona. ¿Cómo pueden saberlo ustedes? Ustedes ven que está trayendo información en el servicio, pero después, no saben si están funcionando, como correctamente, tal como lo indica, ¿sí?, la especificación.

Entonces, lo que tienen que hacer acá van a tener que entrar a la base de datos y verificar, hacer una consulta ustedes en la base de datos y verificar si lo que vuelve del web service, es lo mismo que tendría que volver, ¿está bien?.

Acá les puse otro otro “End Point”, esto es de un sistema de fidelización, ven que dice “fide cliente”, “fide punto”, es un sistema que, de fidelización que tuvimos que hacer, para una heladería. Entonces, esta heladería lo que quiere es entregarle puntos con cada compra a sus... a sus clientes.

Entonces, este... la idea es que, yo quiero, que en la base de datos estén los puntos que tiene cada clientes, ¿sí? Porque, ¿qué pasa?, tiene una tabla en la base de datos, donde va, cada vez que el cliente compra, va agregando una línea, entonces, dice bueno, “compró por 10 puntos”, acá “compró por 50 puntos”, acá “compró por 30 puntos”, y va sumando. Entonces, si yo quiero saber, de un cliente en particular, cuántos puntos tiene, ya hay desarrollado dentro de la de la API, un “End Point”, que se llama “GET puntos parámetro”, ¿sí? y que acepta como... como parámetro de entrada, “parámetro” es un valor que yo le voy a pasar, que el cliente sea igual a uno, o sea, yo les puedo indicar los puntos de qué cliente quiero ver, ¿sí? Fíjense en el caso anterior, cuando fuimos a clientes me devolvió, yo no le pasé ningún parámetro, le dije “dame los datos de todos los clientes”, pero en este ejemplo, lo que estamos haciendo, es ver los datos de un cliente en particular. El cliente número 1 son, no son, digamos, los mismos datos que en el anterior. En el anterior devolvía el nombre y la fecha de visita. Acá está devolviendo los puntos que tiene acumulado ese cliente. Y entonces, yo le tengo que pasar quién es el cliente, sino no tiene sentido, me devuelve los puntos de todos. Y fíjense que cuando uno le pone acá que el cliente se igual a uno, o sea, se agrega solito, en esta solapita que dice “parámetros”, un parámetro que es “cliente igual a uno”.

Si ustedes lo tipean, acá se va a agregar acá, y así. Le pongo enviar. Bien, calcula y me devuelve un JSON donde me dice la cantidad de puntos de ese cliente, ¿está bien?. Y otra vez tenemos un estatus de 200, el tiempo que tardó.

Bien.Eso.

Entonces, vimos cómo hacer la consulta de manera que traiga datos sin un parámetro y cómo hacer las consultas de manera que traigan datos con un parámetro.

Lo que pasa es que, estamos otra vez usando el “GET”, ¿no es cierto? Fíjense la forma en la que, en la que se está llamando al servicio web, que se le está poniendo en la misma ruta

donde se llama, se está poniendo el código del cliente que quiero ver y esto se podría escribir en un... en un browser. Se podría escribir en un browser. y funcionaría también. Yo puedo ir al Chrome, escribir esto y funcionaría también. Ahora, el tema es que se hace visible, dentro de la ruta, el parámetro que está pasando de entrada y a veces yo no quiero que sea visible ese parámetro. Yo no quiero que el usuario vea qué estoy filtrando, ¿sí?, el cliente uno, ¿está bien? Esto es muy común, que es que haya parámetros en la URL, es muy común, pero quizás yo no quiero que sea así. Yo no quiero que sea así, que se vea eso. Lo que puedo hacer acá, está programado de otra forma, tiene una barrita y un "uno" y esto, como que se puede esconder de otra manera. No hay... no hay parámetros que aparezcan en la URL, entonces, hace lo mismo pero está, digamos, pasandose parámetros de otra forma.

Igual acá yo les voy a contar algo, acá estoy usando el verbo "GET", ¿sí?, hay una convención tal que para, pasar este... digamos, para hacer consultas, para hacer consultas en un servicio, un servicio web, se usa el verbo "GET", pero yo podría usar el verbo "post" o el "put", si fuera así, oculto de mejor forma los parámetros que le paso al servicio.

De hecho, no se ven, los parámetros que le paso, si le paso un "post" no se ven, si lo paso un "GET", sí se ven, si le paso a un "post", no se ven. Entonces, todo lo que es más seguro, paso de contraseñas y ese tipo de cosas, no se hace con los verbos "GET", sono con los verbos "POST" o "PUT" o alguno de esos.

Bien, hasta acá, entonces, hicimos tres consultas básicas, ahora el tema es qué... ¿qué pasa? Esto no es lo normal, ¿sí? Lo normal es que ustedes tengan que validar contra un "Token", ¿sí? Pedir un "Token" de validación y validar el "Token" para poder finalmente acceder a ver los puntos. Es decir, que haya un método de seguridad. Entonces, yo les comentaba que, por ejemplo, acá tengo otra llamada a este "end point", el "local host" que es mi máquina, 8.501 que es el puerto y "Ptos" se llama el servicio al cual yo voy invocar.

Bien, entonces, yo lo voy a invocar y hago "SEND" y me devuelve algo que está dentro del código de programación, para que devuelva siempre un JSON, yo lo que le puse es que si puede devolver correctamente el resultado, que lo devuelva, y si no puede devolverlo porque su sucedió un error o algo, que devuelva a un "Status NO", porque a mí se me ocurrió ponerle así. Podría haber sido cualquier otra cosa.

Entonces, devolvió que no, o sea que no está pudiendo acceder, pero no puede acceder porque este, en particular, este "end point" está solicitando un "Token". Entonces, ustedes no pueden ir a accederlo, sin pasarle un t"Token" y ese "Token" tiene que ser un "Token" válido. No puede ser cualquier cosa. Entonces, ese "Token", hay otro "end point" que lo genera. Vamos a llamar a esto, a este "end point" que es el de "Generar Token". Esto es el que genera el "Token". Le pongo "send" y acá acabo de recibir un "Token", que lo voy a copiar.

Entonces, normalmente no es el caso porque yo no lo hice así a esto, porque esto es lo que arme para una prueba, pero normalmente este "Token" puede ser bastante más largo y además mtiene un tiempo de duración. No es algo que sea permanente, o sea, vence, no sé, en 20 minutos, cada 20 minutos vence, por ejemplo, ¿sí?, o cada quince minutos vence,

entonces, digamos, en el caso de que venza ,hay que pedir otro, otro “end point”, pero para otro “Token”, hay que volver a llamar al “end point” y pedir otro “Token”, para poder ingresar.

Porque cuando quieran volver a ingresar con este mismo ,les va a decir que venció, ¿si?, entonces que no, que no sirve. Bueno, en este caso no vence yo no hice que venza. Entonces, copie ya el “Token” y ahora vuelvo acá, a puntos y les lo voy a llamar, pero esta vez le voy a pasar, le voy a pasar el “Token”.

Entonces, ¿ven que acá tengo el JSON que yo mando como parámetro? A este... a este “end point”... Yo mando este JSON, quiero saber los puntos del código uno, le mando una semilla, para general “Token” y le mando el “Token”. Entonces, con eso trabaja la API, va a descifrar, digamos, este “Token” va a fijarse que la semilla sea correcta y, en ese caso, va a devolver los puntos del código, del código uno y del cliente número uno, lo mando y acá, me está devolviendo, me devuelve que “estatus okay” y me está devolviendo los puntos, ¿si?, los puntos correctos, ¿está bien?, ahí pudo hacer la consulta.

Entonces, de esta forma es como se, de esta forma, digamos, a ver, esto tiene más complejidad, ¿está bien?, estamos viendo algo con lo que ya pueden trabajar. O sea, sabiendo esto, ya pueden trabajar. Ya pueden hacer cosas. Hay algunas veces que, bueno, se complica un poco más.

Fíjense que yo utilicé “PUT”, ¿si?, si le ponía otra cosa, otro verbo, iba a dar “error”, pero bueno, esto tienen que saberlo ustedes, con qué verbo está trabajando. También acá hay, a veces, otros métodos de autorización para pasar “Token”.

Entonces, si a ustedes les dicen “bueno, el token que vamos a pasar, no es un token común como el que hicimos recién, es un token de autos”, entonces, vienen acá y el “Token” lo escriben en esta, en este lugar, donde tengo el puesto el cursor. Si te dicen “es un historial del token”, bueno, el “Token” se escribe acá. Fíjense, me quedo guardado el último “Token” que usé, ¿no?, fíjense a ver si les puedo mostrar bien, tocando mal el botón. Les quería mostrar el largo, pero es bastante... bastante largo.

Acá sigue, ahí terminó, pero bueno, es un “Token”... claro, hay más largos que éste, hay más grandes todavía.

Así que, bueno, a veces, por eso la autenticación se configura desde acá y no se pasa dentro dentro del JSON como lo pasé yo, pero esto depende de la forma como esté programado el servicio, ¿está bien? Yo lo pasé en el “body”, acá donde paso, el “body” es el lugar en donde pasamos nuestro JSON, pero a veces no hay que pasarlo en el “body”, hay que ir a autorización y pasarlo desde ahí, pero bueno, depende de la forma que trabaje el servicio. Entonces, por eso, todo esto se lo tienen que decir, ustedes no lo pueden adivinar, les tienen que decir cómo es, pero una vez que ustedes saben cómo probarlo, bueno, de esta forma se pueden probar los servicios, los servicios web, las API’s, se pueden probar desde acá, utilizando POSTMAN y utilizando JSON. Pero para eso tienen que saber la sintaxis del JSON y su personalidad, vimos la clase pasada y es muy sencilla. Tiene lo que sería la clave, dos puntos y el valor, una coma, la clave, dos puntos y el valor.

Hay estructuras más complejas también, vimos en una clase pasada que hay estructuras más complejas, donde se pueden ir anidando adentro de esa clave, dos puntos y el valor, en vez de ser un valor entre comillas, que es, quizás, abre otra llave y tiene adentro otro conjunto de clave, valor. Bueno, se puede complejizar un poco más. Pero sigue siendo legible y si ustedes tienen una herramienta como esta, como el "Visual Studio Code", cuando le ponen el JSON dentro de "Visual Studio Code", el "Visual Studio Code" se los ordena, se los muestra de una forma que sea legible, que es lo que estamos viendo acá.

Bueno, no es lo único. No es la única herramienta. También hay herramientas online, que ustedes pueden poner un JSON y se los ordena y se los muestra ordenados para que, digamos, no se pierdan, porque si no, es difícil de leer si no está así, intentado, digamos.

Bueno, pero esto es lo que quería mostrarles.

Bien, exactamente, esto es lo que quería mostrarles y la verdad que es un ejemplo un poco más técnico de lo que veníamos viendo, pero es algo que es necesario saber.

Está bueno que lo sepan.

Está bueno que hayan visto, de qué forma se puede testear un servicio web, aunque sea un ejemplo sencillo, igual bastante completo, porque pasamos por varios ejemplos, pero está bueno que tengan esa idea, ya lo vieron, ya tienen eso guardados, les queda este video y pueden repasarlo, y si tienen alguna entrevista laboral pueden decir "sé cómo se utiliza POSTMAN, no tengo experiencia, pero sé cómo hacerlo", lo cual ya los diferencia, un poco, de alguna forma.