

POSTMAN

Cuando probamos servicios la información puede venir de diferentes maneras:

1- Mediante un requerimiento:

Yo como Usuario del tipo Cliente registrado en la API de Comidas Ya

Deseo ingresar al sistema

Para poder hacer uso de las funcionalidades del sistema

****Metodo:** ** POST

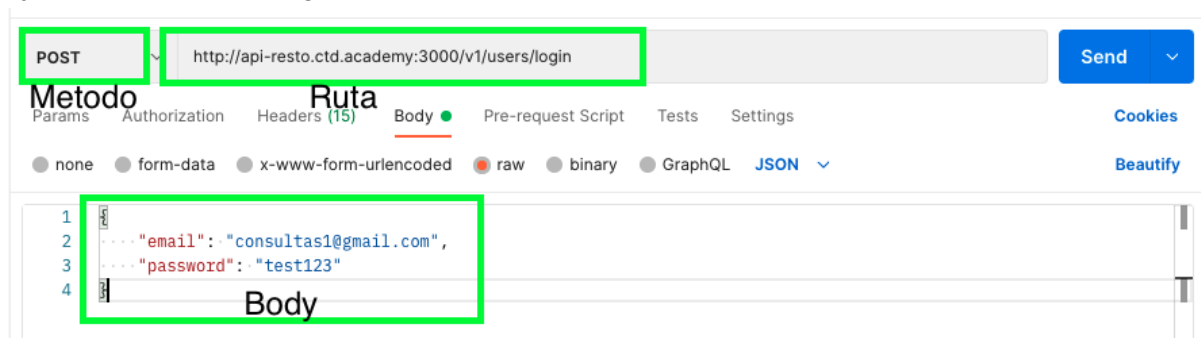
****Request:** ** http://api-resto.ctd.academy:3000/v1/users/login

Body:

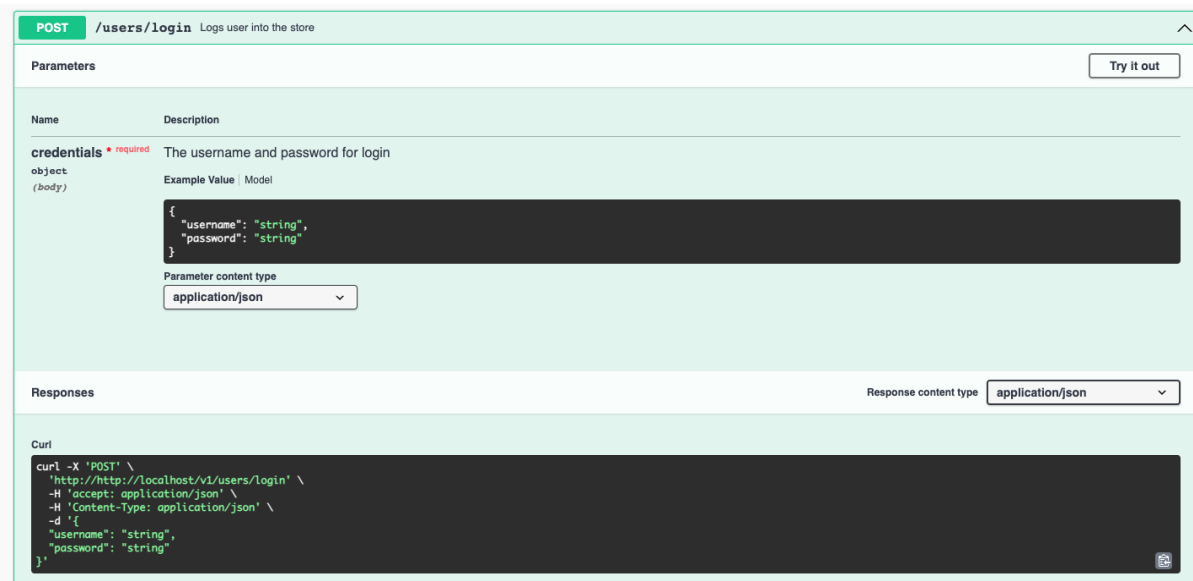
```
{  
  "email": string,  
  "password": string  
}
```

Aquí nos dan toda la información necesaria para llevarla a postman y ejecutar nuestras pruebas. Si se fijan no cambia a como lo veríamos en el FrontEnd, tiene los mismos campos y podemos jugar con los datos de pruebas de la misma forma que lo hicimos en las primeras clases. (Probar con email valido, password invalidas, etc) y aqui tmb tenemos un resultado, pero a diferencia de del frontend, aqui podemos ver los codigos de respuestas, 200,400, etc

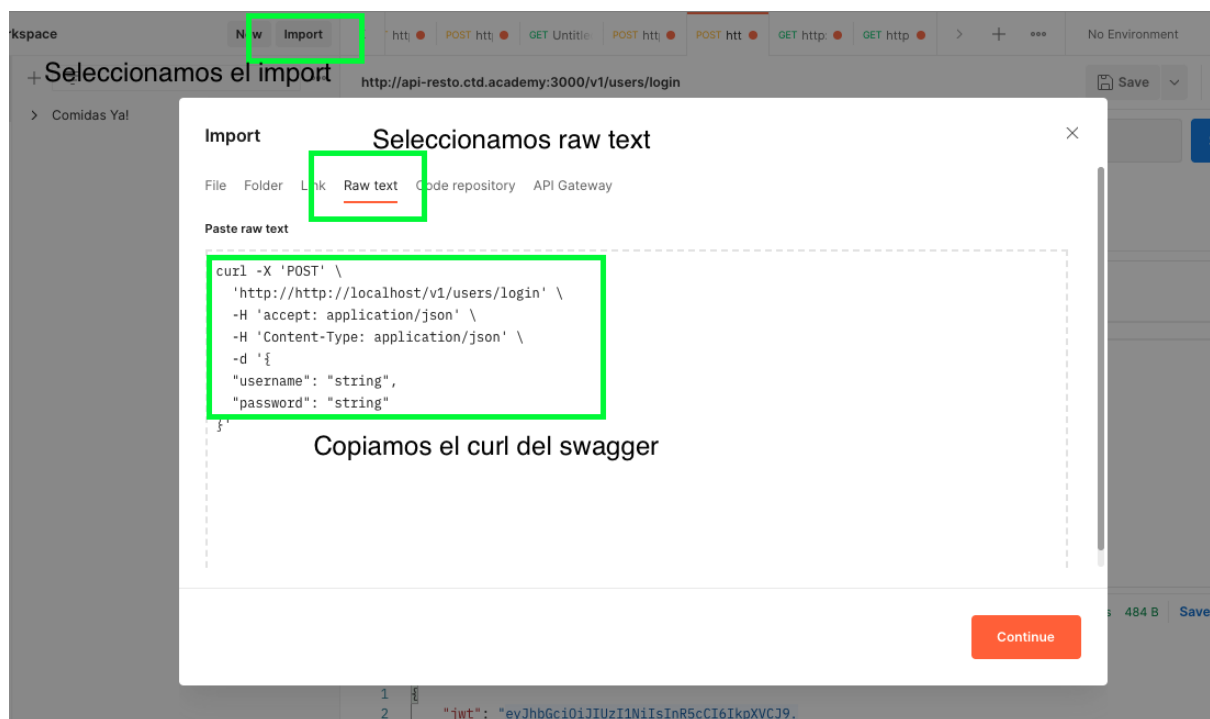
Ejemplo de cómo configuramos en postman:



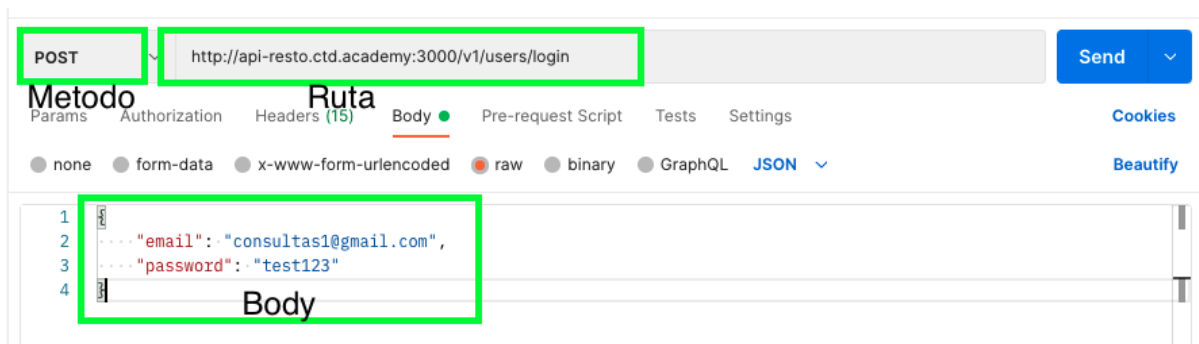
2-Mediante Swagger:



Aquí si se fijan nos da el curl con toda la información y solo debemos importarla en postman de manera más fácil, pero como ?



Una vez que le damos al continúe e importamos el curl que nos brinda el swagger, entonces vemos como toda la información se carga automáticamente en la herramienta a diferencia de la opción uno en la que se completó en base al requerimiento



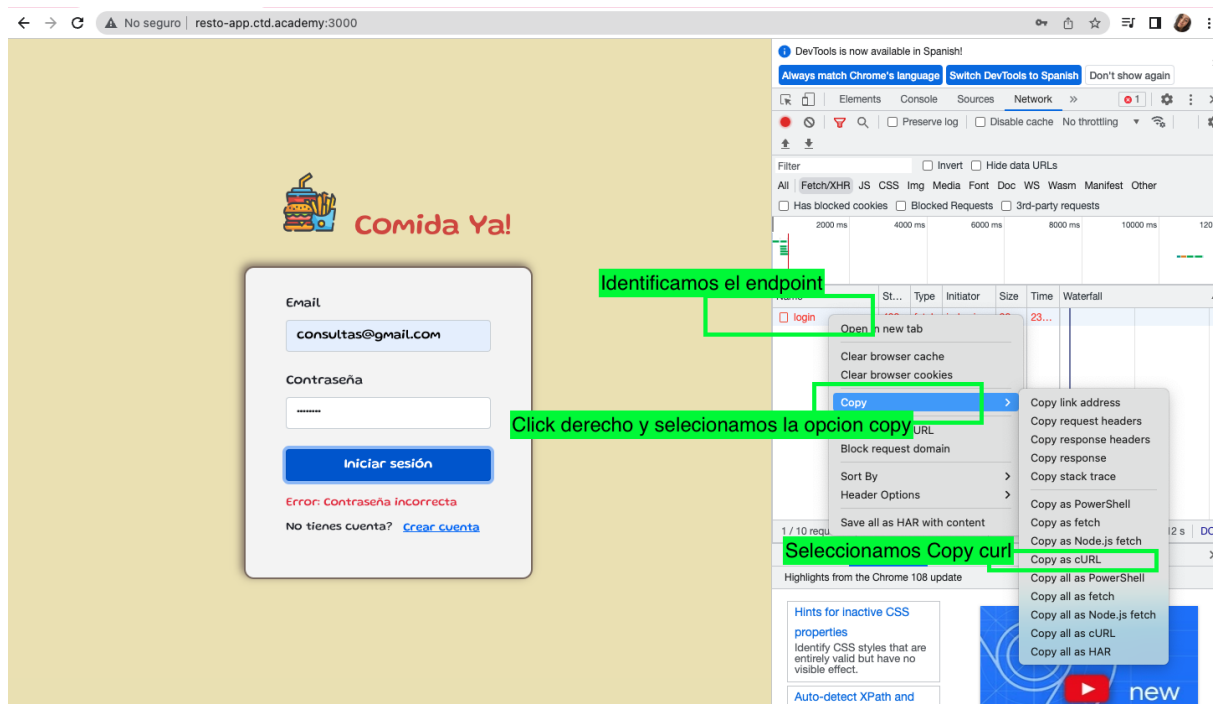
¿Que pasa cuando no tenemos swagger o documentación, pero si tenemos el frontend funcionando? Podemos sacar la información para las pruebas?

Si, esto es mediante la herramienta de desarrollo o devstool o inspector!

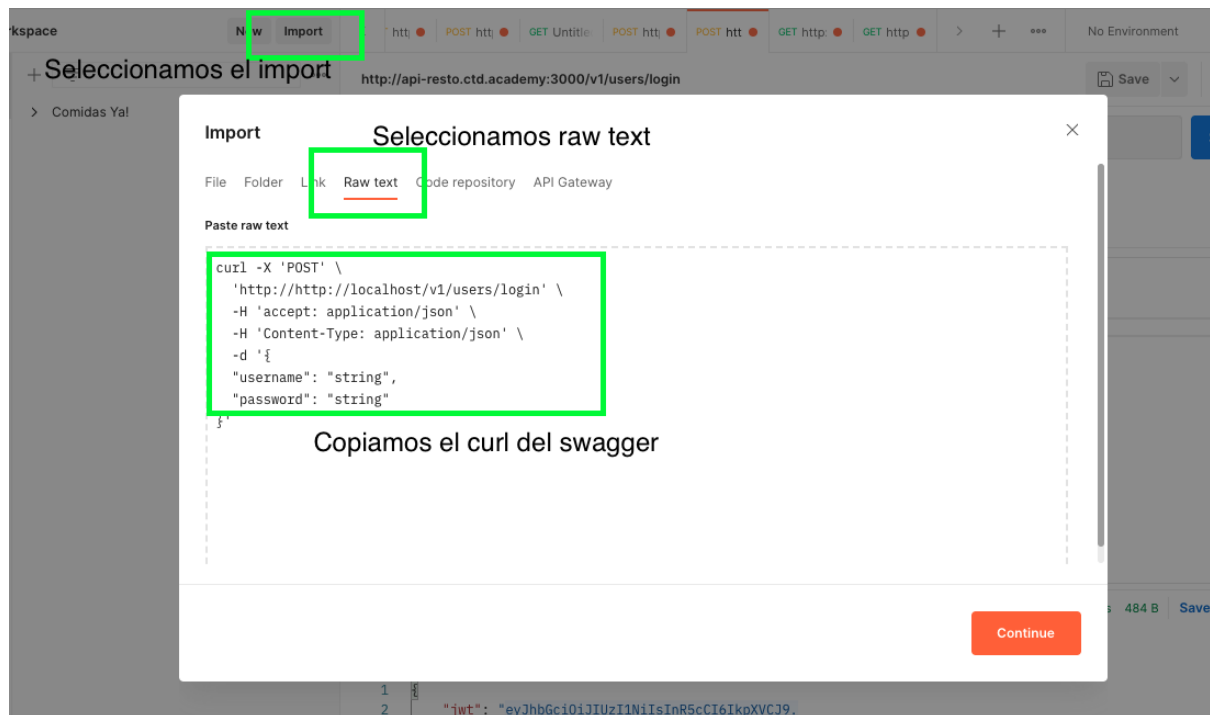
Existe dos maneras:

Opción 1 (La más Fácil)

Identificando el endpoint y copiando el cur (El curl viene con toda la informacion, metodo, ruta y body en caso de ser un POST)

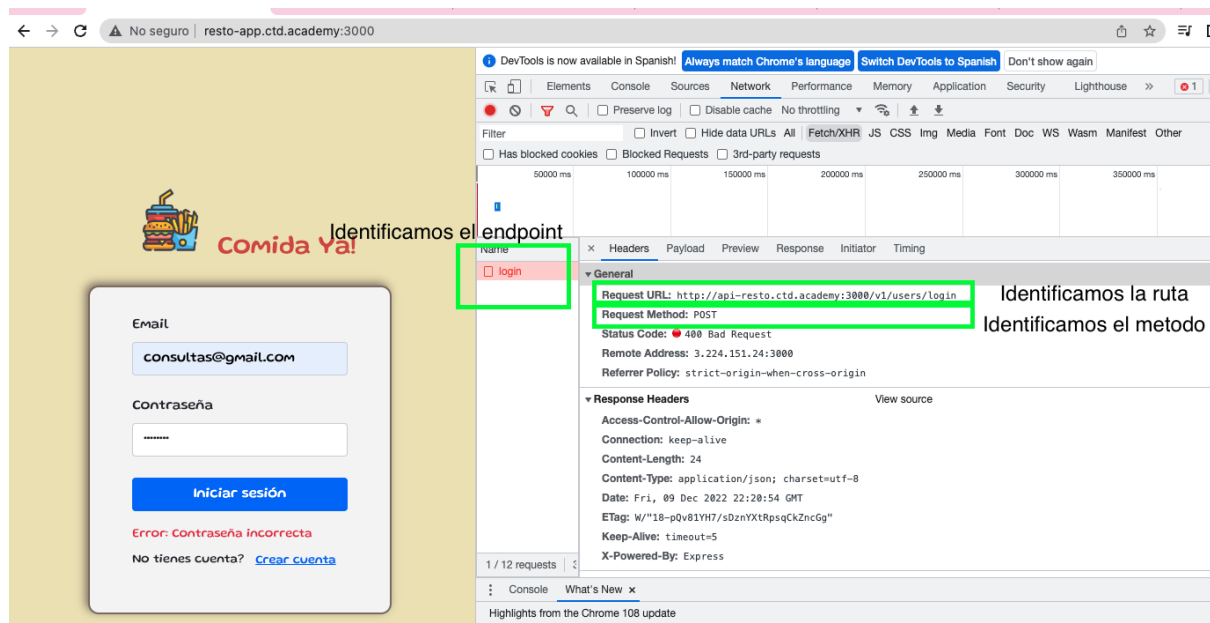


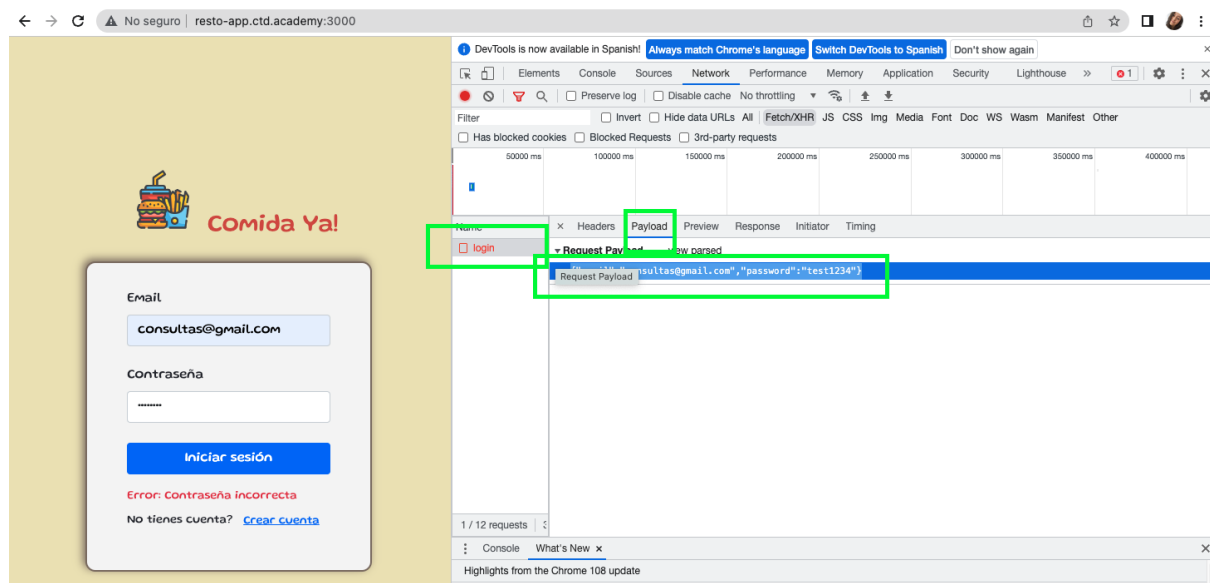
Una vez obtenido el curl, solo lo importamos en el postman



Opcin 2:

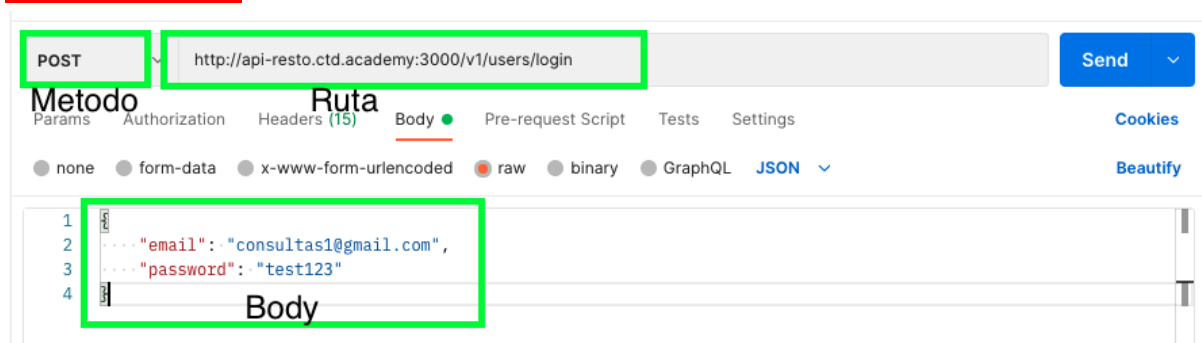
Identificamos el endpoint, la ruta, el mtodo y el body en caso de ser un POST





Toda esta información la llevamos a postman y comenzamos con nuestras pruebas.

Si se fijan, todas las opciones nos llevan a lo mismo. Lo importante es que identifiquen qué información tienen, ya sea documentación o simplemente la aplicación y vean que manera le resulta más útil!



Ya identificamos cómo arrancar nuestras pruebas o cómo llevar la información a postman para empezar a probar. Aquí lo importante es que vean las pruebas de servicios de la misma manera que ven las pruebas que venían haciendo, a diferencia de que no tienen una interfaz bonita y accesible para el usuario.

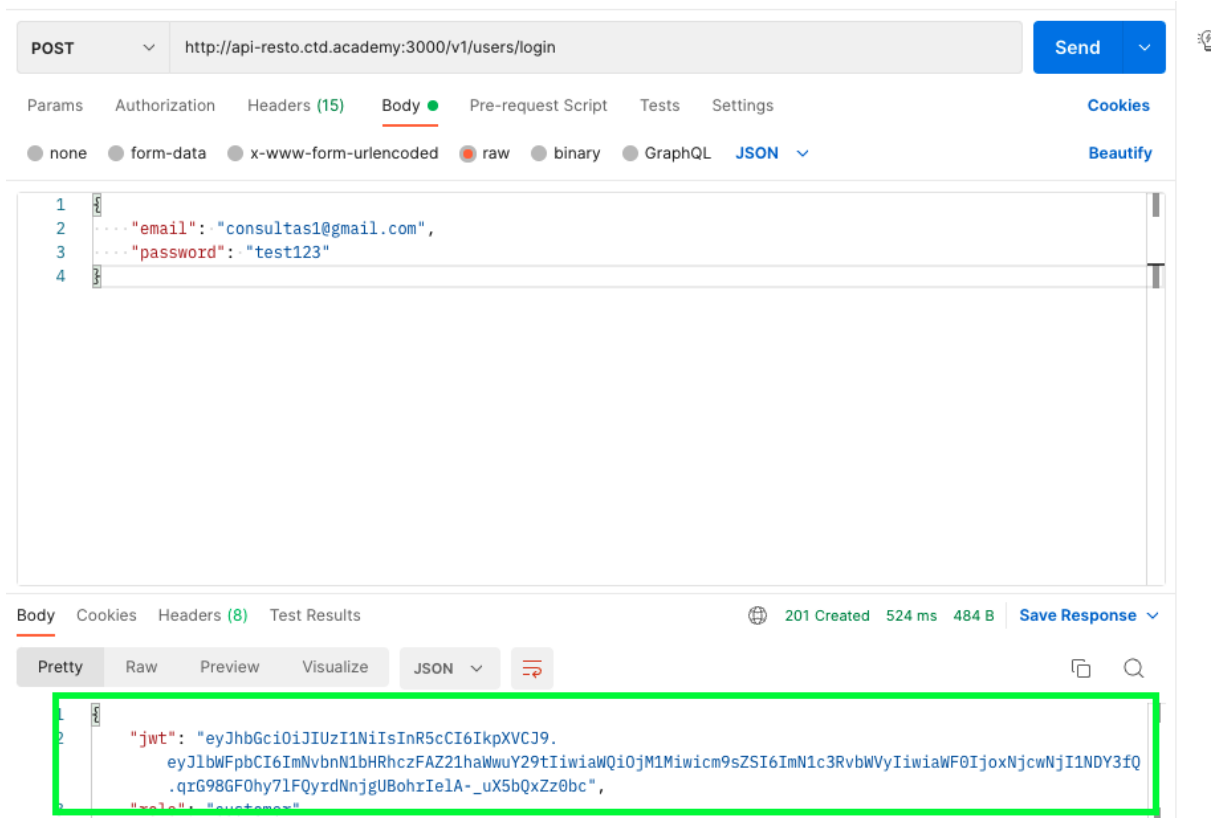
Entonces con esta información ya podríamos registrar o loguear un usuario.

Ahora, cuando usamos el token?

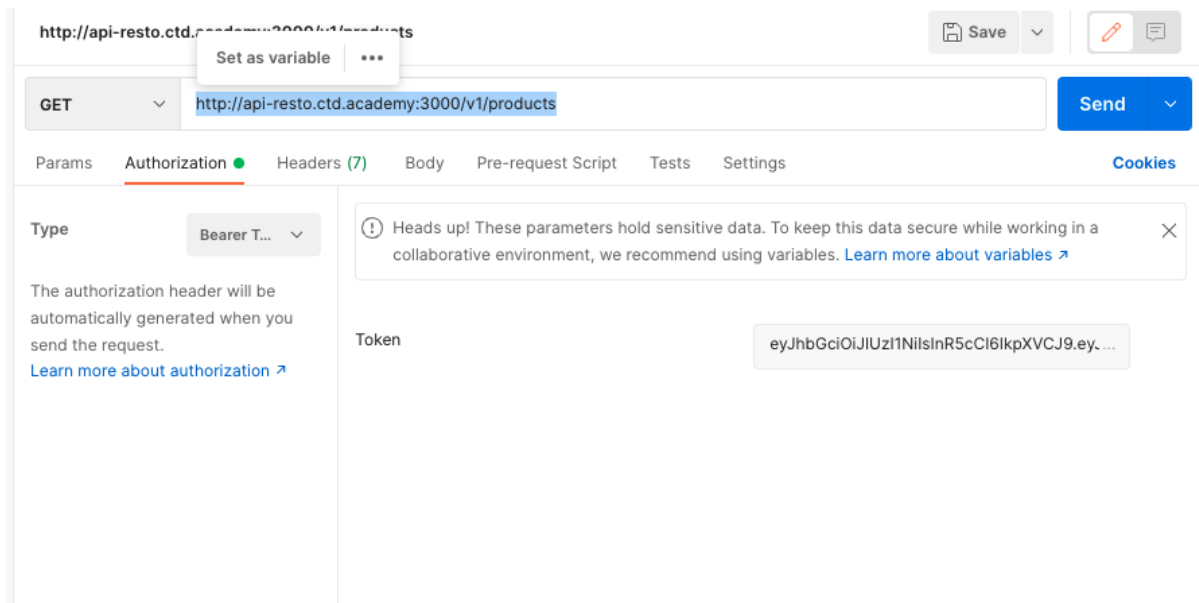
Yo les comentaba que no dejen de ver la aplicación desde front, porque si ingresan a comidas ya por ejemplo, entonces ustedes no ven la home porque deberían estar logueados en el sistema. Desde el lado de servicios pasa exactamente lo mismo, y es aquí cuando nos apoyamos en este caso en un token para tener esa sesión en mi otro endpoint.

Como hacemos eso?

Si me piden acceder a la home page de Comidas Ya, entonces debería haber iniciado sesión o no?



Aquí vemos como el inicio de sesión nos devuelve un token, que nos va a autorizar en otro endpoint para poder acceder a cierta página:



Si se fijan, aqui lo que hice fue poner mi ruta para ver la home page de productos, pero se sume el token de inicio de sesion para poder acceder a esa petición

Aqui les dejo otra aplicación para que puedan practicar:

<https://buggy.justtestit.org/register>

JEST

Pruebas unitarias/componente:

```
const dias = ['Lunes', 'Martes', 'Miercoles', 'Jueves', 'Viernes'];
```

Test unitario

```
it('Verificar la existencia del dia Domingo ', () => {  
  expect(arrProvincias()).toContain('Domingo');  
});
```

Si se fijan aqui les estan compartiendo el archivo con el test en si, pueden llevarlo a la herramienta y correrlo o simplemente pueden leerlo en detalle e identificar cual sería el resultado de ejecución y que lógica se esta usando.

Si observamos en detalle el test fallará porque está esperando buscar "Domingo" en el array "Provincias"