# INTRODUCCION

Hoy en día la seguridad de los sistemas informáticos es vital dada la creciente tendencia de ataques informáticos. Por ello es imprescindible dotar la API de BAK-END de autenticación para hacer uso de los métodos publicados en los cuales no se pueden acceder de forma anónima.

Desde Kabel nos tomamos la seguridad como algo fundamental, y para ello hemos decidido implementar la seguridad de acceso basándonos en el modelo de Json Web Tokken, recomendado como uno de los puntos del TOP 10 de la OWASP.

# QUÉ ES JWT

**JSON Web Token** (abreviado JWT) es un estándar abierto basado en JSON propuesto por IETF (RFC 7519) para la creación de tokens de acceso que permiten la propagación de identidad y privilegios (claims en inglés).

Por ejemplo, un servidor podría generar un token indicando que el usuario tiene privilegios de administrador y proporcionarlo a un cliente. El cliente entonces podría utilizar el token para probar que está actuando como un administrador en el cliente o en otro sistema.

El token está firmado por la clave del servidor, así que el cliente y el servidor son ambos capaz de verificar que el token es legítimo.

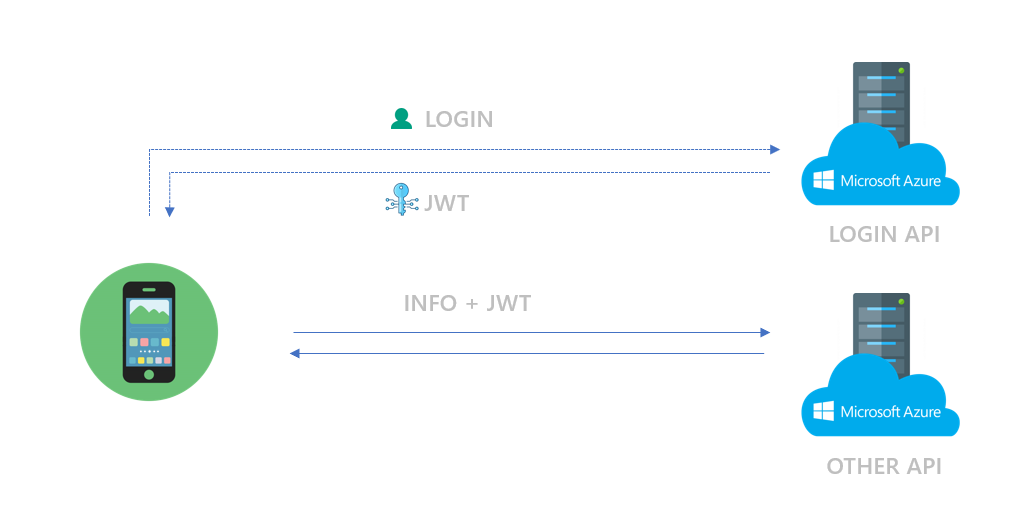
Los JSON Web Tokens están diseñados para ser compactos, poder ser enviados en las URLs al acceder al servicio deseado, y el servidor puede denegar u otorgar acceso.

Los JWTs pueden caducar, y se recomienda por tanto el uso de tiempos de validez (configurable desde la plataforma).

# CÓMO FUNCIONA

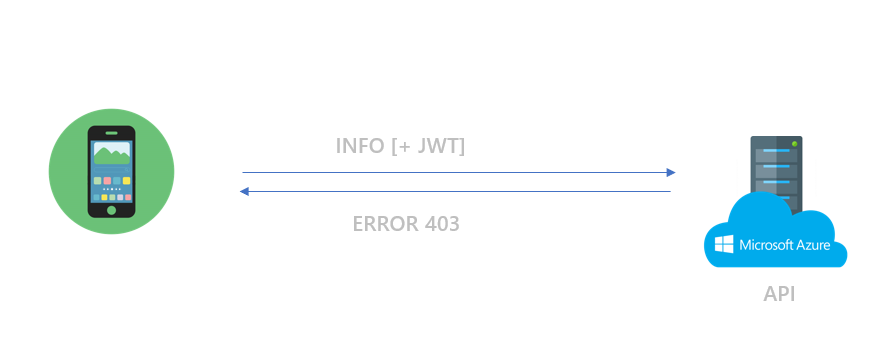
# Funcionamiento regular

1. El usuario hace LOGIN contra el servidor
2. El servidor produce un JWT en base a una contraseña interna del servidor
3. El usuario envía por cabecera dicho token en cada llamada
4. El servidor lo acepta como token válido
   1. El servidor devuelve la información requerida.



# Funcionamiento en caso de error

1. El usuario envía por cabecera dicho token en la llamada
2. El token es inválido o inexistente
3. El servidor devuelve un error 403 como que el end-point no es accesible.
4. El usuario (parte cliente) debe volver a generar el tokken



# IMPLEMENTACIÓN

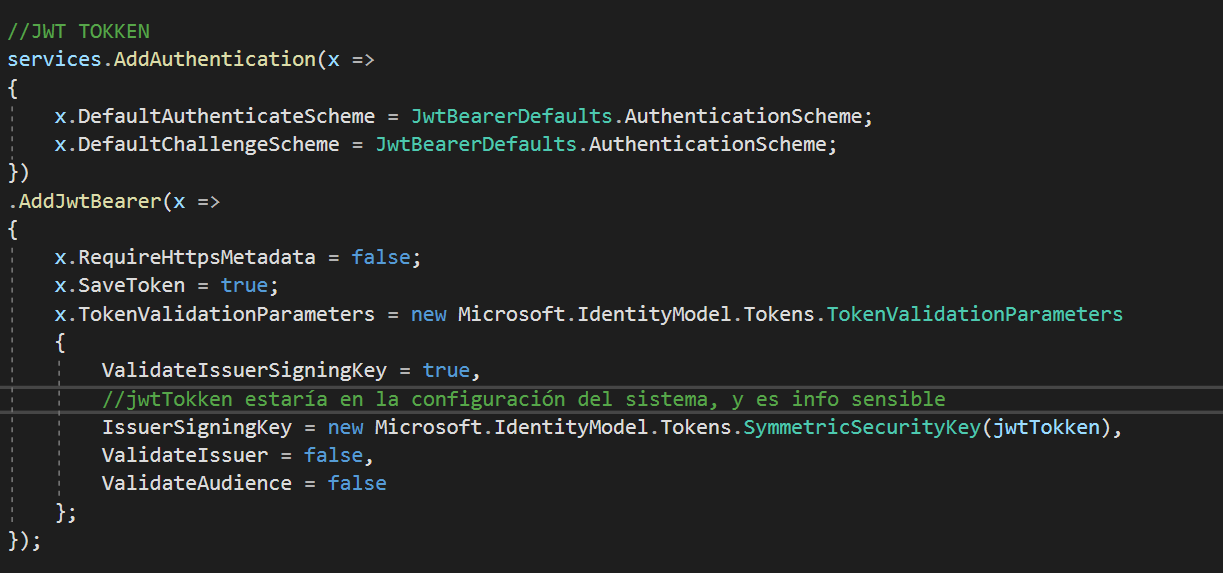
# DESARROLLO en .NET (back-end)

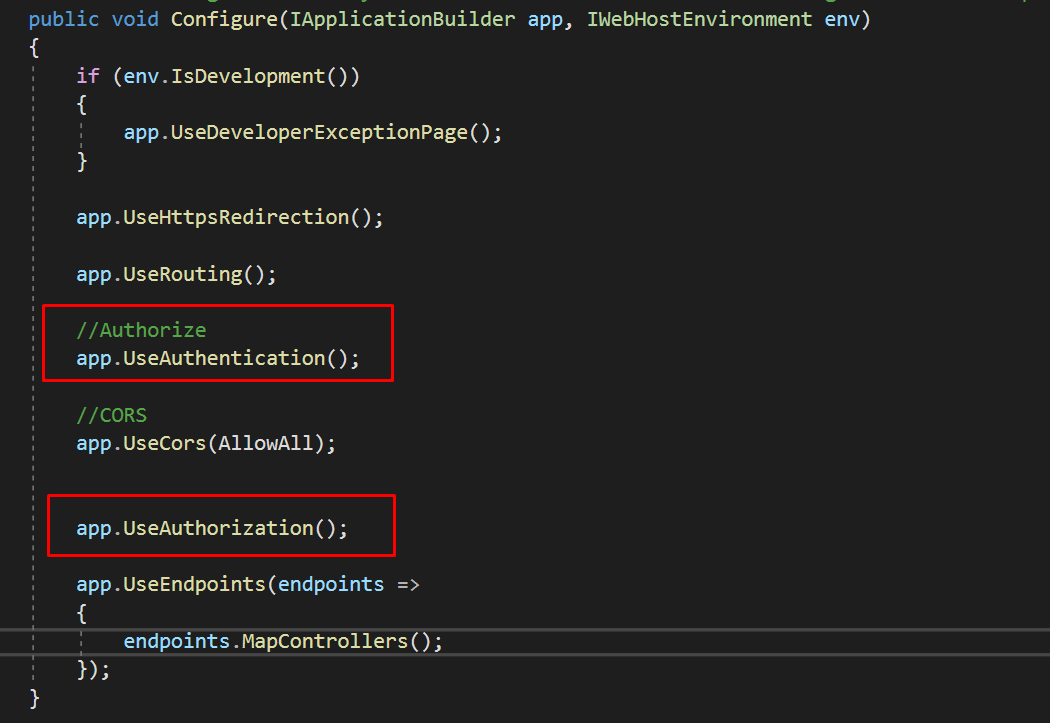
Para el uso de JWT existen librerías disponibles que se acoplan perfectamente a la tecnología de MCV .NET CORE como

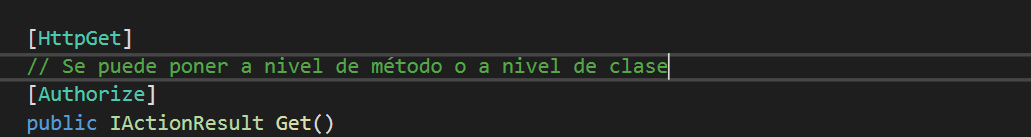
|  |
| --- |
| System.IdentityModel.Tokens.Jwt |

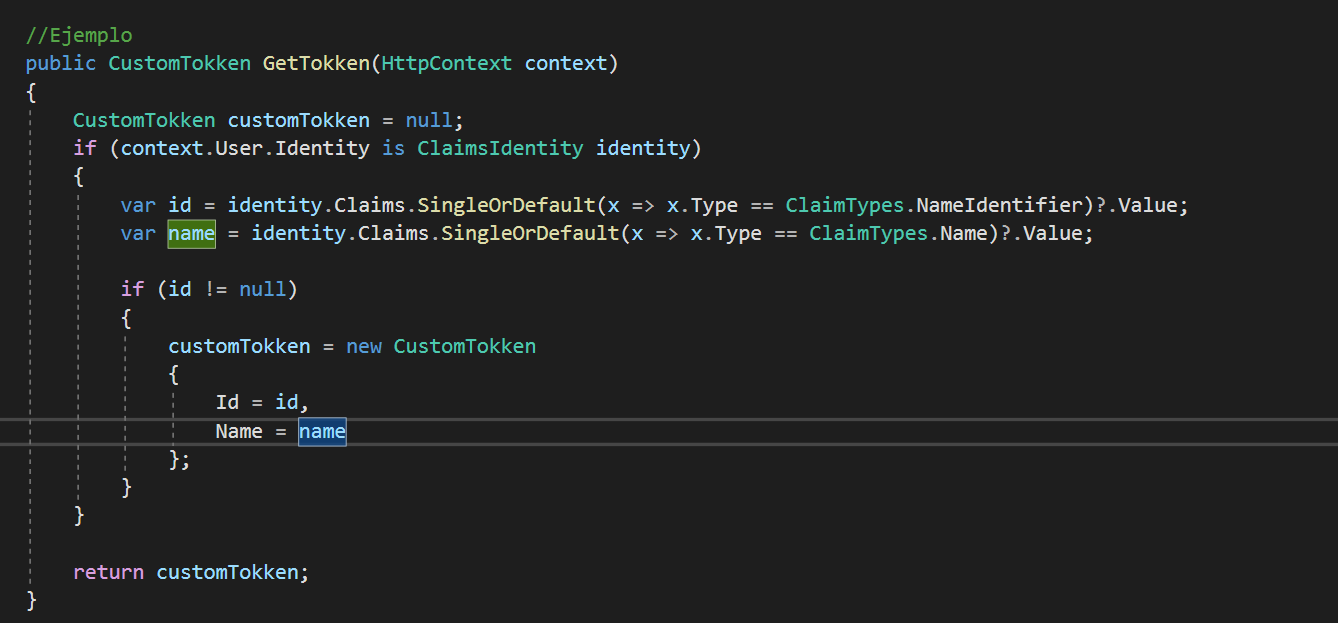
# Pasos

1. Configuración del entorno (se da de alta en los servicios y se añade a la configuración del aplicativo)





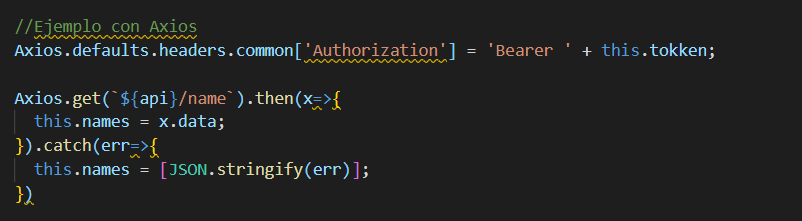
1. Se expresa de forma explícita el uso del TOKKEN para el método en cuestión
2. Obtención de la información guardada en el token generado (ver apartado OTRAS COMPROBACIONES)



# CREACIÓN DEL TOKKEN (back-end)

# DESARROLLO en Angular (front-end)

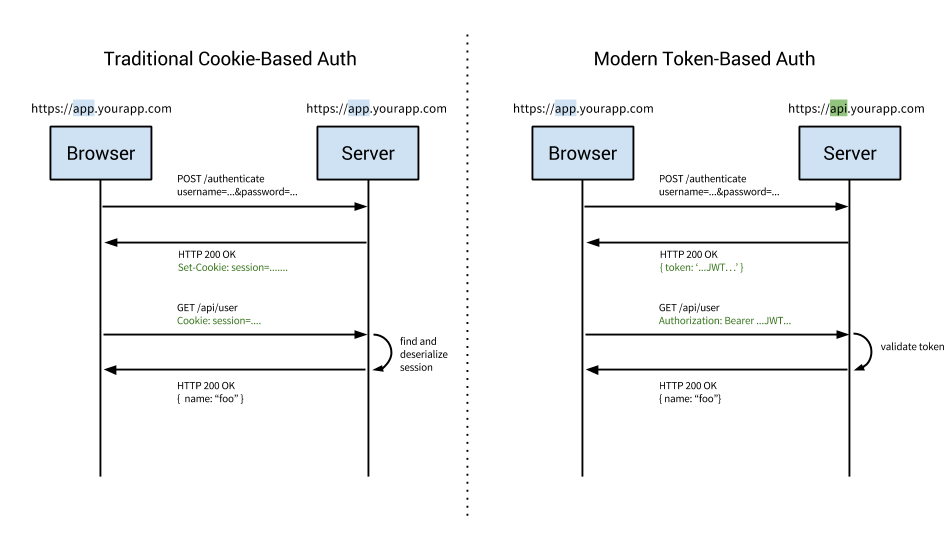
Se trata tan solo de añadir en la cabecera de la llamada el token



# OTRAS COMPROBACIONES

En el momento de la obtención del token, una vez validado, se pueden hacer otras comprobaciones que serán motivo de análisis según las necesidades del cliente.

# ¿Por qué JWT es mejor que usar COOKIES?



# COOKIES

La autenticación basada en cookies ha sido el método predeterminado para gestionar la autenticación de usuarios hasta hace poco tiempo.

La autenticación basada en cookies presenta un es **stateful**, lo que quiere decir, que al iniciar sesión, el usuario envía sus credenciales (y estas se validan), el servidor registra los datos (con el fin de recordar que el usuario se ha identificado correctamente). Estos datos que se registran en el backend, en correspondencia con el identificador de sesión, es lo que se conoce como estado.

En el lado del cliente una cookie es creada para almacenar el identificador de sesión, mientras que los datos se almacenan en el servidor (y son llamados variables de sesión).

Esto conlleva grandes problemas en cuanto a la suplantación, ya que las cookies finalmente se quedan guardadas en el equipo del usuario, además de sobrecargar el uso del servidor ya que se debe tener constancia de cada usuario conectado, y gestionar las sesiones constantemente.

# TOKKEN

La autenticación basada en tokens ha ganado prevalencia en los últimos años debido al aumento de las Single Page Applications (SPA), web APIs y IOT.

Cuando hablamos de autenticación con tokens, generalmente hablamos de autenticación con JSON Web Tokens (JWT), ya que, los JWT se han convertido en un estándar debido a su gran acogida en los diferentes entornos de desarrollo.

Su gran ventaja es la autenticación basada en tokens carece de estado (es stateless), lo que quiere decir que, el servidor ya no guarda información de qué usuarios están conectados o qué tokens se han emitido. Esto es así porque cada solicitud realizada al servidor va acompañada de un token, y el servidor es capaz de verificar la autenticidad del mismo.