



# Protegiendo tu API REST con JWT en aplicaciones ASP.NET MVC



*Autor*

**SANTI MACIAS**

Lider Técnico Comunidad Microsoft en atSistemas y friki de pelis y series de ciencia ficción en mi tiempo libre.



1. Conceptos básicos de seguridad
2. Cookies vs Tokens
3. Definiciones de JWT
4. Fundamentos de JWT
5. Anatomía de JWT
6. Estructura de un Token
7. Nuestros amigos debuggers
8. Ciclo de vida de un Token
9. Librerías y vulnerabilidades
10. Vamos a la acción



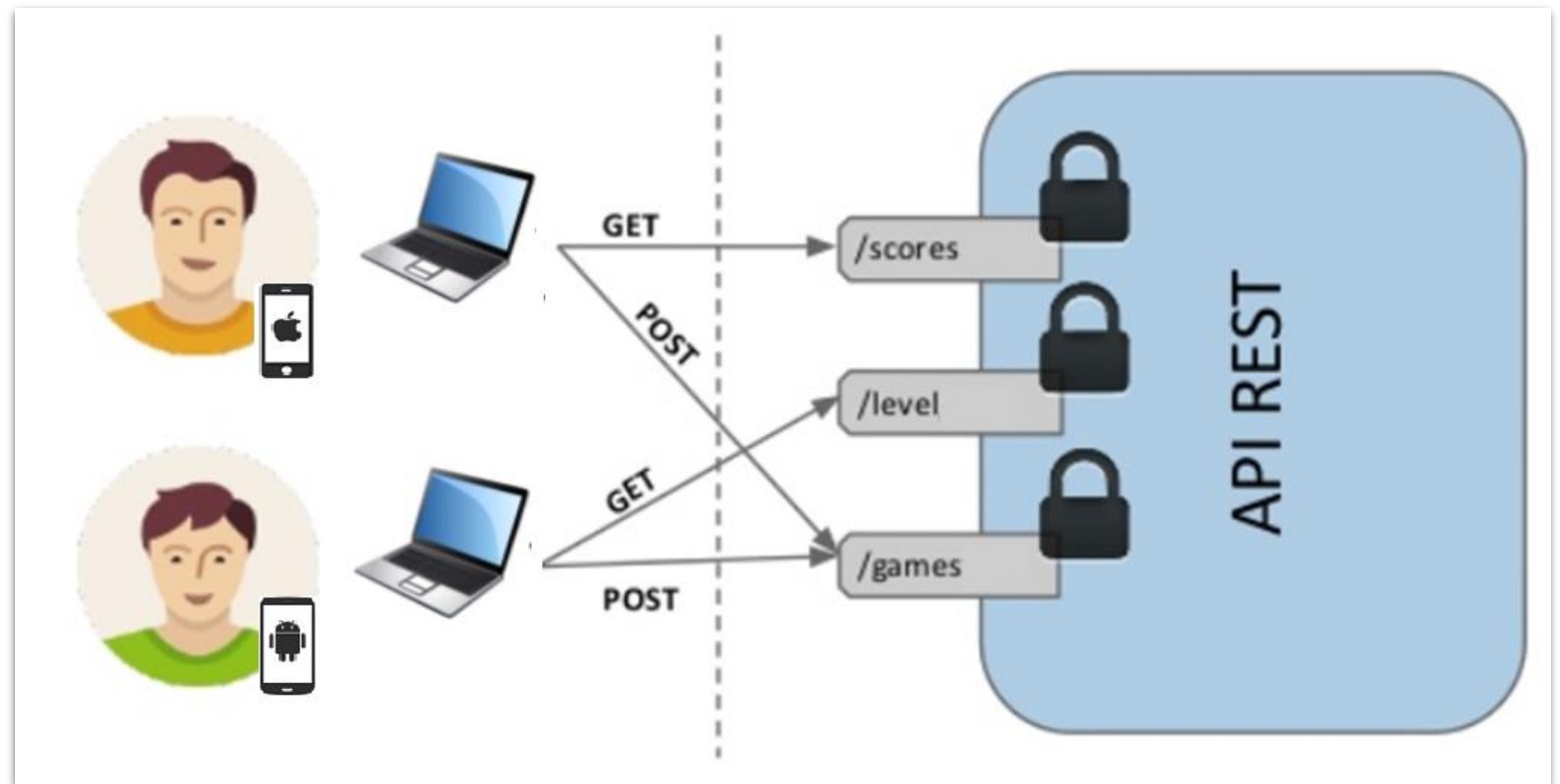
## UN REPASO A CONCEPTOS BASICOS

### Autenticación

- Recepción hotel
- Login con password
- HTTP 401 Unauthorized

### Autorización

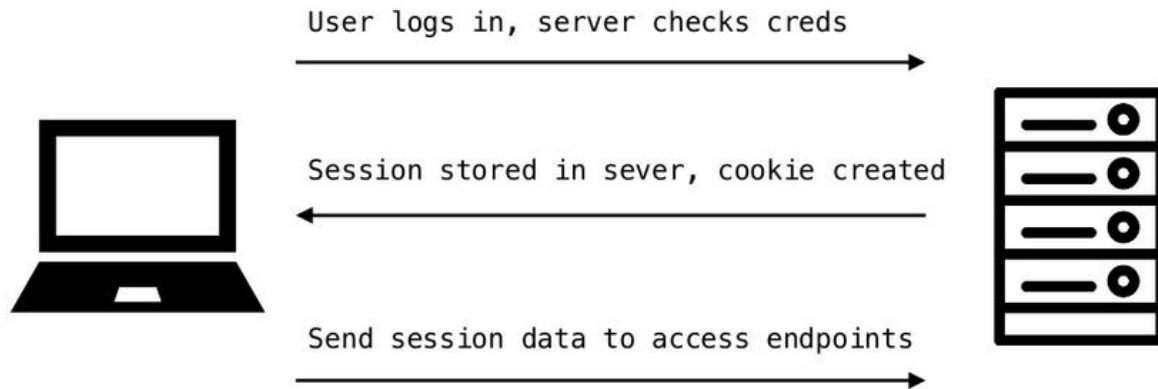
- Llave de la habitación
- Permisos de acceso
- HTTP 403 Forbidden





## Cookies vs Tokens

### Traditional Authentication Systems

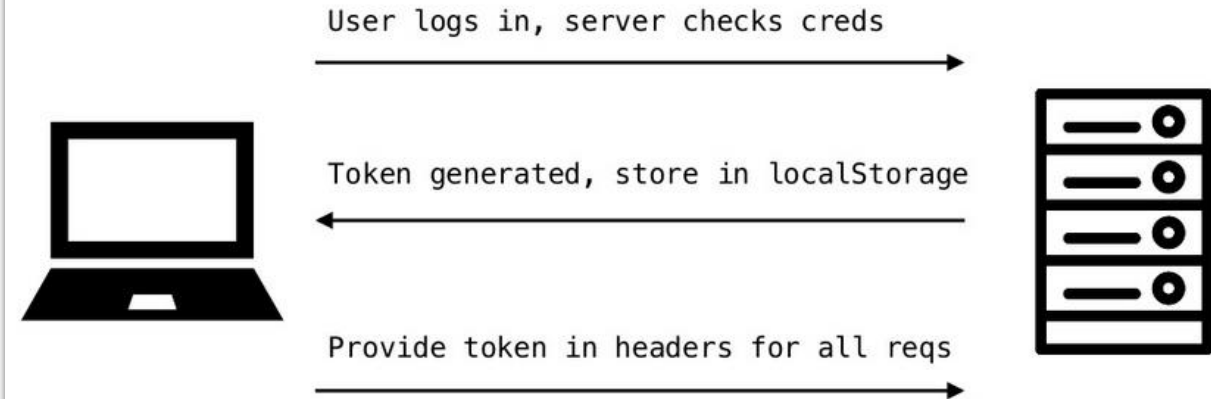


**Cookies:** WebForms, asp.net mvc, etc

**Sesión:** Necesitan ser guardas en el servidor, -Escalable

**Datos:** Contiene la sessionID, AUTH del usuario

### Token-Based Authentication Systems



**Tokens:** Api key, OAuth2, openID, SSO, etc.

**Sesión:** Se almacena en cada cliente, +Escalable

**Datos:** Contiene información del usuario







- Desktop, Mobile & Web Ready!!
- Ligero: podemos codificar gran cantidad de datos y pasarlo como una cadena.
- Self-container: Delegamos mantener el estado al cliente.
- Stateless: Creamos servicios optimizados desacoplados del servidor .
- Scalable: Los webserver pueden escalar sin problemas y aumentar en rendimiento.
- La información es confiable porque está firmada digitalmente.
- *"Authorization: Bearer token"* es la forma más común de enviarlo (existen otras).
- ¡Nos olvidamos de cookies!



- JWT sirve para transmitir información de un usuario garantizando integridad de datos entre un cliente/servidor mediante una cadena de texto codificada en Base64.

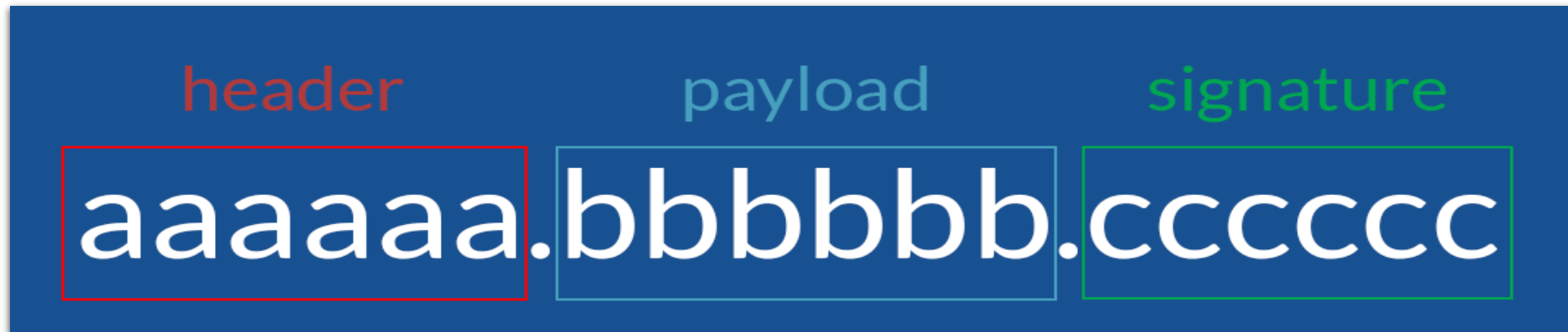


- Los JWT son mecanismos para transferir datos, no para asegurarlo
- Los JWT son seguros cuando se utiliza conjuntamente con HTTPS

**RECORDAR:** Siempre, debemos usar HTTPS entre el cliente/servidor para encriptar las peticiones.



## ANATOMIA DEL TOKEN



**HEADER:** Indica el algoritmo y tipo de Token.

**PAYLOAD:** Datos de usuario/claims

**SIGNATURE:** la firma, para verificar que el token es válido.



# ESTRUCTURA DEL TOKEN (Claims)

## Registered claims

|     |   |                                   |
|-----|---|-----------------------------------|
| jti | → | Id del token: String              |
| iss | → | Issuer (emisor): StringOrUri      |
| aud | → | Audiencia: StringOrUri            |
| sub | → | Subject (tema): StringOrUri       |
| iat | → | Cuándo se creó: NumericDate       |
| exp | → | Cuándo expira: NumericDate        |
| nbf | → | Tiempo hasta validez: NumericDate |

**Claims:** No son obligatorios.

**Claims:** Se recomienda seguir este formato.

**Claims:** No todas las librerías .NET los implementan.

**jti:** Es muy útil para usar Tokens de un solo uso y evitar ataques.

**Especificación:** <https://tools.ietf.org/html/rfc7519>





# JWT – ESTRUCTURA DEL TOKEN

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1bm90aW4iOiJ1bm90aW4iLCJ1c2VmdWwiOiJ1bm90aW4iLnOlg86mQIADPd24\_FnflpkWpE74SSFxsMtcfSmlEjeA

## Header

```
{  
  "alg": "HS256",  
  "typ": "JWT"  
}
```

## Payload

```
{  
  "name": "JWT4B",  
  "useful": true  
}
```

## Signature

```
HMACSHA256  
(  
  base64UrlEncode(Header)  
  + "."  
  + base64UrlEncode(Payload)  
)
```

**HEADER:** Algoritmo Hash HS256 y token JWT.

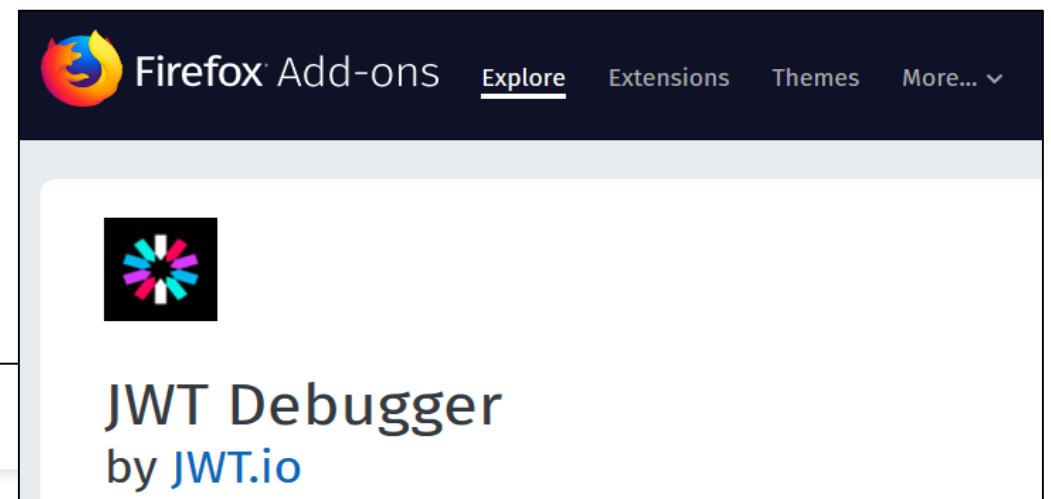
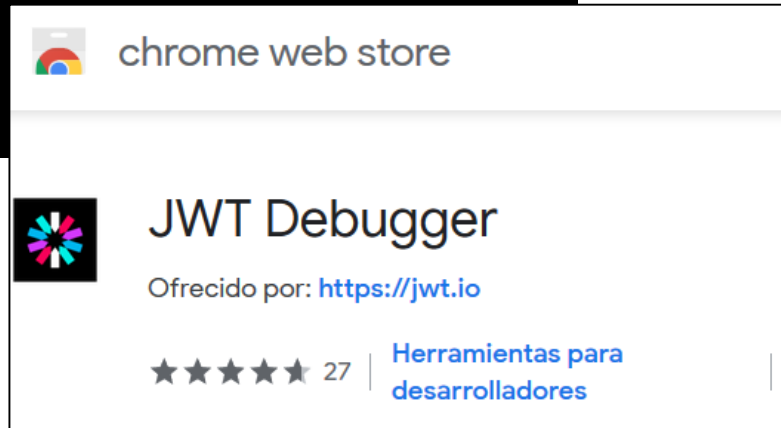
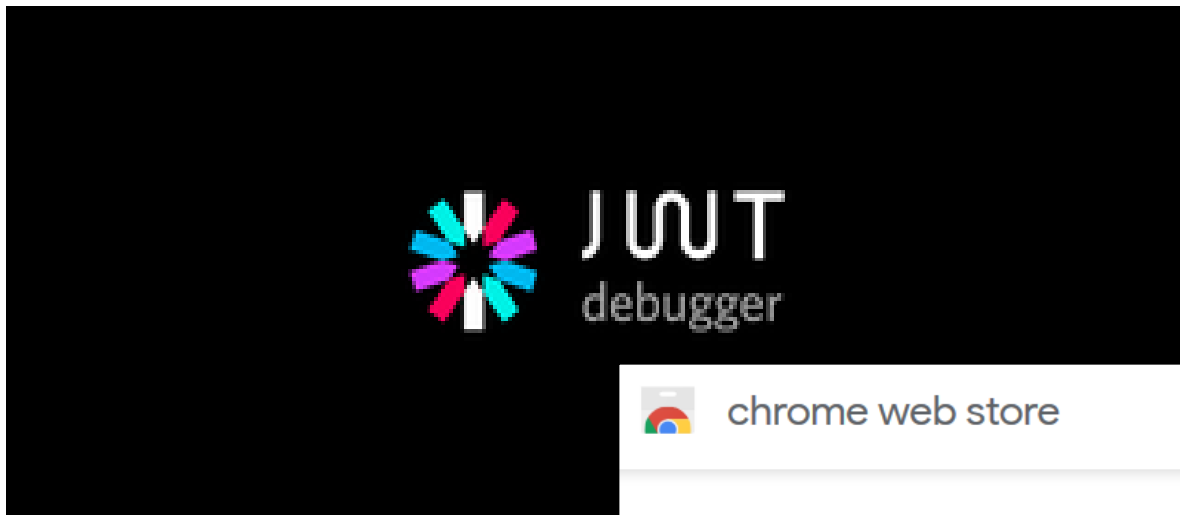
**PAYLOAD:** Datos de nombre usuario y lo que necesite nuestra API para validar la petición, recordar que nosotros generamos el token y podemos incluir todos los atributos que queramos.

**SIGNATURE:** Firma para la integridad del Token

Aquí lo importante es el "SECRET" con el que firmamos y que ahora explicaremos.



# NUESTRO AMIGOS DEBUGGERS





**JWT** Debugger Libraries Introduction Ask Get a T-shirt! Crafted by Auth0

### Debugger

ALGORITHM HS256

**Encoded** PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWVhbnR5dWV9LjJVA950rM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ
```

**Decoded** EDIT THE PAYLOAD AND SECRET (ONLY HS256 SUPPORTED)

HEADER: ALGORITHM & TOKEN TYPE

```
{  "alg": "HS256",  "typ": "JWT"}
```

PAYLOAD: DATA

```
{  "sub": "1234567890",  "name": "John Doe",  "admin": true}
```

VERIFY SIGNATURE

```
HMACSHA256(  base64UrlEncode(header) + "." +  base64UrlEncode(payload),  secret)
```

☐ secret base64 encoded

☒ Signature Verified

**JWT** Debugger Libraries Introduction Ask Get a T-shirt! Crafted by Auth0

### Debugger

ALGORITHM HS256

**Encoded** PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWVhbnR5dWV9LjJVA950rM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ
```

**Decoded** EDIT THE PAYLOAD AND SECRET (ONLY HS256 SUPPORTED)

HEADER: ALGORITHM & TOKEN TYPE

```
{  "alg": "HS256",  "typ": "JWT"}
```

PAYLOAD: DATA

```
{  "sub": "1234567890",  "name": "John Doe",  "admin": true}
```

VERIFY SIGNATURE

```
HMACSHA256(  base64UrlEncode(header) + "." +  base64UrlEncode(payload),  otro-secret)
```

☐ secret base64 encoded

☒ Invalid Signature

**DEMO:** Lo importante es el **SECRET** con el que firmamos el token y no debemos darlo a nadie.



# JWT LIBRERIAS Y VULNERABILIDADES

<https://docs.microsoft.com/en-us/security-updates>

## Security Advisories and Bulletins

📅 10/11/2017 • ⌚ 2 minutes to read • Contributors 🐾

In this library you will find the following security documents that have been released by the Microsoft Security Response Center (MSRC). The MSRC investigates all reports of security vulnerabilities affecting Microsoft products and services, and releases these documents as part of the ongoing effort to help you manage security risks and help keep your systems protected.

- [Security Bulletins](#)
- [Security Bulletin Summaries](#)
- [Security Advisories](#)
- [Microsoft Vulnerability Research Advisories](#)
- [Acknowledgments](#)
- [Glossary](#)





# JWT LIBRERIAS Y VULNERABILIDADES

## Libraries for Token Signing/Verification

FILTER BY All

Warning: Critical vulnerabilities in JSON Web Token libraries with asymmetric keys. [Learn more](#)

| .NET        |         | .NET        |         | .NET (RT)   |         |
|-------------|---------|-------------|---------|-------------|---------|
| ✓ Sign      | ✓ HS256 | ✓ Sign      | ✓ HS256 | ✓ Sign      | ✓ HS256 |
| ✓ Verify    | ✓ HS384 | ✓ Verify    | ✓ HS384 | ✓ Verify    | ✓ HS384 |
| ✓ iss check | ✓ HS512 | ✗ iss check | ✓ HS512 | ✗ iss check | ✓ HS512 |
| ✓ sub check | ✓ RS256 | ✗ sub check | ✓ RS256 | ✗ sub check | ✓ RS256 |
| ✓ aud check | ✓ RS384 | ✗ aud check | ✓ RS384 | ✗ aud check | ✓ RS384 |
| ✓ exp check | ✓ RS512 | ✗ exp check | ✓ RS512 | ✗ exp check | ✓ RS512 |
| ✓ nbf check | ✓ ES256 | ✗ nbf check | ✓ ES256 | ✗ nbf check | ✓ ES256 |
| ✓ iat check | ✓ ES384 | ✗ iat check | ✓ ES384 | ✗ iat check | ✓ ES384 |
| ✓ jti check | ✓ ES512 | ✗ jti check | ✓ ES512 | ✗ jti check | ✓ ES512 |

Microsoft 302 View Repo

Install-Package System.IdentityModel.Tokens.Jwt

DV 378 View Repo

Install-Package jose-jwt

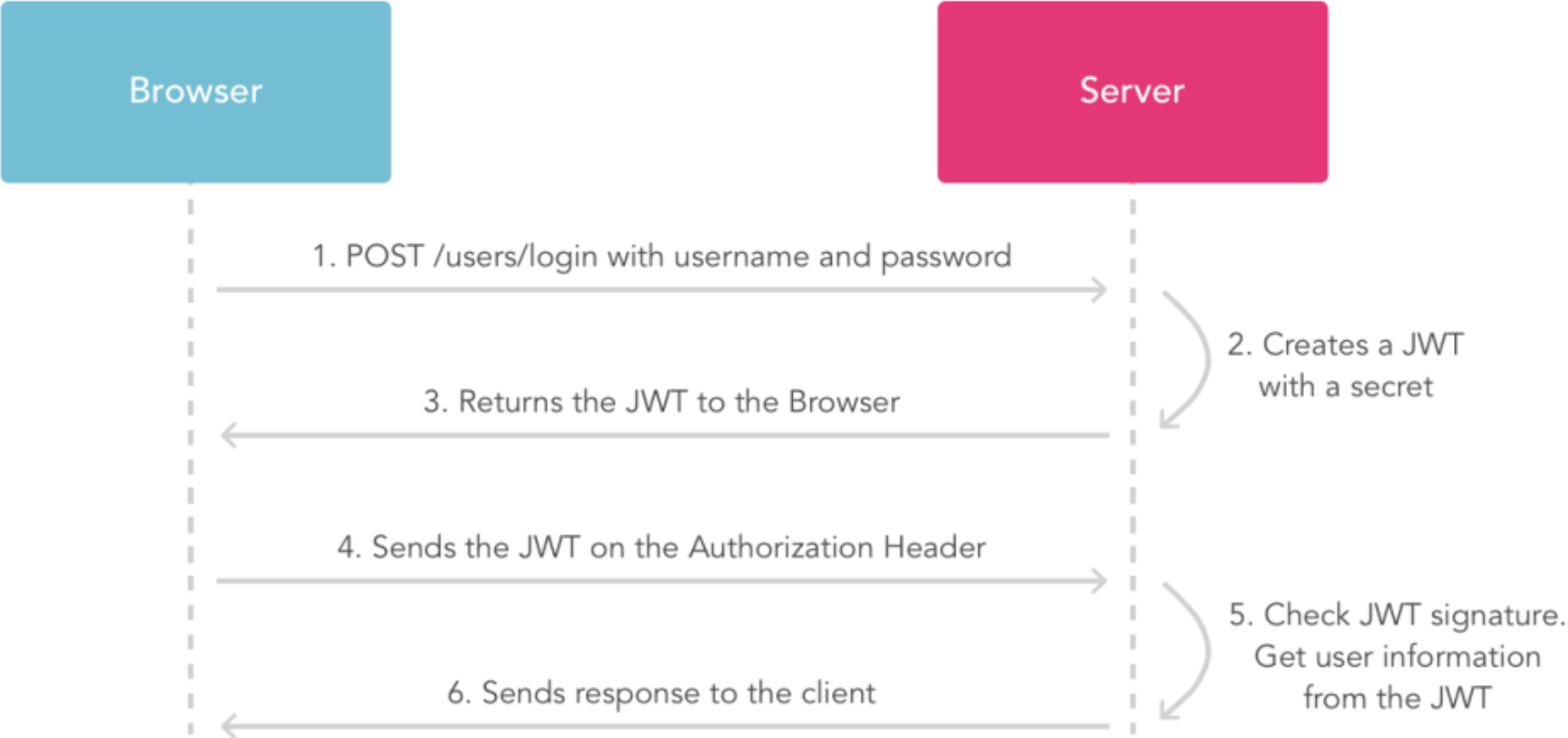
DV View Repo

Install-Package jose-rt

**Vulnerabilities:** <https://docs.microsoft.com/en-us/security-updates/securityadvisories/2017/3214296>

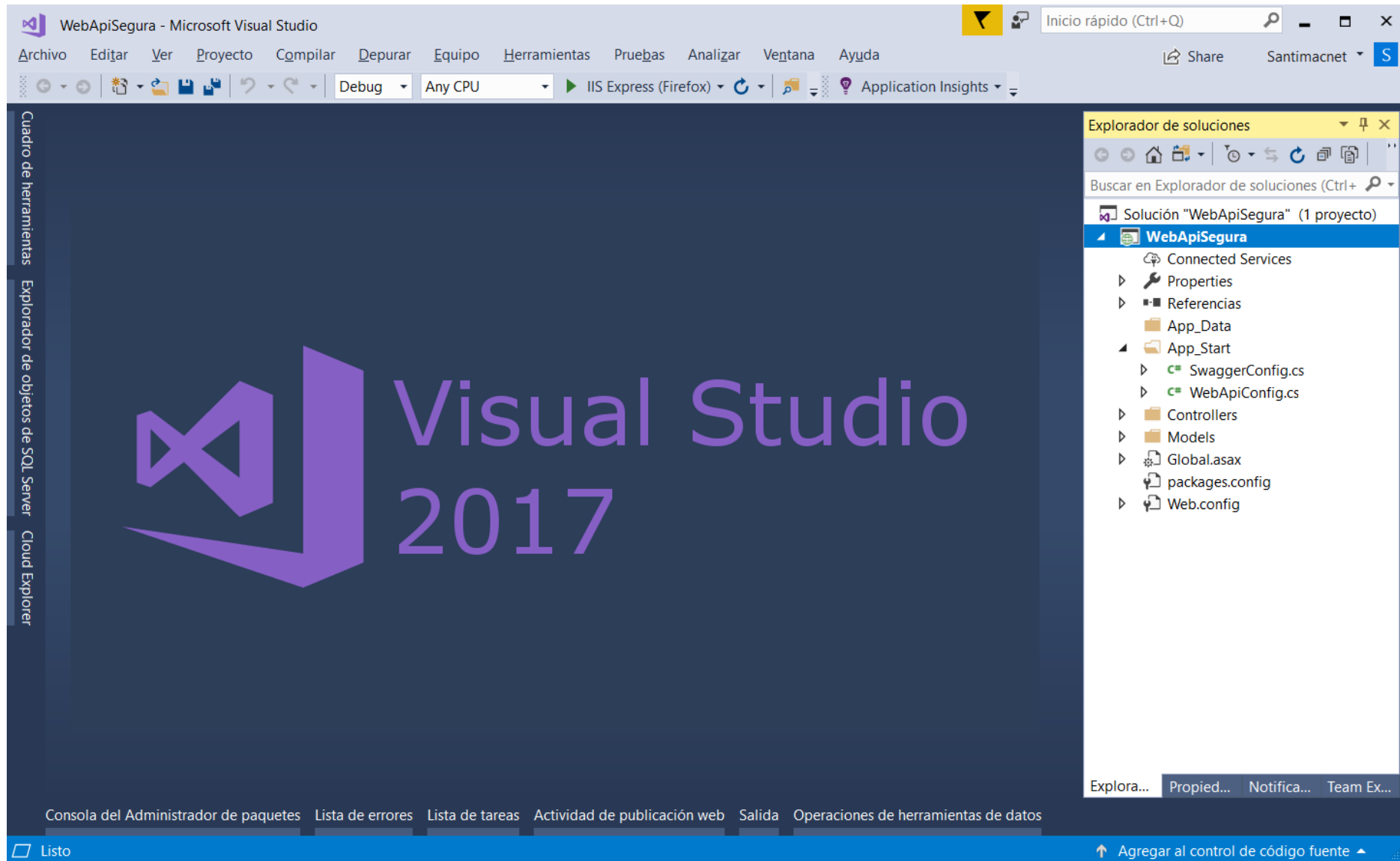


# Ciclo de vida de un Token



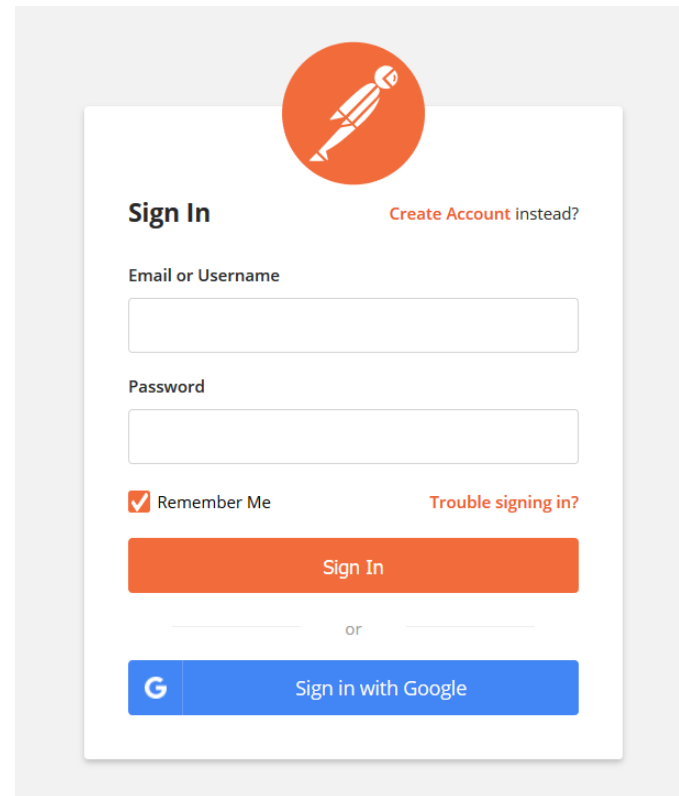
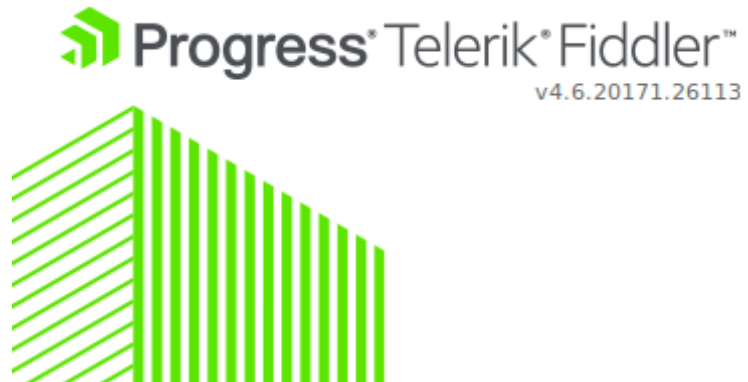


# DEMO ASP.NET WEB API





# DEBUGGERS

A screenshot of the Fiddler web application's sign-in page. At the top center is an orange circular icon containing a white silhouette of a person with a bow and arrow. Below this is the 'Sign In' heading, with a link 'Create Account instead?' to its right. The form contains two input fields: 'Email or Username' and 'Password'. Below the password field is a checkbox labeled 'Remember Me' and a link 'Trouble signing in?'. A large orange 'Sign In' button is positioned below the form fields. Below the button is a horizontal line with the word 'or' in the center. At the bottom is a blue button with the Google 'G' logo and the text 'Sign in with Google'.

# CUESTIONES CLAVE

---

- Que pasa cuando recibo el token en mi controlador
- Que pasa cuando caduca el token en mi aplicación
- Que pasa cuando realizamos logout en app/web
- Que pasa con mis token en devlocal y producción
- Que pasa si tengo varias API REST que usan JWT
- Quien y donde se gestionan los usuarios de mi API

# REFERENCIAS



- <https://enmilocalfunciona.io/construyendo-una-web-api-rest-segura-con-json-web-token-en-net-parte-i/>
- <https://enmilocalfunciona.io/construyendo-una-web-api-rest-segura-con-json-web-token-en-net-parte-ii/>
- <https://enmilocalfunciona.io/construyendo-una-web-api-rest-segura-con-json-web-token-en-net-parte-iii/>
- <https://github.com/santimacnet/WebAPI-Segura-JWT>
  
- <https://jwt.io>
- <https://www.jsonwebtoken.io>
- <https://tools.ietf.org/html/rfc7519>
- <https://docs.microsoft.com/en-us/security-updates/securityadvisories/2017/3214296>
- <https://auth0.com/blog/ten-things-you-should-know-about-tokens-and-cookies/>

# GRACIAS



[www.atsistemas.com](http://www.atsistemas.com)

902 888 902