Se tiene que es una de las nuevas tendencias en cuanto al desarrollo web moderno se refiere, es la autenticación por medio de *Tokens* y que [nuestro backend sea un API RESTful sin información de estado](http://carlosazaustre.es/blog/como-implementar-una-api-rest-con-mongodb-node-js-usando-express-v4/), *stateless*.

El funcionamiento es el siguiente. El usuario se autentica en nuestra aplicación, bien con un par usuario/contraseña, o a través de un proveedor como puede ser Twitter, Facebook o Google por ejemplo. A partir de entonces, cada petición HTTP que haga el usuario va acompañada de un *Token* en la cabecera. Este Token no es más que una firma cifrada que permite a nuestro API identificar al usuario. Pero este Token no se almacena en el servidor, si no en el lado del cliente (por ejemplo en *localStorage* o *sessionStorage*) y el API es el que se encarga de descifrar ese Token y redirigir el flujo de la aplicación en un sentido u otro.

Como los tokens son almacenados en el lado del cliente, no hay información de estado y la aplicación se vuelve totalmente escalable. Podemos usar el mismo API para diferentes aplicaciones (Web, Mobile, Android, iOS, ...) solo debemos preocuparnos de enviar los datos en formato JSON y generar y descifrar tokens en la autenticación y posteriores peticiones HTTP a través de un middleware.

También nos añade más seguridad. Al no utilizar cookies para almacenar la información del usuario, podemos evitar ataques CSRF (*Cross-Site Request Forgery*) que manipulen la sesión que se envía al backend. Por supuesto podemos hacer que el token expire después de un tiempo lo que le añade una capa extra de seguridad.

El estándar para este tipo de autenticación es utilizar [JSON Web Tokens (JWT)](http://self-issued.info/docs/draft-ietf-oauth-json-web-token.html). Al igual que los APIs, el formato JSON es agnóstico del lenguaje, y podemos utilizar el que queramos (Node.js, Python, Ruby, PHP, .NET, Java,...)