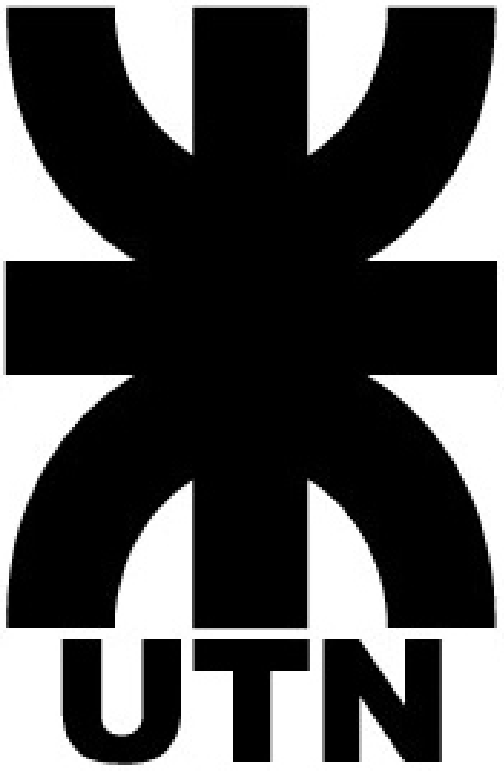
Actividad Semana 7

Token de usuario - introspección - refresh

UTN FRC - Integración de Aplicaciones en Entorno Web - 5K4



Cátedra: 5K4
Profesores:

- María Soledad Romero
- Rubén Aníbal Romero

Estudiante:

- 85828 Gutierrez, Santiago

Fecha: 29/09/2024

Índice

Token de aplicación

El token de aplicación (o client token) se utiliza cuando una aplicación, sin interacción directa del usuario, necesita autenticarse ante un servidor o API para realizar tareas automatizadas o de servicio. Identifica a la aplicación.

cURL

```
curl --location '<token_url>' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'grant_type=client_credentials' \
--data-urlencode 'client_id=<client_id>' \
--data-urlencode 'client_secret=<client_secret>
```

Token de usuario

El user token (token de usuario) se utiliza en los escenarios donde se necesita autenticar y autorizar a un usuario individual para acceder a recursos protegidos en nombre de ese usuario. Este tipo de token es emitido cuando el usuario se autentica y se asocia con sus permisos y roles específicos.

cURL

```
curl --location '<token_url>' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'client=' \
--data-urlencode 'grant_type=password' \
--data-urlencode 'client_id=<client_id> \
--data-urlencode 'client_secret=<client_secret> \
--data-urlencode 'username=<username>' \
--data-urlencode 'password=<password>'
```

Endpoint introspect

El endpoint introspect se usa para verificar la validez y contenido de un token. Este endpoint permite a un sistema de terceros o a la propia aplicación comprobar si un token (ya sea de acceso o refresh) es válido, si ha expirado, o si ha sido revocado. Se utiliza generalmente cuando no se quiere o no se puede validar el token directamente en la aplicación.

cURL

```
curl --location '<token_introspect_url>' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'token=<access_token>' \
```

```
--data-urlencode 'client_id=<client_id>' \
--data-urlencode 'client_secret=<client_secret>'
```

Endpoint refresh

El endpoint refresh se utiliza para renovar un token de acceso que ha expirado o está por expirar, utilizando un refresh token. El objetivo es evitar que el usuario tenga que volver a autenticarse cada vez que el token de acceso expira, siempre y cuando el refresh token siga siendo válido.

cURL

```
curl --location '<token_url>' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'grant_type=refresh_token' \
--data-urlencode 'client_id=<client_id> \
--data-urlencode 'client_secret=<client_secret> \
--data-urlencode 'refresh_token=<refresh_token>'
```

Agregar un atributo al usuario y mostrarlo en el jwt

Para agregar un atributo al usuario y mostrarlo en el JWT, es necesario configurar el atributo y agregarlo al payload. En el siguiente artículo se encuentra explicado el procedimiento en Keycloack: “Adding user attributes to JWT token in Keycloak”.

cURL

```
curl --location '<token_introspect_url>' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'token=<access_token>' \
--data-urlencode 'client_id=<client_id>' \
--data-urlencode 'client_secret=<client_secret>'
```

Validar el token a través de código en lugar de introspect

Utilizando el modulo de node **jsonwebtoken**, verificamos la signatura del JWT utilizando el token, la clave pública que obtuvimos de Keycloack y el algoritmo de firma. Si el token es inválido, se informa por consola. Si es válido, se muestra por consola su payload.
Puede visualizar el código en el siguiente repositorio:
https://github.com/GutierrezSantiago/UTN-FRC-IAEW-2024/tree/main/JWTDecodingJS

Como decodear un token usando código

Separar el Token:

- Divide el token usando `split('.')` para obtener las tres partes: encabezado, carga útil y firma.

Reemplazar Caracteres:

- En la parte de carga útil, reemplaza `-` por `+` y `_` por `/`.

Agregar Padding:

- Ya que la longitud de la cadena Base64 tiene que ser un múltiplo de 4, se añade `=` si es necesario.

Decodificar:

- Utiliza `atob()` para decodificar la parte de carga útil y convertirla a texto.

Parsear a JSON:

- Convierte la cadena JSON a un objeto JavaScript usando `JSON.parse()` y retorna el objeto.

Puede visualizar el código en el siguiente repositorio:
https://github.com/GutierrezSantiago/UTN-FRC-IAEW-2024/tree/main/JWTDecodingJS

Bibliografía

Hardt, D. (Ed.). (2012, Octubre). *The OAuth 2.0 authorization framework* (Request for Comments No. 6749). Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/rfc6749

Richer, J. (Ed.). (2015, Octubre). *OAuth 2.0 token introspection* (Request for Comments No. 7662). Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/rfc7662