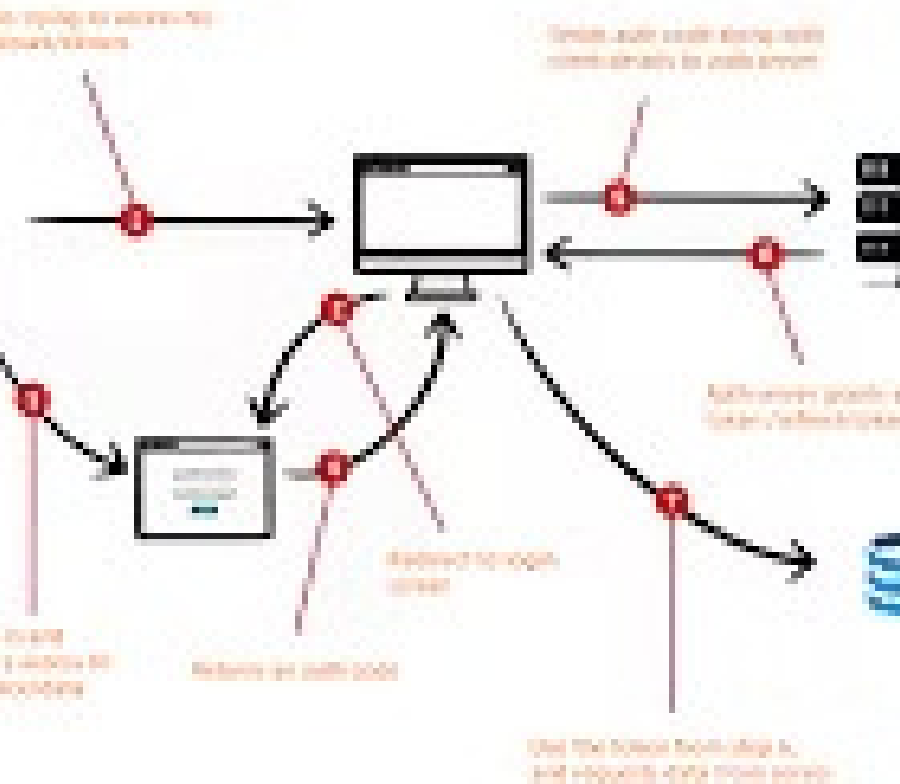


OAuth 2.0



OAuth 2.0: Autorización Segura para Aplicaciones Modernas

Integrantes

- Simbaña Adrian
- Socasi Moises
- Marlon Torres

OAuth 2.0

¿Qué es OAuth 2.0 y por qué es importante?

Definición


OAuth 2.0 es un protocolo de autorización. Permite a las aplicaciones acceder a recursos. Lo hace sin compartir credenciales.

Propósito

Su objetivo es proporcionar acceso limitado. Una aplicación puede actuar en nombre del usuario. No necesita la contraseña.

Importancia

Mejora la seguridad y la experiencia del usuario. Es fundamental en el desarrollo de aplicaciones modernas.



OAuth 2.0

Los Cuatro Roles Clave en la Arquitectura OAuth 2.0

Propietario del recurso

Es el usuario que autoriza el acceso a sus datos.

Cliente

Es la aplicación que solicita acceso a los recursos.

Servidor de autorización

Autentica al usuario y emite tokens de acceso.

Servidor de recursos

Aloja los datos protegidos y verifica el token.

Flujos de Autorización: Una Mirada Detallada a las Opciones

1

Código de Autorización

Utilizado para aplicaciones web del lado del servidor.

2

Credenciales del Cliente

Para acceso a la API de la aplicación, no al usuario.

3

Implícito

Utilizado para aplicaciones del lado del cliente (desaconsejado).

4

Dispositivo

Para dispositivos sin navegador (TV, IoT).

OAuth 2.0 y Spring Boot: Integración Paso a Paso

Agregar Dependencias
Incluir Spring Security OAuth2 en tu proyecto.

1

Configurar Servidor

Definir clientes, alcances y tipos de concesión.

2

Implementar Endpoints
Crear endpoints para autorización y token.

3

Proteger Recursos

Asegurar endpoints con OAuth 2.0.

4

Ejemplo Práctico: Implementando OAuth 2.0 con Spring Security



Código

Mostrar la configuración del servidor de autorización.



Seguridad

Cómo proteger los recursos con anotaciones.



API

Ejemplo de acceso a un recurso protegido.



Consideraciones de Seguridad y Mejores Prácticas

Validar Tokens

Asegurarse de que los tokens sean auténticos.

HTTPS Obligatorio

Usar HTTPS para toda comunicación.

Almacenar Tokens Seguros

No guardar tokens en el cliente.

Rotación de Tokens

Cambiar los tokens periódicamente.

Conclusiones:


OAuth 2.0 es esencial para la seguridad. Es crucial en las aplicaciones modernas.


Proporciona acceso controlado a los recursos. Mejora la experiencia del usuario.


Implementar buenas prácticas es fundamental. Garantiza la seguridad de las aplicaciones.


Implementacion


mbsocasi/
micro_usuarios_oauth




 1
Contributor

 0
Issues

 0
Stars

 0
Forks





GitHub - mbsocasi/micro_usuarios_oauth

Contribute to mbsocasi/micro_usuarios_oauth development by creating an account on GitHub.