Autenticação e Autorização

Luigi Ciambarella Filho

Conhecendo a turma

https://www.menti.com/aljs4wxbmab9

Acordos

Autenticação

O que é?

- Verificação da identidade de um usuário.
- Geralmente envolve nome de usuário e senha.
- Responde à pergunta: "Quem é você?"

Autenticação

Tipos de Autenticação

- Local: Usuários armazenados no próprio sistema (banco de dados, arquivos, etc.).
- Externa: Utiliza serviços de terceiros (Google, Facebook, Microsoft, etc.).
- Baseada em Tokens: Um token é gerado após a autenticação e usado para acessar recursos.

Autorização

O que é?

- Determinação do que um usuário autenticado pode acessar.
- Define permissões e restrições.
- Responde à pergunta: "O que você pode fazer?"

Autorização

Tipos de Autorização

- Baseada em Funções (Role-Based): Permissões definidas para grupos de usuários (ex: administrador, usuário comum).
- Baseada em Permissões (Permission-Based): Permissões específicas para cada ação (ex: ler, escrever, executar).
- Baseada em Atributos (Attribute-Based): Permissões concedidas com base em atributos do usuário ou recurso.

Por que Autenticação e Autorização são Importantes?

- **Segurança**: Protege dados sensíveis e recursos de acesso não autorizado.
- **Privacidade**: Garante que os usuários só acessem informações relevantes para suas funções.
- **Conformidade**: Atende a requisitos regulatórios (ex: LGPD).
- **Experiência do Usuário**: Permite personalizar o acesso e funcionalidades.

Modelo Zero-Trust

O modelo Zero Trust, ou "confiança zero", é um conceito de segurança que parte do princípio de que nenhum usuário ou dispositivo deve ser automaticamente confiável, mesmo que esteja dentro da rede da organização. Em vez disso, todos os acessos devem ser verificados e autorizados explicitamente, independentemente da localização ou do dispositivo.

https://www.menti.com/aljs4wxbmab9

ada

Implementação

Como implementar autenticação?

- Autenticação local
  - Autenticação baseada em cookies
    - Microsoft Identity Platform
  - Autenticação baseada em tokens
    - OpenID Connect
- Autenticação externa
  - Single Sign-On (SSO)
    - Provedores de identidade social
    - SAML
    - OpenID Connect
- Autenticação biométrica
- Autenticação baseada em certificado digital
- Autenticação multifator (MFA)(Inclui 2FA)

Implementação

Como implementar autorização?

- Controle de acesso baseado em função (RBAC)
- Controle de acesso baseado em atributos (ABAC)
- Controle de acesso baseado em permissões (PBAC)
- Listas de controle de acesso (ACL)
- Autorização baseada em tokens
  - OAuth 2.0

OAuth 2.0

O que é OAuth 2.0?

- Um padrão de autorização que permite que aplicativos de terceiros acessem recursos de um usuário em um servidor, sem precisar compartilhar suas credenciais (nome de usuário e senha).
- Baseado em "tokens de acesso", que concedem permissões específicas e limitadas a um aplicativo.
- Suporta diferentes tipos de aplicativos (web, mobile, desktop) e fluxos de autorização.

OAuth 2.0

Como funciona?

- Requisição: O aplicativo (cliente) solicita acesso a recursos do usuário (dono do recurso) em um servidor (provedor de recursos).
- Redirecionamento: O usuário é redirecionado para o servidor, onde faz login e autoriza o aplicativo a acessar seus recursos.
- Token de acesso: O servidor emite um token de acesso para o aplicativo, que é usado para acessar os recursos protegidos.
- Acesso: O aplicativo utiliza o token de acesso para solicitar os recursos do servidor em nome do usuário.

OAuth 2.0

Fluxos de autorização

- Fluxo de código de autorização: Utilizado para aplicativos web, onde o código de autorização é trocado por um token de acesso.
- Fluxo implícito: Utilizado para aplicativos móveis e JavaScript, onde o token de acesso é retornado diretamente ao aplicativo.
- Fluxo de credenciais do cliente: Utilizado para aplicativos que acessam recursos em nome próprio, sem um usuário específico.

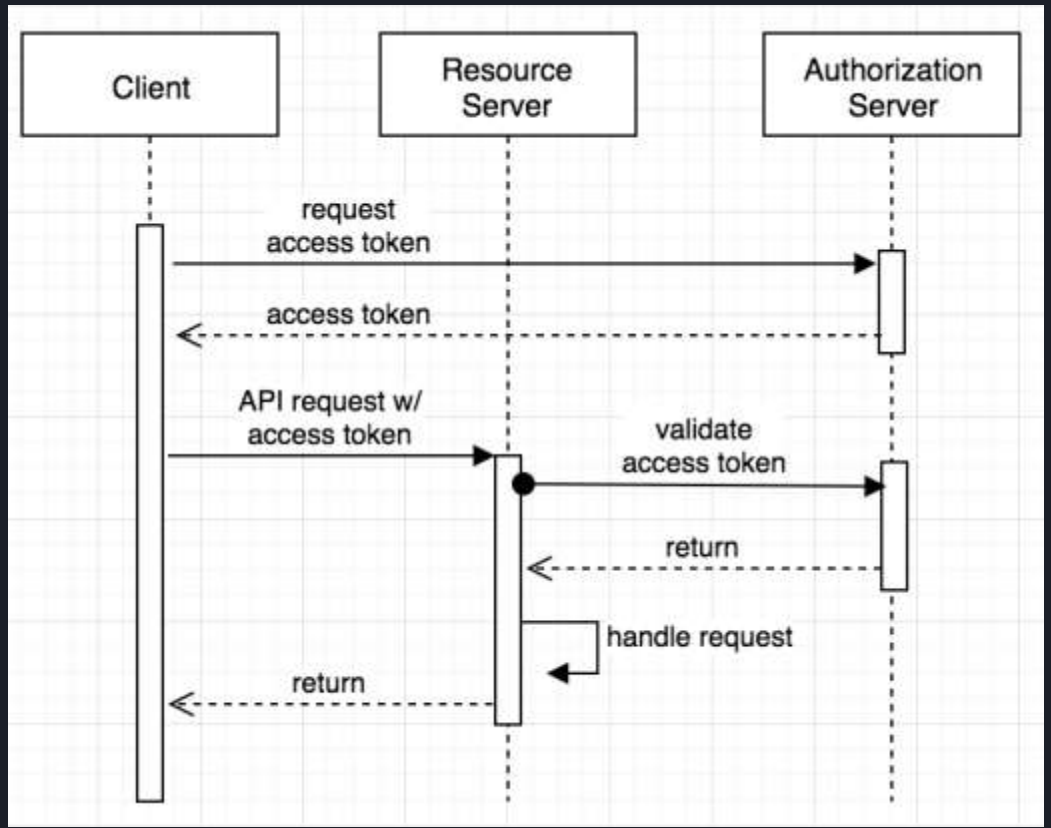
OAuth 2.1

Principais mudanças e melhorias

- O OAuth 2.1 não introduz novas funcionalidades, mas sim **consolida as melhores práticas** e extensões do OAuth 2.0 em um único padrão
- **Fim do fluxo implícito**: O fluxo implícito, que retornava o token de acesso diretamente ao cliente, foi considerado inseguro e removido do OAuth 2.1.
- **Obrigatoriedade do PKCE**: O PKCE, uma extensão que protege contra ataques de interceptação de código de autorização, agora é obrigatório para todos os clientes que utilizam o fluxo de código de autorização.
- **Restrições no uso de refresh tokens**: O OAuth 2.1 introduz restrições no uso de refresh tokens, que são utilizados para obter novos tokens de acesso sem precisar que o usuário conceda autorização novamente.

OAuth 2.0

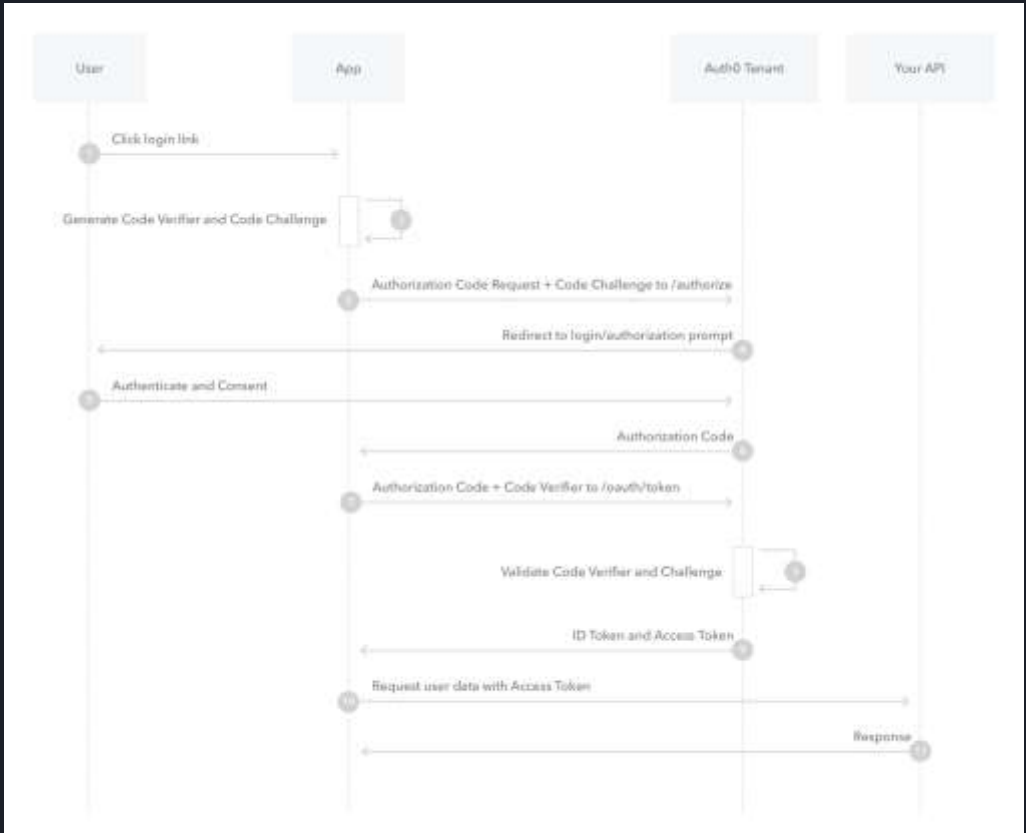
Fluxo de autorização – Credenciais do Cliente



ada

OAuth 2.0

Fluxo de autorização – Fluxo de código de autorização



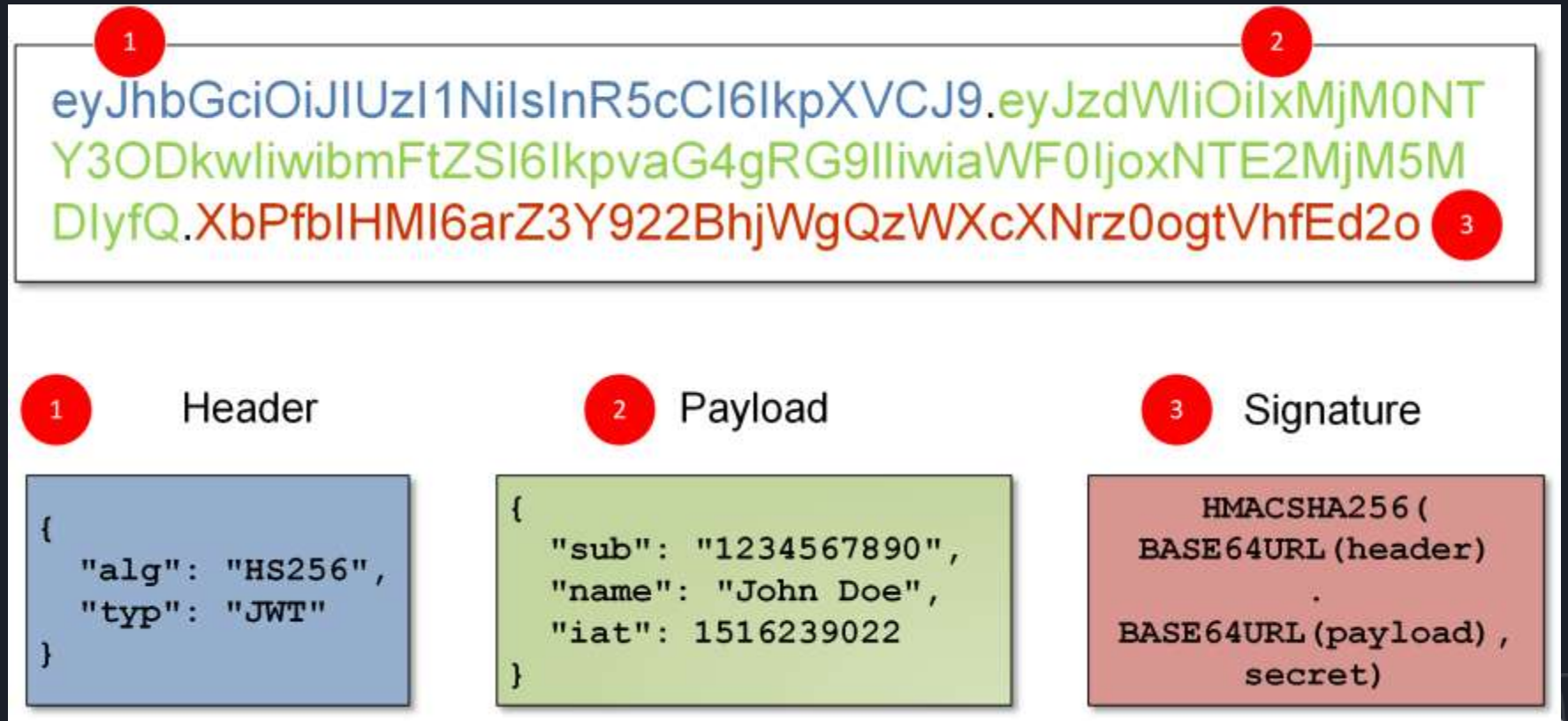
JWT

JSON Web Token

- JSON Web Token (JWT) é um padrão aberto (RFC 7519) que define uma maneira compacta e independente de transmitir informações com segurança entre as partes como um objeto JSON.
- Essas informações podem ser verificadas e confiáveis porque são assinadas digitalmente. Os JWTs podem ser assinados usando um segredo (com o algoritmo HMAC) ou um par de chaves pública/privada usando RSA ou ECDSA.

JWT

JSON Web Token



https://www.menti.com/aljs4wxbmab9

Obrig.ada