at_S15_A3_SL06_backend

Roteiro de atividade prática

Nome: _____ Turma: _____

Título da atividade: Criação de APIs seguras

Pontos principais do conteúdo:

Para garantir que uma API (*application programming interface* – interface de programação de aplicação) seja segura, é essencial implementar técnicas que protejam o acesso e o fluxo de informações entre o cliente e o servidor. Dois conceitos fundamentais para garantir essa segurança são a **autenticação** e a **autorização**, seguidos pela **criptografia de dados** para proteger as informações em trânsito.

Autenticação é o processo de verificar a identidade de um usuário ou aplicação, garantindo que quem está tentando acessar a API é quem realmente diz ser. Já a autorização determina quais recursos ou operações um usuário autenticado pode acessar. Finalmente, a criptografia protege os dados, garantindo que, mesmo que sejam interceptados, não possam ser lidos por terceiros.

Vamos explorar esses conceitos de forma prática e entender como aplicá-los para proteger suas APIs.

Explicação detalhada

Autenticação e autorização

- **Autenticação:** pode ser implementada utilizando métodos como **JWT (JSON Web Token)**, **OAuth** ou **chaves de API**. Por exemplo, em um

sistema de entrega de alimentos, cada cliente que acessa a API precisa fornecer um *token* válido para garantir que a solicitação seja de um usuário legítimo.
- **Autorização**: após a autenticação, a API verifica se o usuário tem permissão para realizar a operação solicitada. Em nosso exemplo, um cliente pode visualizar seu histórico de pedidos, mas apenas um administrador pode cancelar pedidos.

Criptografia de dados

- Para proteger os dados transmitidos entre o cliente e o servidor, utilizamos **HTTPS** (*hypertext transfer protocol secure* – protocolo de transferência de hipertexto seguro). Isso garante que as informações enviadas, como detalhes de pagamento, estejam protegidas.
- Criptografia de senhas usando algoritmos como **bcrypt** garante que, mesmo que um banco de dados seja comprometido, as senhas dos usuários não sejam facilmente descobertas.

Exemplo prático:

Imagine que você está desenvolvendo uma API para uma plataforma de serviços financeiros que lida com dados confidenciais. Ao criar um *endpoint* para acessar o histórico financeiro de um usuário, é necessário:

- autenticar o usuário com JWT para garantir que ele esteja devidamente identificado;
- autorizar o usuário para verificar se ele tem permissão para visualizar esses dados;
- transmitir as informações via HTTPS para proteger os dados contra interceptação.

Situação-problema

Você foi contratado para desenvolver uma API segura para uma startup que gerencia pagamentos on-line. A prioridade é garantir que os dados dos

usuários estejam protegidos contra acessos não autorizados e que as transações sejam seguras.

Responda:

1. Explique o que é autenticação e por que ela é necessária em uma API que lida com informações sensíveis.

2. Descreva como a autorização é usada para controlar o acesso a recursos dentro de uma aplicação.

3. Identifique a importância do uso de HTTPS em APIs que transmitem dados financeiros.

4. Relacione os benefícios da criptografia de dados em repouso e em trânsito para proteger informações sensíveis.