sempre_alerta_doc_servidor

quinta-feira, 9 de novembro de 2023 12:07

◉ SESSÕES

- No painel, cada sessão dura 24 horas. O usuário é automaticamente direcionado para tela de acesso.
- Diferente do app que, uma vez logado, não há tempo para expirar.

◉ ESTRUTURA DE DADOS

▪ ACCOUNTS

Todas as contas dos usuários

userId (obrigatório, String): email do usuário
role: (obrigatório, String): tipo usuário: admin, equipe, autoridade, dev
instituteId (opcional, String): id da instituição que o usuário está cadastrado.

Todos os usuários precisam ser gravados com instituteId, e esse id é associado automaticamente no cadastro dentro do painel.

Acontece que quando é cadastrado o primeiro admin da instituição, ainda não há cadastrado um id de instituição. Neste caso, no primeiro acesso ao painel deste primeiro admin, haverá um form de cadastro de instituição obrigatório para ser associado a cada novo usuário cadastrado posteriormente.

Enquanto o admin sem um ID de instituição não preencher e salvar os dados da instituição, não será possível acesso ao painel principal e form de cadastro mostrado no acesso do painel.

Todos os cadastros posteriores, dentro do painel, serão associados ao ID da instituição criado pelo primeiro admin.

Uma vez criada a instituição, o ID é inalterado mesmo se o admin trocar o nome da instituição. Isso porque esse ID será a base para todos os usuários e alertas precisando ser sempre valor fixo.

▪ USERS

Todos os usuários e seus dados. De acordo com tipo usuário, pode haver dados específicos ou dados em comum como dados de cadastro (id, nome usuário, list de contatos, etc)

▪ INSTITUTES

◉ SEGURANÇA - PRINCIPAIS

▪ ENDPOINTS

- Todos os endpoints são protegidos com proteção CORS que bloqueia tanto sites externos e principalmente outras ferramentas REST ou de requerimento tipo servidor-a-servidor (tipo postman, reqbin, etc) que não estejam previamente autorizados.
-
- É usado JWT para iniciar sessões que expiram depois de um certo tempo. Certificar que o mesmo usuário que acessou é o

mesmo que está enviando as requisições durante cada sessão;
-
- Além disso, é checado se o ID e tipo de usuário é o mesmo do JWT a cada requisição de um endpoint iniciado dentro de uma sessão.

▪ SERVIDOR

- É usado Helmet para aumentar a segurança do servidor. Dentro as principais proteções extras, incluem:
- não enviar a origem do servidor (X-Powered-By)
- Mitiga ataques de clickjacking (https://en.wikipedia.org/wiki/Clickjacking) com X-Frame-Options
- Bloquear o carragamento de recursos cross-origin

- É basicamente usado com middlewares de segurança que verifica se o usuário tá autorizado e verificação de origem. Também usei proxy, usando o próprio site para ocultar a origem do servidor.

```
router.get("/list/all", mwCors, mwAuthSession, readUserListAllAPI);

router.post("/session/init", mwCors, mwAuthSession, loadAuthUserSessionData);
```

- Por exemplo, mesmo descobrindo o caminho relativo e tentar acessar um rota válida de API pública do servidor (https://semprealertasos.com/api/sys/waker) não é possível fazer a requisição, é redirecionado para tela inicial.

◉ POR QUE NÃO É USADO ENDPOINT PARA CRIAR?

- É usado o URD, uma variação do CRUD, para fazer as operações no banco de dados.

  U: criar ou atualizar
  R: ler
  D: deletar

- Na operação U, a função de criar acontece apenas no cadastro, todas as demais operações são de atualizar.

- Se isSignup for true, vai CRIAR um novo item. isSignup é false por padrão. Se já houver o ID usuário com dados, vai ATUALIZAR de acordo com os dados dos campos enviados.

- Esse modelo de operação permite maior flexibilidade para atualizar dados e criar um único endpoint para atualizar uma tabela ou documento;