Base de Segurança API e WebSite

Client

auth.middleware

Toda Requisição

Valida JWT

Valido?

Sim

Não

TokenExpiredError

Registra Auditoria de Acesso a Recurso

quota.middleware

O tempo do rate-limit e o max-rate-limit são configurados conforme configuração no POD

CACHE

Toda Requisição

Recupera quantos acessos está computado para este recurso a partir do IP de origem

Incrementa o cabeçalho de resposta com:

x-rate-limit
x-rate-limit-remaining
x-rate-limit-time

Verifica se a cota foi excedida

Excedido?

Sim

HTTP Status: 429
HTTP Message: Quota Exceeded

Não

CACHE

Incrementa a contagem de acesso ao recurso

Libera que o fluxo de navegação siga

API

login.service

Autentica Usuário

Tempo para reset da contagem e numero máximo de tentativas conforme configuração do POD

Autentica no CA

Sucesso

Sim

Não

Recupera Dados Adicionais e Retorna Dados do Usuário

DATA

Não

máximo de tentativas atingido?

Sim

TooManyLoginAttemptsError

Incrementa contador de tentativas inválidas

NotAuthorizedError

CACHE

Checa se há tentativas de login inválidas do cache

Email
Senha

User {
userId: string,
userName: string,
userPhone: string,
userEmail: string,
companyId: string,
companyName: string,
verificado: boolean,
securityPhrase:string,
}

Dispara envio de Token por SMS

{
smsToken,
userPhone
}

Autentica usuário

Autentica Usuário

sim

Token Valido?

Não

NotAuthorizedError

Valida Token SMS

Cache de sessão expira conforme configuração no pod

CACHE

Cria e Armazena Sessão no cache

Gera Token JWT

auth.controller

Refresh Token

Verifica se a assinatura do JWT

Extrai o id da sessão de dentro do JWT

Tenta recuperar a sessão do cache

CACHE

Cache de sessão expira conforme configuração no pod

Encontrou sessão?

Sim

Verifica se o token é o token que a sessão reconhece como o último válido

Não

Token é o esperado?

Não

Sim

Gera um novo token

Headers:
x-access-token: JWT
x-device-id: DeviceFingerprint

Body:
empty

HTTP Status: 401
HTTP Message: Impossible to Renew

HTTP Status: 200
Response Body: JWT

POST /api/v1/auth/refreshtoken

Send Token

envia token por SMS

levantou erro?

Não

Sim

Gera mensagem de SMS envia mostrando 4 ultimos digitos do telefone de destino

Headers:
x-device-id: DeviceFingerprint

Body:
email: login
password: senha

TooManyLoginAttemptsError

NotAuthorizedError

HTTP Status: 429
HTTP Message: Too Many Login Attempts

HTTP Status: 401
HTTP Message: Not Autorized

HTTP Status: 200
Response Body: { userPhone }

POST /api/v1/auth/sendtoken

Login

Faz login

Levantou Erro?

Não

Sim

Headers:
x-device-id: DeviceFingerprint
x-device-location: LatLong

Body:
email: login
password: senha
smsToken: token

TooManyLoginAttemptsError

NotAuthorizedError

HTTP Status: 429
HTTP Message: Too Many Login Attempts

HTTP Status: 401
HTTP Message: Not Autorized

HTTP Status: 200
Response Body: { JWT }

POST /api/v1/auth/login

app.component

acessa o portal

checa se há um token no LocalStore

token expirado ou ausente?

Sim

Não

Redireciona pra Login

Habilita Controles e vai para a url de destino

Criptografa a informação e atualiza Local Storage com Novo Token

login.component

login "Email First" com 2fa

verifica se possui permissão para usar localização do dispositivo

autorizou usar localização?

não

Habilita página instruindo a permitir acesso a localização

sim

Solicita e-mail

Solicita Senha

Exibe Mensagem de Erro

Solicita Token Receibido por SMS

Exibe Mensagem de Erro

Habilita Controles e redireciona pra a URL de destino

Criptografa a informação e atualiza Local Storage com Novo Token

login.service

Gera o fingerprint do dispositivo

Chama entrypoint de renovação de token na API

Retorna Insucesso

Retorna Sucesso

Gera (ou pega do cache) o fingerprint do dispositivo

Chama a API solicitando envio de token por SMS

Retorna Insucesso

Retorna Sucesso

Gera (ou pega do cache) o fingerprint do dispositivo

Pega a localização do dispositivo

Chama a API para fazer login

Retorna Insucesso

Retorna Sucesso

auth-guard

Sim

token expirado ou ausente?

Sim

Autoriza a Navegação

checa se há um token no LocalStore

app-routing

Navega pelo Sistema

Realiza verificação de segurança

Redireciona para a URL de destino

http.interceptor

Redireciona para tela de Login

Outros

Exibe mensagem de erro genérica

Intercepta todo retorno de requisição externa

401

429

Exibe Mensagem de Erro para cota excedida

HTTP STAUS CODES

200

Deixa o fluxo seguir sem alteração

Informações gerais da segurança

- No Navegador, os dados da sessão do usuário são criptografados. Para a chave é gerado um hash em SHA-256, o valor criptografado com AES usando como chave o fingerprint da máquina que é calculado em tempo real.
- Toda requisição envia o fingerprint do equipamento que está sendo usado pelo usuário, que é gerado baseado no hardware e base de software da máquina.
- Toda requisição de acesso (não login) envia a localização do equipamento. Não é possível navegar sem essa permissão habilitada.
- A localização é enviada em latitude/longitude, precisão e altitude (este último apenas se disponível).
- O frontend é minifyed e unglyfied.
- A comunicação client e server deve ser feita apenas po HTTPS
- Todo acesso a recurso gera uma entrada de auditoria
- São gerados logs de segurança para todas as operações de autenticação e autorização
- os usuários cadastrados devem ter e-mail com domínio registrado no cadastro da empresa
- A empresa pode registrar IPs em White ou BlackList
- Usuário deve cadastrar uma frase de segurança, ele será enviada em toda comunicação por e-mail
- Acessos de novos dispositivos (baseados no fingerprint) envia um e-mail para o usuário
- O usuário deve ativar seu usuário através do e-mail de ativação antes de poder acessar o sistema
- O logger faz obfuscação de automatica de dados sensíveis baseado em norme de propriedades e valores coincidentes com dados do objeto de sessão.