



AULA 08 - Vídeo 05

Desenvolvimento

Retomando de onde paramos, quando o perfil não for encontrado, a estrutura barra a criptografia com uma mensagem de erro, e também na parte de dar **match** mandamos a mensagem de token padrão que no momento definimos aqui na estrutura mesmo. Então o sistema faz a parte de encontrar o email e comparar para ver se é válido, nisso avaliamos se está de acordo ou se está inválido. Da mesma forma a senha.

Vale ressaltar que não é indicado entregar informações diretas a quem está logado, como por exemplo: “senha invalida”, assim, se for uma invasão, deixa claro o que deve-se buscar primeiramente.

```
if (match) {  
  
    //Devolver o token + ID do perfil + email  
    res.json({  
        token: "Fabrica de programador"  
        email: perfilEncontrado.usuario.email  
        perfil: perfilEncontrado._id  
    })  
}
```

Quando for chamado o “autenticar” na rota de login, devemos chama-lo no perfil.

```
router.post("", async (req, res) => {  
    const resposta = await loginService.autenticar(req.body)  
    res.json(resposta)  
    loginService.autenticar(req.body)  
})
```

(router)
alteramos no perfil de “req, res” para “usuário”.

```
autenticar: async (usuario) =>{
```

```
let usuario = usuario
```

E para devolvermos o **json** direto damos o **return no try catch** com throw para retornar.

```
else {
  throw{
    message:"erro ao efetuar login: credenciais invalidas"
  }
}
else {
  throw{
    message:"erro ao efetuar login: credenciais invalidas"
  }
}
else {
  throw({
    message:"erro ao efetuar login: faltando credenciais"
  })
}
else {
    message:"erro ao efetuar login: faltando credenciais"
  }
}
} catch (error) {
  throw{
    message: error.message
  }
}
```