



AULA 08 - Vídeo 08

Agora para parte do “token”, temos que criar a função de gerar o token, primeiramente começamos mudando o “verificationtoken” para “gerarToken”

```
module.exports = {  
  gerarToken: (usuario) =>{  
  
    let token = request.headers["token"]  
  
  }  
  
}
```

e para usarmos precisamos da biblioteca jsonwebtoken alem do .env

```
require('dotenv').config()  
const jwt =require('jsonwebtoken')
```

Para o json gerar o token, o jwt usará uma função chamada “sign” composto por payload(conteudo a ser inserido), secretOrPrivate(a chave privada) e options. Como pode dar um eventual erro, vamos usar o “try catch” na função, jogando para cima o erro.

```
module.exports = {  
  gerarToken: (usuario) =>{  
  
    try {  
      return jwt.sign(usuario, process.env.SECRET)  
    } catch (`ERRO: ${error.message}`) {  
      throw{  
        status: 500,  
        message: "erro na geração do token."  
      }  
    }  
  
  }  
}
```

```
}  
}
```

E para que o erro seja injetado vamos mexer no “middleware”

```
module.exports = {  
  verificar:(req, res, next) =>{  
    const token = req.headers.token  
    if(!token){  
      res.status(401).json({  
        message:"autenticação necessaria"  
      })  
    }else next()  
  }  
}
```

E quando tudo estiver correndo corretamente, o token irá fazer a requisição comparando o token em meio a criptografia para validar a entrada.

```
if(match){  
  const token = tokenUtil.gerarToken(perfilEncontrado.usuario)  
  return{  
    token:token,  
    email:perfilEncontrado.usuario.email,  
    perfil:perfilEncontrado._id  
  }  
}
```