



## AULA 08 - Vídeo 09

na aula anterior, fizemos a implementação do login na parte de geração do token, agora vamos fazer a parte da verificação do mesmo. Então antes de criarmos a verificação, ou melhor, para criarmos teremos que ir lá no “tokenUtil” usando a verificação do `jsonWebToken`.

```
verificarToken: (token) =>{  
  try {  
    return jwt.verify(token, process.env.SECRET)  
  } catch (`ERRO: ${error.message}`) {  
    throw{  
      status: 500,  
      message: "erro ao autenticar token invalido."  
    }  
  }  
}
```

Agora o nosso tokenUtil além de gerar ele também verifica se está tudo correto com o token.

na função “verificar”, vamos tirar a function next do else e retornar o token para ver se esta dentro dos conformes. O JSONWEBTOKEN trabalha da seguinte forma, ele verifica se temos o segredo e o payload para aí sim dar a requisição de descriptografar as credenciais.

```
else {  
  
  const retorno =tokenUtil.verificarToken(token)  
  console.log(JSON.stringify(retorno))  
}
```

Logo após capturando uma exceção de token invalido colocamos o try dentro do else, e caso ele verifique o token e de como valido ai cai no next().

```
try {  
    const retorno = tokenUtil.verificarToken(token)  
    console.log(JSON.stringify(retorno))  
    next()  
  
} catch (error) {  
    res.status(error.status).json({  
        message: error.message  
    })  
}
```