ROTEIRO PARA CRIAR AUTENTICAÇAO DE UMA API REST COM JWT

1. Criar sistema de login da API
2. NA pasta APIDEGAMES
3. Abrir o arquivo index.js
4. Inserir no banco de dados “uma tabela de usuários” para realizar a autenticação
5. users:[
6. {
7. id:1,
8. email:"adm@adm.com.br",
9. nome:"admin",
10. senha:"12345"
11. },
12. {
13. id:2,
14. email:"claudisnei@teste.com",
15. nome:"Claudisnei",
16. senha:"teste123"
17. }
18. ]
19. Criar uma rota para autencicação
20. /\*\*rota para autenticação \*/
21. app.post("/auth",(*req*,*res*)=>{
22. var {email, password} = *req*.body;
23. //validações
24. if(email != undefined){
25. //Verificando se o usuário é cadastrado
26. var user = DB.users.find(*us* => *us*.email == email);
27. if(user != undefined){
28. if(user.password === password){
29. *res*.status(200);
30. *res*.json({token: "Token provisório!"});
31. }else{
32. *res*.status(401);
33. *res*.json({err: "Email ou senha inválidos"});
34. }
35. }else{
36. *res*.status(404);
37. *res*.json({err: "Email ou senha inválidos"});
38. }
39. }else{
40. *res*.status(400);
41. *res*.json({err: "O email enviado é inválido."});
42. }
43. });
44. Para gerar o token temos que instalar a biblioteca jsonwebtoken
45. ‘npm install –save jsonwebtoken’
46. Criar uma variável para instanciar a jwt
47. Criar uma chave secreta
48. //instanciando a biblioteca jsonwebtoken
49. const jwt = require("jsonwebtoken");
50. //criando uma chave secreta para o jwt
51. const secretKeyJwt = "ffhhae@#$¨%fdd!@#4";
52. Gerar o token
53. Incluir as informações que vc quer carregar no token
54. /\*\*deve-se passar as informações necessárias
55. \* , a chave secreta e o prazo que irá expirar
56. \* o token
57. \*/
58. jwt.sign(
59. {
60. id: user.id,
61. email: user.email
62. },
63. secretKeyJwt,
64. {expiresIn:'48h'},(*err*, *token*)=>{
65. if(*err*){
66. *res*.status(400);
67. *res*.json({err: "Falha interna"});
68. }else{
69. *res*.status(200);
70. *res*.json({token: *token*});
71. }
72. }
73. )