ROTEIRO PARA UMA API RESTFULL COM O HATEOAS

1. O HATEOAS tem como finalidade desacoplar o back end e o Front end da aplicação através da utilização de links da API ao invés de usar diretamente o endereço da rota.
2. Na pasta sessao17
3. vamos incluir no arquivo index.js os links do HATEOAS, sendo que a escolha é individual, o desenvolvedor que escolherá quais links terão cada rota.
4. Dentro da rota ‘app.get(‘/games/:id’)....
5. /\*\*rota para devolver ao usuário o game pelo ID \*/
6. app.get('/game/:id',auth,(*req*,*res*)=>{
8. //tratando o id verificando se é numero
9. if(isNaN(*req*.params.id)){
10. *res*.sendStatus(400);
11. }else{
12. var id = parseInt(*req*.params.id);
13. //criando links para API RESTFULL
14. var HATEOAS = [
15. {
16. href:"http://localHost:4040/game/"+id,
17. method:"DELETE",
18. rel:"delete\_game"
19. },
20. {
21. href:"http://localHost:4040/game/"+id,
22. method:"PUT",
23. rel:"edit\_game"
24. },
25. {
26. href:"http://localHost:4040/games",
27. method:"GET",
28. rel:"get\_all\_games"
29. }
30. ]
32. //var game = DB.games.find( game =>  game.id == id );
33. Game.findOne({
34. where:{id:id}
35. }).then((*game*)=>{
36. if(*game* != undefined){
37. *res*.status(200);
38. *res*.json({game, \_links: HATEOAS});
39. }else{
40. *res*.sendStatus(404);
41. }
42. }).catch(*err*=>{
43. console.log("Erro: "+*err*);
44. });
45. }
46. });
47. Lembrando que os links serão adicionados as rotas escolhidas pelo desenvolvedor.