

API

1. **ws/login_token** (POST) → Obtenção do token de autorização de um utilizador quando realiza o login, envia username e password, se for autenticado com sucesso o django retorna o token
2. **admin/** → Página de administração do Django.
3. **ws/groups** (GET) **** → Retorna todos os tipos de grupos de utilizadores disponíveis
4. **ws/create_user** (POST - cria token) → Permite criar um utilizador depois do mesmo fazer registo na plataforma. Fornece, também, o token de utilizador caso o registo seja válido.
5. **ws/userupd** (PUT)**** → Update de um utilizador. Utilizado quando um gestor/admin altera o grupo de um determinado utilizador.
6. **ws/users** (GET) **** → Retorna todos os utilizadores.
7. **ws/user** (GET) **** → Retorna um utilizador específico a partir de um *id*.
8. **ws/get_user** (GET)**** → Retorna um utilizador específico a partir de um *username*.
9. **ws/pubstatusgetOne** (GET) → Retorna um tipo de estado de publicação a partir da sua descrição.
10. **ws/pubstatusgetAll** (GET) → Retorna todos os tipos de estados de publicação.
11. **ws/pubtopicsgetOne** (GET) → Retorna um tópico de publicação a partir do *id*.
12. **ws/pubtopicsgetByDescription** (GET) → Retorna um tópico de publicação a partir da descrição.
13. **ws/pubtopicsgetAll** (GET) → Retorna todos os tópicos de publicações.
14. **ws/pubtopicsenabledgetAll** (GET) → Retorna todos os tópicos de publicação que estão ativados. As publicações já realizadas de tópicos desativados continuam ativas, mas novas publicações com um tópico desativado não são permitidas.
15. **ws/pubtopicscreate** (POST)**** → Criação de um tópico.
16. **ws/pubtopicsupdate** (PUT) **** → Atualização da descrição de um tópico de uma publicação.
17. **ws/pubtopicsdisable** (PUT)**** → Desativação de um tópico de publicação.
18. **ws/pubtopicsenable** (PUT)**** → Ativação de um tópico de publicação.
19. **ws/pubs** (GET)**** → Retorna todas as publicações, independentemente do estado da mesma.
20. **ws/pub** (GET) → Retorna uma publicação a partir de um *id*.
21. **ws/pubcrate** (POST) **** → Criação de uma publicação.
22. **ws/pubupd** (PUT) **** → Atualização do estado de uma publicação (passar de 'Por Aprovar' para 'Aprovado' ou de 'Aprovado' para 'Arquivado').
23. **ws/comment** (GET) → Retorna um comentário a partir de um *id*.
24. **ws/comments** (GET) → Retorna todos os comentários.
25. **ws/commentsPublication** (GET) → Retorna todos os comentários de uma determinada publicação, a partir do *id* da última.
26. **ws/commentcre** (POST)**** → Criação de um comentário.
27. **ws/commentdel/int:id** (DELETE) **** → Remove um comentário a partir do seu *id*.
28. **ws/fav** (GET) → Verificação de um favorito a partir do seu *id*. (*não usado*)
29. **ws/favs** (GET) **** → Retorna todos os favoritos.

30. **ws/favcre** (POST) **** → Cria um favorito, associado a uma publicação e a um utilizador.
31. **ws/favdel/int:id** (DELETE) **** → Apaga um favorito de uma publicação, por parte de um utilizador.
32. **ws/getAuthorPublications** (GET)**** → Retorna a lista de publicações realizadas por um determinado autor, independentemente do estado da mesma.
33. **ws/getAuthorPublicationsFiled** (GET)**** → Retorna a lista de publicações arquivadas de um utilizador.
34. **ws/getPublicationsFiled** (GET)**** → Retorna a lista de todas as publicações arquivadas.
35. **ws/getAuthorPublicationsPendent** (GET)**** → Retorna a lista de publicações pendentes de um utilizador.
36. **ws/getPublicationsPendent** (GET)**** → Retorna a lista de todas as publicações arquivadas.
37. **ws/getAuthorPublicationsApproved** (GET)**** → Retorna a lista de publicações aprovadas de um utilizador.
38. **ws/getAuthorFavoritePublications** (GET)**** → Retorna a lista de publicações aprovadas de um autor que foram favoritas por utilizadores.
39. **ws/getPublicationsApproved** (GET) → Retorna a lista de todas as publicações aprovadas.
40. **ws/getSearchPublicationsApproved** (GET) → Retorna a lista de todas as publicações aprovadas com filtros de pesquisa.
41. **ws/getSearchPublicationsPendent** (GET) → Retorna a lista de todas as publicações pendentes com filtros de pesquisa.
42. **ws/getSearchPublicationsFilled** (GET) → Retorna a lista de todas as publicações arquivadas com filtros de pesquisa.
43. **ws/getSearchPublicationsApprovedByUser** (GET)**** → Retorna a lista de todas as publicações aprovadas de um determinado autor, por *id*.
44. **ws/getSearchPublicationsPendentByUser** (GET) **** → Retorna a lista de todas as publicações pendentes de um determinado autor, por *id*.
45. **ws/getSearchPublicationsFilledByUser** (GET)**** → Retorna a lista de todas as publicações arquivadas de um determinado autor, por *id*.
46. **ws/getSearchPublicationsFavorites** (GET) → Retorna a lista de publicações favoritas de um utilizador, com filtros de pesquisa.
47. **ws/checkIfFavorite** (GET) → Verifica se uma publicação está, ou não, na lista de favoritos de um utilizador.
48. **ws/getSearchUsers** (GET) → Procura de utilizadores com filtros de pesquisa.
49. **ws/getSearchUsersPossible** (GET)**** → Retorna a lista de utilizadores possíveis possíveis de atualizar (o seu grupo)
50. **ws/get_groupDescription** (GET) **** → Retorna a lista de grupos possíveis de atribuir aos utilizadores.

Legenda todas os urls com **** significa que tem que se enviar o token no header , porque no django é feito @authentication_classes([TokenAuthentication])
@permission_classes([IsAuthenticated])