
 main

[DjangoForensics](#) / [Group2](#) / [internetForensics](#) / [decorators.py](#) / [Jump to](#)




[Go to file](#) [...](#)

 MUTEIGbeatrice [UPDATES 2](#) ...

Latest commit 0343527 2 days ago [History](#)

1 contributor

32 lines (23 sloc) | 1017 Bytes

[Raw](#) [Blame](#)   

```
1 from django.http import HttpResponseRedirect
2 from django.shortcuts import redirect
3
4 # FUNCTION DECORATORS TO RESTRICT UNAUTHENTICATED USERS (TO ENABLE ACCESS CONTROL)
5 def unauthenticated_user(view_func):
6     def wrapper_func(request, *args, **kwargs):
7         if request.user.is_authenticated:
8             return redirect('mainpage')
9         else:
10            return view_func(request, *args, **kwargs)
11
12     return wrapper_func
13
14 # FUNCTION DECORATORS FOR AUTHENTICATED USERS (TO ENABLE ACCESS CONTROL AS PER THE USER ROLES)
15 def allowed_users(allowed_roles=[]):
16     def decorator(view_func):
17         def wrapper_func(request, *args, **kwargs):
18
19             group = None
20             if request.user.groups.exists():
21                 group = request.user.groups.all()[0].name
22
23             if group in allowed_roles:
24                 return view_func(request, *args, **kwargs)
25             else:
26                 return HttpResponseRedirect('You are not authorized to access this page')
27         return wrapper_func
28     return decorator
29
30
31
```