

Django CSRF Exempt Class Based Views

Posted on 2015-06-10 by Zach Stevenson

When you are building APIs, there are many times when the Cross-Site Request Forgery (CSRF) limitations of Django really set you back. What's more frustrating is when you can't find the right information to solve the problems that are right in front of you!

I like using Class based Views in Django instead of the function based views. So when I was making a simple API that returned JSON I was surprised to find that there were very few places that talked about CSRF Exempt methods for Class based views. Every source I found just kept mentioning to use the function decorator `@csrf_exempt` from `django.views.decorators.csrf`. It turned out that the simplicity of that suggestion did not quite work the way I planned. I did, however, find a way to get it working, but my other issue was that I had maybe 15 or so classes that had at least 2 functions (get/post/delete) on them, so copying the same decorator over each of the functions felt like terrible bloat.

Instead, I decided to make a base class that everything extended from. With Python's multiple-inheritance model this was the obvious choice.

```
1  from django.utils.decorators import method_decorator
2  from django.views.generic import View
3  from django.views.decorators.csrf import csrf_exempt
4
5  class CSRFExemptView(View):
6      @method_decorator(csrf_exempt)
7      def dispatch(self, *args, **kwargs):
8          return super(CSRFExemptView, self).dispatch(*args, **kwargs)
9
10 class APIClassExample(CSRFExemptView):
11     #This function will be run without CSRF Checks
12     def get(self, request, *args, **kwargs):
13         pass
```

Hopefully someone else finds this useful.

Posted in [Development](#), [Django](#), [Python](#).
