



# Django Framework do Zero

## Como Criar um Aplicativo Web

Canal [youtube.com/@VamosCodar](https://www.youtube.com/@VamosCodar) - Wanderson Reis

## 32 - Usar Grupos para Controle de Acesso

### Requisitos

- 1) Aulas 01 a 31 - Django Framework do Zero - Playlist:  
<https://www.youtube.com/playlist?list=PLFOqHo8oljzewcT23HCxJV0xWO451CTJe>

### Introdução

Nesta aula vamos aprender a usar controle de acesso usando grupos. É um método simples para limitar funções a determinados usuários que pertencem a um ou mais grupos.

#### Nesta aula:

- 1) Implementar um decorator que valida grupo
- 2) Aplicar o decorator para limitar o acesso via grupo

**Observação:** Lembre-se de estar com o ambiente virtual ativado (venv).

**Preparação:** Acesse a área administrativa com um super usuário, acesso “grupos” crie um novo grupo chamado “Editor”.

### Passo 1: Implementar um decorator que valida grupo

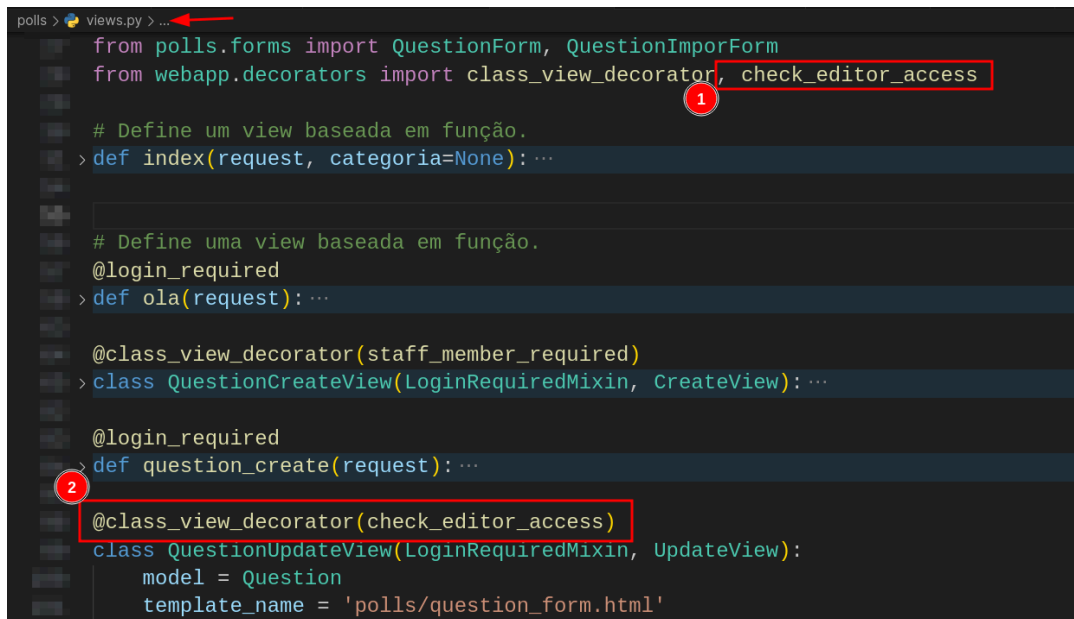
Editar o arquivo `django-do-zero/webapp/decorators.py` e acrescente o novo decorator de acordo com o mostrado na imagem:

```
webapp > decorators.py > ...  
> def class_view_decorator(function_decorator): ...  
  
def check_editor_access(view_function):  
    def wrapper(request, *args, **kwargs):  
        if request.user.groups.filter(name='Editor').exists():  
            return view_function(request, *args, **kwargs)  
  
        return HttpResponseForbidden('Você não tem permissão para acessar este recurso.')  
    return wrapper
```

(salve o arquivo).

## Passo 2: Aplicar o decorator para limitar o acesso via grupo

Editar o arquivo **django-do-zero/polls/views.py** e altere o código de acordo com a próxima imagem:



```
polls > views.py > ...  
from polls.forms import QuestionForm, QuestionImportForm  
from webapp.decorators import class_view_decorator, check_editor_access  
  
# Define um view baseada em função.  
> def index(request, categoria=None): ...  
  
# Define uma view baseada em função.  
@login_required  
> def ola(request): ...  
  
@class_view_decorator(staff_member_required)  
> class QuestionCreateView(LoginRequiredMixin, CreateView): ...  
  
@login_required  
> def question_create(request): ...  
  
@class_view_decorator(check_editor_access)  
class QuestionUpdateView(LoginRequiredMixin, UpdateView):  
    model = Question  
    template_name = 'polls/question_form.html'
```

(salve o arquivo).

Neste cenário vamos ter dois tipos de limitação / exigências diferentes uma para criar novas enquetes (staff da aula anterior) e agora para editar uma enquete (pergunta) será necessário o usuário da sessão pertencer ao grupo “Editor”

Em seguida suba o servidor de desenvolvimento.

**python manage.py runserver (ENTER)**

Para testar, acesse a função de editar uma enquete e tente modificá-la. Se necessário, acesse a área administrativa com um super usuário para atribuir o perfil (papel) de Editor a usuário que terá permissão / acesso para editar enquetes. Consulte a aula em vídeo para mais detalhes.