

Manager

Study link

<https://www.youtube.com/watch?v=eGVBCizGUfM&t=83s>

Using manager you can write query in models.py and call that query in views.py

To do that you need to create a class `PublishedManager(models.Manager)` in which you write query and then link that class to `Post` class using following lines

```
objects = models.Manager() # default Manager
published = PublishedManager() # custom model manager
```

And then in views.py

```
data=Post.published.all()
```

Example social media

models.py

```
from django.db import models
```

```
class PublishedManager(models.Manager):
    def get_queryset(self):
        return super(PublishedManager, self).get_queryset().filter(status='published')
```

```
class Post(models.Model):
```

```
    objects = models.Manager() # default Manager
    published = PublishedManager() # custom model manager
```

```
    STATUS_CHOICES=(
        ('draft', 'Draft'),
        ('published', 'Published')
    )
```

```
    title = models.CharField(max_length=200)
    slug = models.SlugField(max_length=100)
    author = models.ForeignKey(User, related_name="blog_posts", on_delete =
models.DO_NOTHING,)
    body = models.TextField()
    created = models.DateTimeField(auto_now_add=True) # Automatically set the field to now when
the object is first created.
    updated = models.DateTimeField(auto_now=True) #Automatically set the field to now every time
the object is saved.
    status = models.CharField(max_length=100 ,choices=STATUS_CHOICES, default='draft')

    def __str__(self):
        return self.title
```

Views.py

```
def post_list(request):
```

```
    data=Post.published.all()
```

Manager

```
context={
  'post' : data,
}
return render(request, 'blog/post_list.html', context)
```