**1. Django 如何使用 Redis?**

这里用到了一个包 django-redis ：<https://github.com/niwinz/django-redis>

使用方法和简单，在你的环境中运行 pip **install** django-redis 就可以安装它。

在你的 settings 文件中加入：

CACHES = {

"default": {

"BACKEND": "django\_redis.cache.RedisCache",

"LOCATION": "redis://127.0.0.1:6379/1",

"OPTIONS": {

"CLIENT\_CLASS": "django\_redis.client.DefaultClient",

}

}

}

这样就在你的 Django 项目中配置好了

要确保你的系统中安装了 Redis  
可以用 systemctl status redis 来查看 Redis 的运行情况

**代码部分**

***# 代码片段***

***# 导入包***

**from** django.core.cache **import** cache

***# 设置一个 key***

key = 'article'

***# 判断 key 是否存在 cache 中， 存在则在 cache 中取， 不存在则查询数据库***

**if** key **in** cache:

articles = cache.**get**(key)

**else**:

***# 查询数据库***

articles = Article.objects.all()

***# 将查询的内容 和 key 作为键值对存入 cache， 并设置缓存时间***

cache.**set**(key, articles, 15 \* 60)

为了提升网站的性能，加一层缓存是少不了的，由于之前做的东西是用django写的，所以就介绍一下django的缓存系统。

又由于我还使用了redis作为消息队列的后端，这里就不在介绍其他内存数据库，直接使用redis。

# django使用redis作为缓存

## redis

### 安装

由于笔者使用的是Ubuntu系统，安装较为简单，其他系统请自行查看官方文档。

$ sudo apt-get install redis-server

### 测试

使用redis命令行客户端测试一下是否成功

$ redis-cli

能出现以下结果即为成功

redis 127.0.0.1:6379>

其他的redis知识这里不左介绍，如果有兴趣，可以自行了解

## django使用redis

### 安装django-redis

为了让django能够使用redis作为缓存，有人开发了一个库来帮我们实现。[django-redis](https://link.jianshu.com?t=https:/github.com/niwinz/django-redis)，我们需要安装一下才能使用redis

在终端中输入一下命令进行安装(pip方式)

$ pip install django-redis

### 配置django-redis

在你项目下的**setting.py**中添加如下代码即可

CACHES = {

"default": {

"BACKEND": "django\_redis.cache.RedisCache",

"LOCATION": "redis://127.0.0.1:6379/1",

"OPTIONS": {

"CLIENT\_CLASS": "django\_redis.client.DefaultClient",

}

}

}

这样就能开始使用了，如果你想添加修改更多配置，想让其更适合自己的项目一点，可以自己去查看[官方文档](https://link.jianshu.com?t=http:/django-redis-chs.readthedocs.io/zh_CN/latest/)

这里随便配置一下**session backend**，同样在下面添加

SESSION\_ENGINE = "django.contrib.sessions.backends.cache"

SESSION\_CACHE\_ALIAS = "default"

### 使用缓存

在setting中配置好上述配置之后我们就可以开始使用了，django中自带了几个常用的修饰器，可以视情况使用。  
比如：

* django.views.decorators.cache.cache\_page（）针对每个视图的缓存

from django.views.decorators.cache import cache\_page

@cache\_page(60 \* 15)

def my\_view(request):

...

* 在setting添加配置，对全站进行缓存

MIDDLEWARE = [

'django.middleware.cache.UpdateCacheMiddleware',

'django.middleware.common.CommonMiddleware',

'django.middleware.cache.FetchFromCacheMiddleware',

]

* 还可以调用低级api，直接操作缓存。

>>> cache.set('my\_key', 'hello, world!', 30)

>>> cache.get('my\_key')

'hello, world!'