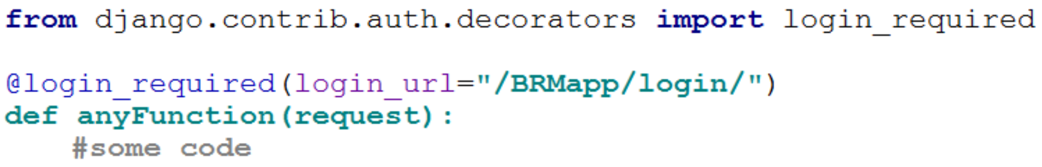
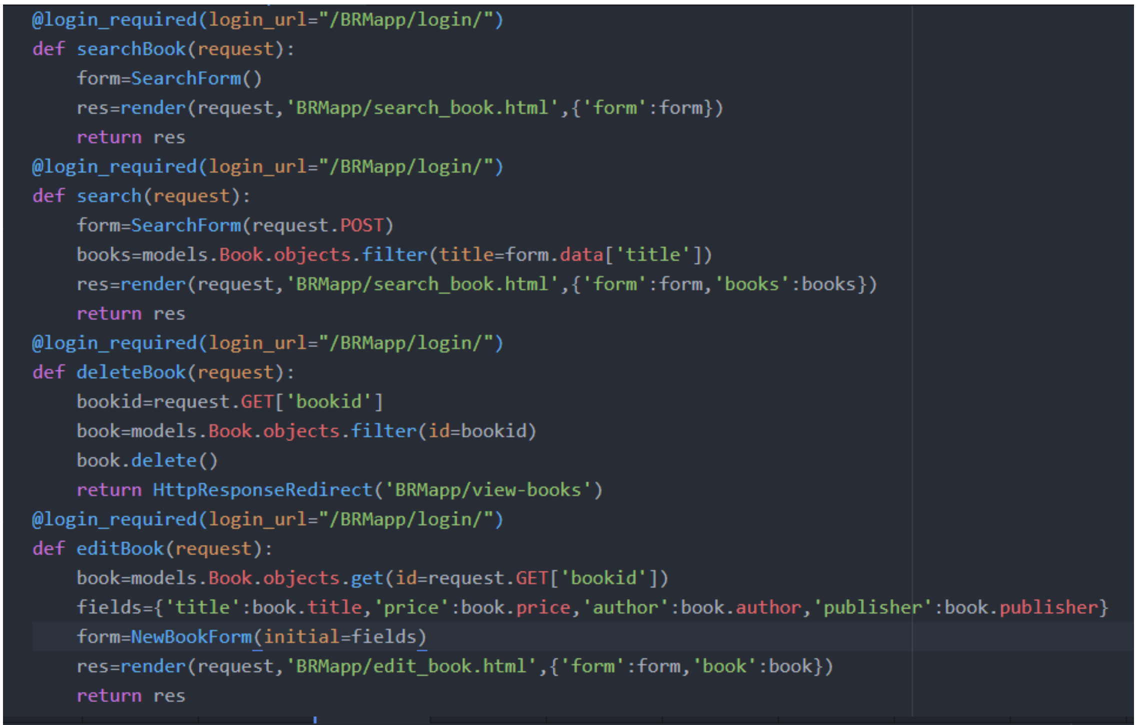
Overview | Restricting User to access pages without Login

Restricting user to access pages without login

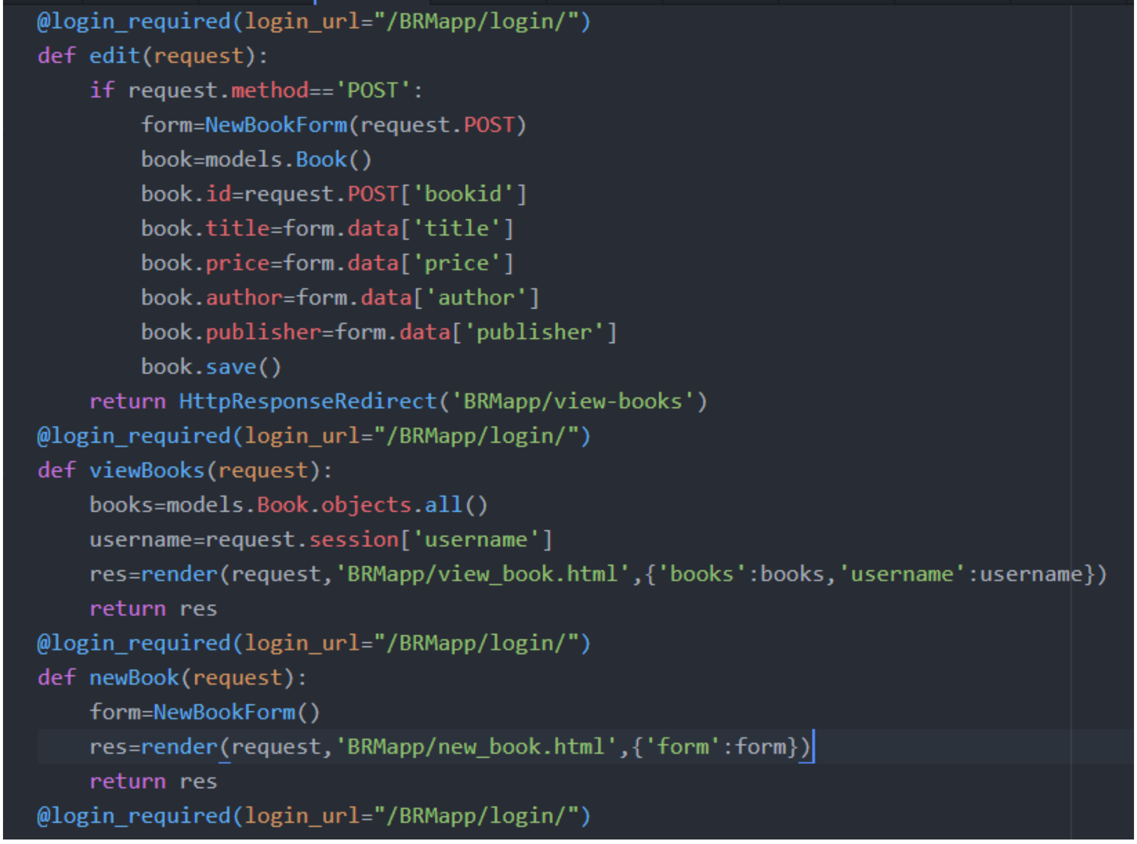
- If user knows the url, then he can access the page without logged in.
- Restricting user to access app pages without authentication
- Use decorator for every view, you want to restrict a user who send request without authenticated from login



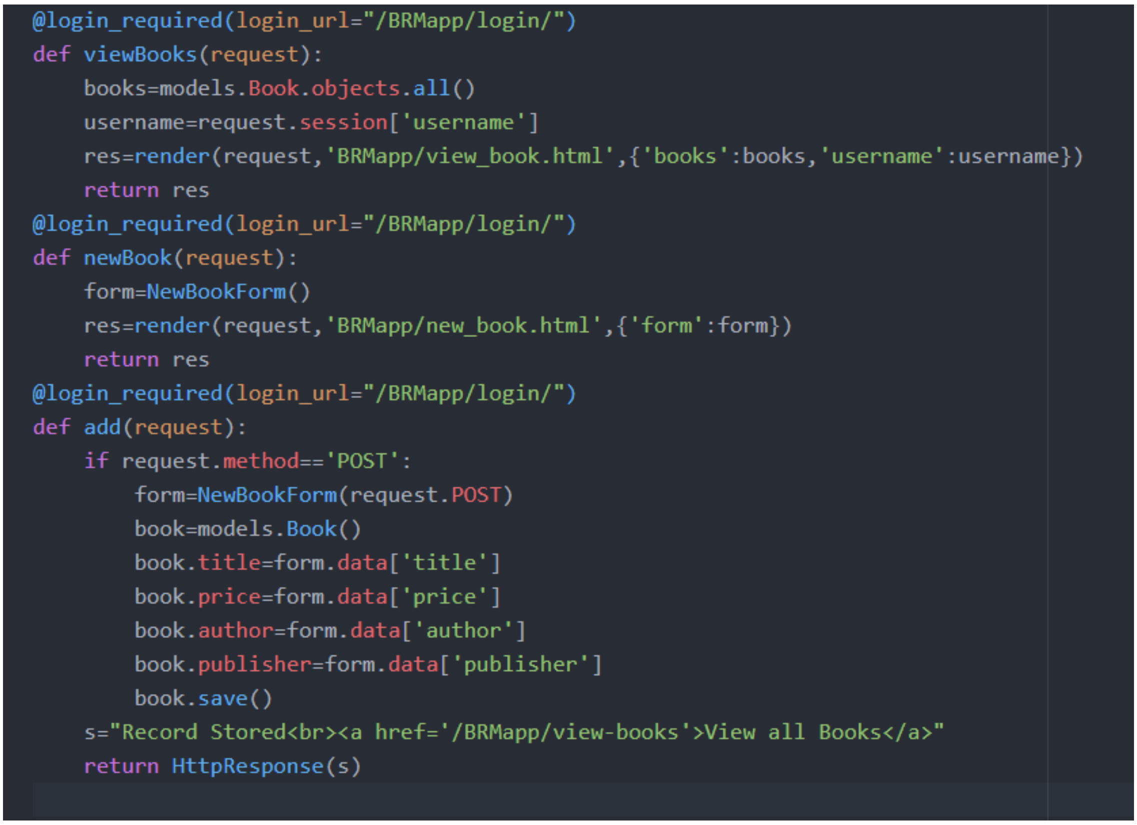
```
from django.contrib.auth.decorators import login_required

@login_required(login_url="/BRMapp/login/")
def anyFunction(request):
    #some code
```



```
@login_required(login_url="/BRMapp/login/")
def searchBook(request):
    form=SearchForm()
    res=render(request,'BRMapp/search_book.html',{'form':form})
    return res
@login_required(login_url="/BRMapp/login/")
def search(request):
    form=SearchForm(request.POST)
    books=models.Book.objects.filter(title=form.data['title'])
    res=render(request,'BRMapp/search_book.html',{'form':form,'books':books})
    return res
@login_required(login_url="/BRMapp/login/")
def deleteBook(request):
    bookid=request.GET['bookid']
    book=models.Book.objects.filter(id=bookid)
    book.delete()
    return HttpResponseRedirect('BRMapp/view-books')
@login_required(login_url="/BRMapp/login/")
def editBook(request):
    book=models.Book.objects.get(id=request.GET['bookid'])
    fields={'title':book.title,'price':book.price,'author':book.author,'publisher':book.publisher}
    form=NewBookForm(initial=fields)
    res=render(request,'BRMapp/edit_book.html',{'form':form,'book':book})
    return res
```



```
@login_required(login_url="/BRMapp/login/")
def edit(request):
    if request.method=='POST':
        form=NewBookForm(request.POST)
        book=models.Book()
        book.id=request.POST['bookid']
        book.title=form.data['title']
        book.price=form.data['price']
        book.author=form.data['author']
        book.publisher=form.data['publisher']
        book.save()
    return HttpResponseRedirect('BRMapp/view-books')
@login_required(login_url="/BRMapp/login/")
def viewBooks(request):
    books=models.Book.objects.all()
    username=request.session['username']
    res=render(request,'BRMapp/view_book.html',{'books':books,'username':username})
    return res
@login_required(login_url="/BRMapp/login/")
def newBook(request):
    form=NewBookForm()
    res=render(request,'BRMapp/new_book.html',{'form':form})
    return res
@login_required(login_url="/BRMapp/login/")
```



```python
@login_required(login_url="/BRMapp/login/")
def viewBooks(request):
    books=models.Book.objects.all()
    username=request.session['username']
    res=render(request,'BRMapp/view_book.html',{'books':books,'username':username})
    return res
@login_required(login_url="/BRMapp/login/")
def newBook(request):
    form=NewBookForm()
    res=render(request,'BRMapp/new_book.html',{'form':form})
    return res
@login_required(login_url="/BRMapp/login/")
def add(request):
    if request.method=='POST':
        form=NewBookForm(request.POST)
        book=models.Book()
        book.title=form.data['title']
        book.price=form.data['price']
        book.author=form.data['author']
        book.publisher=form.data['publisher']
        book.save()
    s="Record Stored<br><a href='/BRMapp/view-books'>View all Books</a>"
    return HttpResponse(s)
```