**REST API (Django Rest Framework-DRF)**

**Session-16**

**Permission classes:**

* REST Framework will provide list of permission classes

1. AllowAny
2. IsAuthenticated
3. IsAdminUser
4. IsAuthenticatedorReadOnly
5. DjangoModelPermissions
6. DjangoModelPermissionsorAnonReadOnly
7. DjangoObjectPermissions

**AllowAny :**

* Any users can access API

**IsAuthenticated:**

* Authenticated users only can access API

**IsAdminUser** :

* Authenticated users and users must have staff status is true.

**IsAuthenticatedorReadOnly :**

* Authenticatedusers can access API and Unauthenticated users can only read but can not write.

**DjangoModelPermissions :**

* Users should be authenticated and must have enough permissions for read and write.

**DjangoModelPermissionsorAnonReadOnly :**

* If user is authenticated, he should have permissions for read and write,
* Id user is not authenticated he can read only.

**DjangoObjectPermissions :**

* User must be authenticate and user should have enough permission on particular objects.

**Working with SessionAuthentication:**

* This authentication scheme uses Django's default session backend for authentication.
* If successfully authenticated, SessionAuthentication provides the following credentials.
* request.user will be a Django User instance.
* request.auth will be None.
* Unauthenticated responses that are denied permission will result in an HTTP 403 Forbidden response.

**CustomPermissions:**

* To implement a custom permission, override BasePermission and implement either, or both, of the following methods:

.has\_permission(self, request, view)

.has\_object\_permission(self, request, view, obj)

* The methods should return True if the request should be granted access, and False otherwise.

**Settings.py:**

INSTALLED\_APPS = [  
 **'django.contrib.admin'**,  
 **'django.contrib.auth'**,  
 **'django.contrib.contenttypes'**,  
 **'django.contrib.sessions'**,  
 **'django.contrib.messages'**,  
 **'django.contrib.staticfiles'**,  
 **'myapp.apps.MyappConfig'**,  
 **'rest\_framework'**,  
  
 **'myapp7'**,  
]

**Models.py:**

**from** django.db **import** models  
  
*# Create your models here.***class** Manager(models.Model):  
 name=models.CharField(max\_length=20)  
 address=models.CharField(max\_length=20)  
 mail = models.CharField(max\_length=20)  
 age = models.IntegerField()

**admin.py:**

**from** django.contrib **import** admin  
**from** myapp7.models **import** Manager  
*# Register your models here.*@admin.register(Manager)  
**class** ManagerAdmin(admin.ModelAdmin):  
 list\_display = [**'id'**,**'name'**,**'address'**,**'mail'**,**'age'**]

**Now go to terminal and then type the following commands**

* Python manage.py makemigrations
* Python manage.py migrate
* Python manage.py createsuperuser
* Now run server: python manage.py runserver
* Now go to browser: <http://127.0.0.1:8000/admin>
* Insert few records

Create a new python file with the name serializers.py.

**serializer.py:**

**from** rest\_framework **import** serializers  
**from** myapp7.models **import** Manager  
  
**class** ManagerSerializer(serializers.ModelSerializer):  
 **class** Meta:  
 model=Manager  
 fields=[**'id'**,**'name'**,**'address'**,**'mail'**,**'age'**]

**views.py:**

**from** django.shortcuts **import** render  
**from** myapp7.models **import** Manager  
**from** myapp7.serializers **import** ManagerSerializer  
**from** rest\_framework **import** viewsets  
**from** rest\_framework.authentication **import** SessionAuthentication  
**from** rest\_framework.permissions **import** IsAuthenticatedOrReadOnly,DjangoModelPermissions,DjangoModelPermissionsOrAnonReadOnly  
  
**from** myapp7.custompermissions **import** CustomPermission  
  
*# Create your views here.***class** ManagerModelViewSet(viewsets.ModelViewSet):  
 queryset = Manager.objects.all()  
 serializer\_class = ManagerSerializer  
 authentication\_classes = [SessionAuthentication]  
 *#permission\_classes = [IsAuthenticatedOrReadOnly]  
 #permission\_classes = [DjangoModelPermissions]  
 #permission\_classes = [DjangoModelPermissionsOrAnonReadOnly]* permission\_classes = [CustomPermission]

**urls.py**:

**from** django.contrib **import** admin  
**from** django.urls **import** path,include  
**from** myapp7 **import** views  
**from** rest\_framework.routers **import** DefaultRouter  
  
*#create a router object*router=DefaultRouter()  
  
*#register your viewset with router*router.register(**'ManagerViewset'**,views.ManagerModelViewSet,basename=**'manager'**)  
  
urlpatterns = [  
 path(**'admin/'**, admin.site.urls),  
 path(**''**,include(router.urls)),  
 path(**'authentication/'**,include(**'rest\_framework.urls'**,namespace=**'rest\_framework'**))  
  
]

Note: create a new python file with the name custmpermissions.py

**custompermissions.py:**

**from** rest\_framework.permissions **import** BasePermission  
  
**class** CustomPermission(BasePermission):  
 **def** has\_permission(self, request, view):  
 **if** request.method==**'GET'**:  
 **return True  
 return False**