



WEB DEVELOPMENT

Lesson 13

CRUD using DRF

Wrapping API views

Wrapping API views

1. @api_view —FBV

```
from rest_framework.decorators import api_view
```

2. APIView —CBV

```
from rest_framework.views import APIView
```

- receive **Request** instance in view
- add context to **Response**
- method not allowed
- parsing error

Function Based Views

Rewriting our API using @api_view

```
from rest_framework.decorators import api_view
```

Class Based Views

Rewriting our API using class-based views

```
from rest_framework.views import APIView
```

Requests and Responses

Request

```
from rest_framework.request import Request
```

request.POST

- form-data
- POST

request.data

- attribute data
- POST, PUT, PATCH

Response

```
from rest_framework.response import Response
```

```
return Response(data)
```

Status code

```
from rest_framework import status
```

```
return Response(data, status=status.HTTP_200_OK)
```

Generic class-based views and Mixins

```
from rest_framework import mixins
from rest_framework import generics

class StudentList(generics.ListCreateAPIView):
    queryset = Student.objects.all()
    serializer_class = StudentSerializer
```

Views for User model

```
class UsersList(generics.ListAPIView):
```

```
...
```

```
class UsersDetail(generics.RetrieveAPIView):
```

```
...
```

Authentication & Permissions

DRF Authentication

1. BasicAuthentication
2. TokenAuthentication
3. SessionAuthentication
4. *Third party auth*

DRF Authentication

```
REST_FRAMEWORK = {  
    'DEFAULT_AUTHENTICATION_CLASSES': (  
        'rest_framework.authentication.BasicAuthentication',  
        'rest_framework.authentication.SessionAuthentication',  
        'rest_framework.authentication.TokenAuthentication',  
    )  
}
```

DRF TokenAuthentication

INSTALLED_APPS

```
rest_framework.authtoken
```

views.py

```
def login(request):  
    ...
```

urls.py

```
path('api/login/', views.login)
```


Permissions

```
from rest_framework.permissions import IsAuthenticated
from rest_framework.permissions import AllowAny
```

```
#view class
permission_classes = (IsAuthenticated, )
```

Authentications

```
from rest_framework.authentication import TokenAuthenticationfrom  
  
#view class  
authentication_classes = (TokenAuthenticationfrom, )
```

Authenticated user in view

```
def get_queryset(self):  
    return Model.objects.filter(owner=self.request.user)  
  
def perform_create(self, serializer):  
    serializer.save(created_by=self.request.user)
```

Questions?