Django REST Framework (DRF) API View Classes - Examples with Output

1. APIView - Low-Level Base Class

Example:

```
class HelloWorld(APIView):
def get(self, request):
return Response({"message": "Hello, world!"})
```

Output (GET /hello/):

```
{
"message": "Hello, world!"
}
```

2. Generic Views

a. CreateAPIView

POST /items/create/

Request:

```
{
"name": "Pen",
"price": 10
}
```

Response:

```
{
"id": 1,
"name": "Pen",
"price": 10
}
```

b. RetrieveAPIView

GET /items/1/

Response:

```
{
"id": 1,
"name": "Pen",
"price": 10
}
```

c. ListAPIView

GET /items/

```
[
{"id": 1, "name": "Pen", "price": 10},
{"id": 2, "name": "Pencil", "price": 5}
]
```

d. UpdateAPIView

PUT /items/1/

Request:

```
{
"name": "Pen",
"price": 12
}
```

Response:

```
{
```

```
"id": 1,

"name": "Pen",

"price": 12

}
```

e. DestroyAPIView

DELETE /items/1/

Response: 204 No Content

3. Combo Generic Views

a. ListCreateAPIView - GET/POST /items/

b. RetrieveUpdateDestroyAPIView - GET/PUT/PATCH/DELETE /items/<id>/

4. ViewSets

a. ModelViewSet - All CRUD operations at /items/ via routers

b. ReadOnlyModelViewSet - Only GET (list/retrieve)

5. Mixins + GenericAPIView

```
class ItemCustomView(...):

def get(self, request): return self.list(request)

def post(self, request): return self.create(request)
```

6. Function-Based View with @api_view

```
@api_view(['GET'])

def simple_view(request):

return Response({"msg": "Hi from FBV"})
```

GET /simple/

Response:

```
{

"msg": "Hi from FBV"

}
```