Django REST Framework "mixins" module Concept

- The mixins classes provide the "actions" methods that are used to provide the basic view behavior.
- Note that the mixins classes proide action methods rather than defining the handler methods, such as .list(), .retrieve(), .destroy()....
- This allows for more flexible composition of behavior.
- The mixin classes can be imported like below
- **from rest_framework.mixins import <required_mixins_classes>**
- Here we use both handler methods and action methods in class based views to handel corresponding requests like get, post, put and delete.
- For example,

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

Mixins_class		purpose	action methods	handler methods
ListModelMixin	----	Getting all records	.list()	get()
CreateModelMixin	----	Create new record	.create()	post()
RetrieveModelMixin	----	Getting Specific record	.retrieve()	get()
UpdateModelMixin	----	Update Specific record	.update()	put()
DestroyModelMixin	----	Delete Specific record	.destroy()	delete()

1. ListModelMixin:

- It provides .list(request) action method, that list out all available model instances.

2. CreateModelMixin:

- It provides .create() action method, that allows us to post a new model instance and also it will save the new model instance.

3. RetrieveModelMixin:

- It provides .retrieve() method, that returns an existing model instance as a response.

4.UpdateModelMixin:

- It provides .update() method, that allows us to modify the existing model instance, after modifying it also save the model instance.

5. DestroyModelMixin:

- It provides .destroy() method, that allow us to delete the existing model instance.

Different handler methods are,

1. get()
2. post()
3. put()
4. delete()

Addantages of using mixins module:

- mixins module is providing readymade templates for post and put operations purpose.
- So directly we can send plain text formatted data using html templates.
- No need to validating user sending data is JSON type or not.
- No need to checking requested data contains all the required fileds or not and if yes, valid type or not.
- No need to return the response explicitly to consumer using Response() method. So no need to import Response().
- No need to send status codes information along with response. Automatically mixins module will handle about status codes.

Create APIs using Django REST Framework mixins module concept

Step1 : Create Project Name like **Mixins_Project**

Step2 : Create appName like **Mixins_App**

Step3: Open Project using **Pycharm** IDE

Step4: Open **models.py** and Crete required models file code

Step5 : Create **serializers.py** file in side our ApplicationName and create required logics.

Step6: Open views.py file and Create required views logics

Step7: Open project level **urls.py** and create mapping url for application level urls.py.

Step8: Create urls.py file inside Application level and Create required urls.py code

Step9: Execute the **makemigrations** command to generate SQL code in migration file.

python manage.py makemigrations

Step10: Execute the **migrate** command to create the tables for models

python manage.py migrate

Step11: Execute the **createsuperuser** command for creating admin creadentials

python manage.py createsuperuser

Username: Srinivas
Email : srinivas@gmail.com
Password: admin123
Password (Again) : admin123

Step11: Execute the **runserver** command to run the server

python manage.py runserver

Step12: Now Login to the admin site and enter some product details

Step13: Now open new tab and enter ipaddress along with /api/product/

127.0.0.1:8000/api/product