1. **Introduction to API :** Application Programming Interface (Integration Medium which comes from other source)

2. **What is REST API, CRUD operation :**
   An API that follows REST architecture is called REST API.

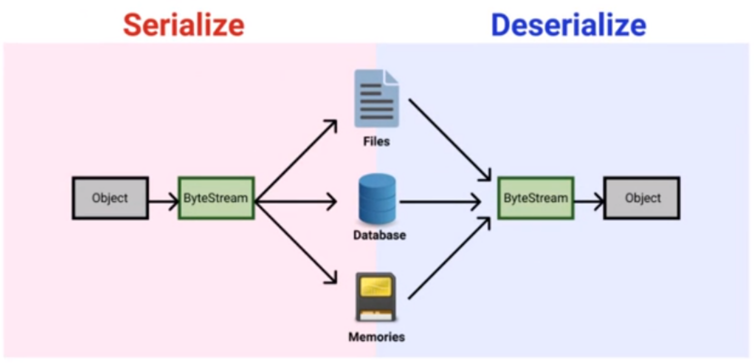
   REST → Representational State Transfer :
   a. Endpoint → url
   b. Method → http
   c. Headers → status code
   d. Data → information

   HTTP's four methods which are : **GET, POST, UPDATE, DELETE** altogether is called **CRUD Operation.**

   CREATE, READ/RETRIEVE , UPDATE, DELETE.

   <u>REQUESTs :</u>
   CREATE → POST Request
   READ / RETRIEVE → GET Request
   UPDATE → PUT Request
   DELETE → DELETE Request

3. **Api recap and Serializer : DRF** → Django REST Framework
   Api communicate through using JSON / XML format.



JSON → JavaScript Object Notation

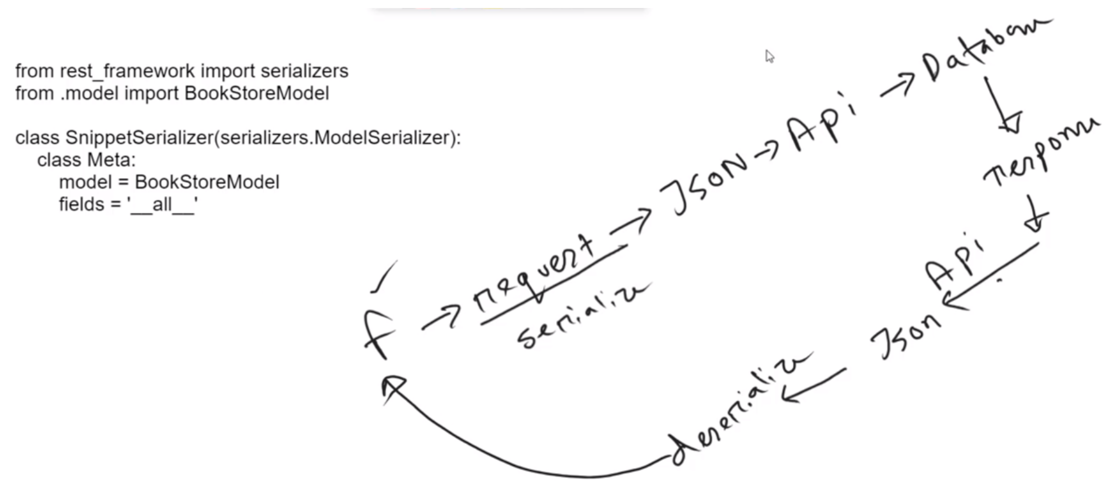
Serialize : Update in server using JSON format through API is called **Serialize.**
Deserialize : Getting Updated data from server through API is called **Deserialize**.

Deserialze & Serialize both are done by **Serializer.**

Frontend ← → **[API]** ← → Backend
Frontend →**[Serialize]** → Backend
Backend → **[Deserialize]** → Frontend

4. **Frontend and Backend Folder setup :**
   'rest_framework',
   ,'snippets',

5. **Making Serializer Class :**



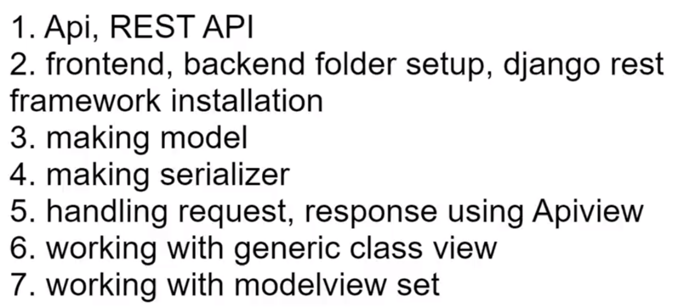
```
from rest_framework import serializers
from .model import BookStoreModel

class SnippetSerializer(serializers.ModelSerializer):
    class Meta:
        model = BookStoreModel
        fields = '__all__'
```

6. **Handling Request , Response using Apiview :**
   a. **Api View :** Long Cut
   b. **Generic View :** Medium Cut
   c. **View :** Short Cut

<u>REQUESTs</u> :
CREATE → POST Request
READ / RETRIEVE → GET Request
UPDATE → PUT Request
DELETE → DELETE Request

<u>RESPONSE</u> : YES / NO, ERROR's

7. **Handing CORS error :**
8. **Update and Delete Request Using Apiview :**
9. **Working with Generic Class view :**
10. **Working with Model Viewset and Router :** Multiple Request
11. **Summary :**



1. Api, REST API
2. frontend, backend folder setup, django rest framework installation
3. making model
4. making serializer
5. handling request, response using Apiview
6. working with generic class view
7. working with modelview set