What is Meta class :-

Meta is nothing but data description about the existing data, meta class gives some extra information about the existing class like what is our model name, what are the model fields...... to the current class

APIView :-

If we are ApiView class to creating the our user defined views like for business logics: We are going to write id & Non-id based classes. For executing these two classes id & Non-id based classes we are going to create.

Generally when users trying to create (or) update the data in our database as a API developer, we need to check two conditions

Condition-1 :- user sending data is JSON type data (or) Not ?

condition-2 :- user sending data is containing all the fields with data. and valid type or not, we need to check

By using this 'APIView' class only data is JSON type or type not is checking automatically, but not checking it is valid type or not so manually by using 'is_valid()' we are going to check data is valid (or) not.

If data is not valid it returns 'exceptions', if valid it saves into database.

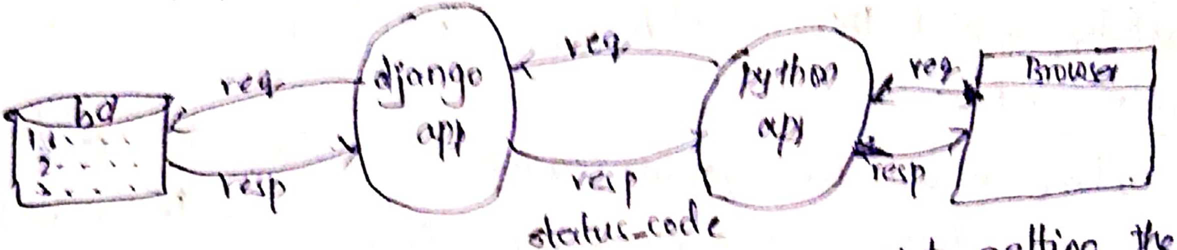
status codes :-

→ when we are communicating between two applications, "status code" concept is more important.

→ when we are sending one request from partner application to provider application, Then provider application given response back

→ so for every request we are getting the corresponding response back, may be this ~~res~~ response is succesful info (or) may be this response ~~failure msgs~~ failure info

→ for every response we are getting one respective status ~~do~~code based on the status codes we will handle the all exceptions & give the proper msgs



* Here user ~~python~~ sending request to 'python app' to getting the data from data base, then pythoding the request to django to take the date from django-database

* Now django app return response to the python application along with response it returns respective status code also

* python developers based on the status code they will return respective msgs to the user, by checking the status code values

* To return all these status codes, RestApi providing one module name as 'status' module. this status module providing so many status code classess to representing the respective information

eg:-
```
from rest_framework import status
status.Http_200_ok
status.Http_201_created
```

status codes with Description:-

* 200 ok → when we are sending request to gett all the records (or) single record, and updating the existing record. if succesfully info coming the 200 will coming.

sep!

201 created :-

when we are sending the request to creat the date into ~~the~~ data base, if successfully created date, then it returns '201 created'

204 no content :-

when we are sending the request to delete the date, it successfully deleted then it returns 204, no content

400 badrequest :-

when we are sending the request for creating and updating the data, if successfully not created ~~to~~ then it returns 400 'badrequest'

404 NotFound :-

when we are sending the request to searching for particular resource, but the resource are not available then it returns '404 - NOTFOUND'

401 → UnAuthorized → user id, password

403 → Forbidden
↓
when we are missing the 'csrf-token' when submitting the form

405 → MethodNotAllowed
↓
when we are sending request by using one http method and sending another type url
eg: 127.0.0.1:8000/api/emp/—

500 → Server Error :-

All 200 → successfull

All 300 → redirectional information

All 400 → clientside exception

All 500 → serverside exception