**REST API**

API stands for *Application Programming Interface*

To access some functionality of some application we need an interface called API

To communicate one application with another application we use API

Gui interface

**WEB API / WEB SERVICES:**

Communication between two applications over web is known as web API

Ex: when we login to an app we see *login with google, login with Facebook*

**REST: (RE**presentational **S**tate **Transfer)**

REST is an architecture which defines several guidelines to develop web APIs very easily.

Limitations

1. Over fetching and under fetching
2. Does not fit for complex requests
3. Satelessness
4. Does not for applications with frequent updates.

**REST API / RESTful API**

The APIs which are developed by using REST architecture is known as REST API.

**DJANGO REST FRAMEWORK:**

The framework that provides several tools to develop RESTful API very easily is known as Django rest framework.

By using some methods we use to communicate with application

SOAP( Simple Object Access Protocol)

* XML based
* Heavy height so they need high bandwidth
* Very complex
* Parsing xml is slowdown ( parsing means converting)
* Data is more secure than restfulapi
* It supports HTTP, SMTP, FTP

RESTful (Representational State Transfer)

* It supports only HTTP request and response
* It provides the data in JSON format
* JSON is a Light weight
* Performance is more
* Human understandable format
* Less secured

Types of API 1) Private (within a group)

2) Partner (developing an api for our business partners only )

3) Public (this APIs can access by anyone)

**Requests** follow HTTP methods

**HTTP METHODS:**

1. **GET**

To get the data from database

1. **POST**

To post/create new data into database

1. **PUT**

to update a resource/data

1. **PATCH**

Partial update of a resource/data

1. **DELETE**

to delete a resource/data

**DATABASE CRUD OPERATIONS:**

**1) CREATE post**

**2) RETRIVE/READ get**

**3) UPDATE put**

**4) DELETE delete**

Json -----> JavaScript Object Notation

Any software application can understand the Json format Json is both human and machine understandable language

Python contains inbuilt module: JSON module

Inside JSON module there are two functions are there: 1) dumps()

2) loads()

Dumps () function is used to convert python dic to Json data

Loads () function is used to convert Json to python dic

**Python dic and Json are similar but different languages**

* We can send HTTP request from command line by using httpie and curl

Pip install httpie

http <http://127.0.0.1:8000/api/>

**VALIDATORS:**

1. **Field level validators**

These validators check only a particular field and check that is correct or not.

def validate\_name(self, value):  
 if len(value)>15 or len(value)<4:  
 raise serializers.ValidationError("name is too low enter a valid name")  
 else:  
 return value

1. **Object level validators**

When we validate more than one field we use object level validators. By using this we can validate entire object.

def validate(self,data):  
 if data['age']<18:  
 raise serializers.ValidationError("age should me more than 18 ,enter a valid age")  
 if data['name']==data['section']:  
 raise serializers.ValidationError("name and description should not be equal ")  
 else:  
 return data