26/04/2017

5. MTTP code: 404

Meaning: Not found

Description: The specified REST endpoint alors not exist. The caller can try again.

6. MTTP code: 405

Meaning: Not allowed

Description: A request was made of a resource using a medical that it not supported by that resource ource. For example, using GET on a form which requires data to be presented via Post, or using PUT on a read-only resource.

7. MTTP code: 406

Meaning: Not acceptable

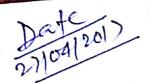
Description: when the planser sends a request, It will dispatch an Accept header. The blowser client will specify the characteristics of the data that it accepts. There are many stypes of accept headers including accept-charset, accept-encoding and accept-banges. The Accept header informs the server about the format in which the blowser wants to accept the data. The server will return the HTTP ever 406 not acceptable if it is unable to send data in the requested format.

D. HTTP code 1.500

Meaning: bystem Errors

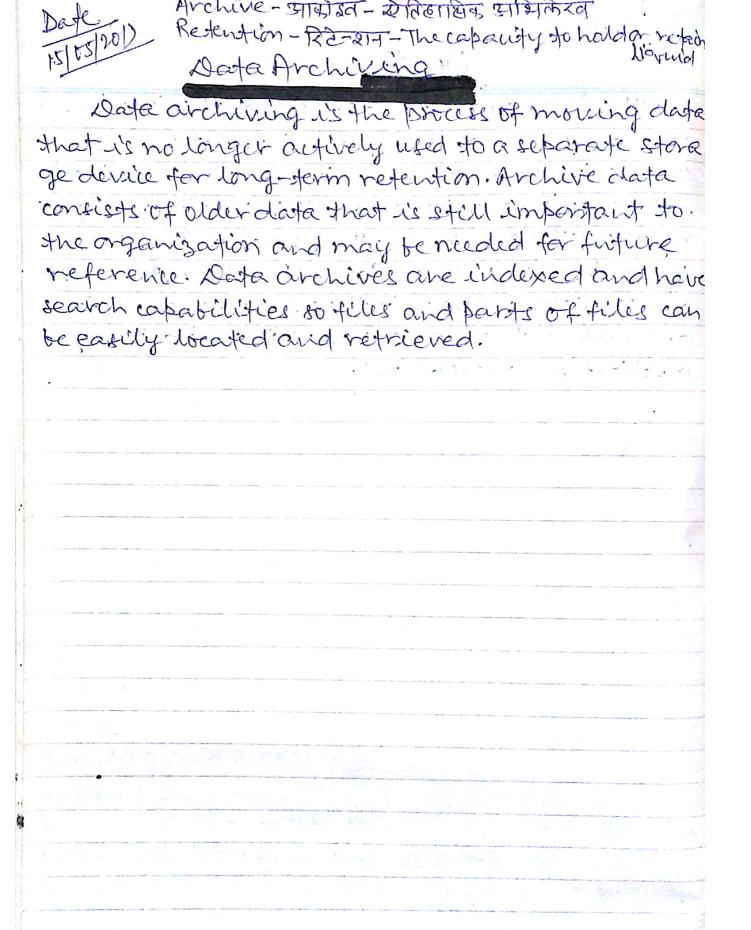
Description: If service implementation throws any other exception like network errors, database communication, framework returns HTTP 500:

Scanned by CamScanner



OAuth

commonly used as a way for internet users to authorize websites or applications to access their information on other websites but without gering them the passwords. This mechanism is used by companies such as Gazle, Facebook, Microsoft and Twitter to permit the users to share information about their accounts with third party applications or websites.



Reference: https://scotch.ro/bar\_talk/s-o-L-i-d-the-first-five-principles-of-object-origint T05/2019 S.O.L.I.D.: The First 5 Principles of Object Oriented mogramming & Design SOLLD are five basic principles which help to create good software architecture. SOLID is an acronym where: रेक्रानिम S -> stands for SRP(Single responsibility principle) 0 - stands for OCP (Open/Clused principle) L + stands for LSP (Liskov substitution. principle') किस्माम सक्रिट्राम I - stands for ISP (Interface segregation prin ciple) सेग्रग्रीशन D - stands for DIP (Dependency inversion princ iple) उ-गर्भन 1. Single responsibility principle: A class should have only one reason to change, meaning that a class should have only one job. For Ex; we have some shapes and we wanted to sum all the areas of the shapes. Well this is pretty simple right ? class circle public gradius; public function \_ construct (gradius) \$this - radius = stadius; class Square -119-

Dation Reference Blog agarque	my bless
25/05/2012	recipies
public slength;	
[sublic function_construct(\$10	
	of the grants
Athis - length = slength;	· na symbol and
	4
} - (vidinal of to polyment of the	47 Li Mints 4 0
First, we create our shapes classe	s and have the
constructors setup the required	
we move on by creating the Areac	
and then write up our logic to &	,
of all provided. Shapes.	104 15 1512 to 14
class Area Calculator	OHI
£	
protected schalpes;	15x 157 12 15 15 15
bublic function_construct (\$8h	The second secon
bublic function_construct (48h	The second secon
bublic function_construct (\$ sh {: Athis -> shapes = qshapes;	The second secon
and for a single of the section of	The second secon
and for a single of the section of	apes=array().
Sthis> shapes: = qshapes;  public function sum()	apes=array().
3+this -> shapes = qshapes;	apes=array().
public function sum()  { // logic to sum the areas	apes=array().
public function sum()  { // logic to sum the areas	apes=array().
Sthis -> shapes = & shapes;  public function sum()  { //logic to sum the areas }	apes=array().
Sthis -> shapes = qshapes;  public function sum()  { //logic to sum the areas  public function output()	apes=array().
public function sum()  { // logic to sum the areas  public function output()  {  return 'sum of the areas or	apes=array().
Sthis -> shapes = qshapes;  public function sum()  { //Logic to sum the areas  public function output()	apes=array().
public function sum()  { //Logic to sum the areas  public function output()  {  return 'sum of the areas or	apes=array().
Sthis -> shapes = qshapes;  public function sum()  { // logic sto sum the areas  public function output()  teturn 'sum of the areas of shapes: '. Ithis -> sum();  }	apes=array().
public function sum()  { //Logic to sum the areas  public function output()  {  return 'sum of the areas or	apes=array().