**Episode:38:Laravel Service Container:**

* It’s a container for services
* It is a place to store and retrieve services
* Services can be anything it may be a string ,object or number etc
* We store data in a container we use **add ,put, bind**
* If we use bind then associate with **(‘key ‘,’value’)**

**Code:**

**RouteFile**

Route::get('/', function () {

//need to instantiate the created container

$container = new \App\Container();

//if we need to store things in container we use the method add,put,bind

//when we call bind we need to give a key with associated data

$container->bind('example',function(){

//example is a key

return new\App\Example();

});

//resolve helper function is used to retrieve data from the container

$example =$container->resolve('example');

//ddd($example);

$example->go();

**Container.php**

* It’s a container for service
* It’s a place to store and retrieve a services
* Created a container named class

class container

{

//there is a method named bind()to store things in container

//representing an array

protected $bindings=[];

public function bind($key,$value)

{

$this->bindings[$key]=$value;

}

//using resolve() to fetch data

public function resolve($key)

{

if(isset($this->bindings[$key])){

//call\_user\_func is used to call the funcytion during runtime

return call\_user\_func($this->bindings[$key]);

}

}

}

**Example.php**

* Created a class named example which is also a keyword
* With a method go()

<?php

namespace App;

//example class

class Example

{

public function go()

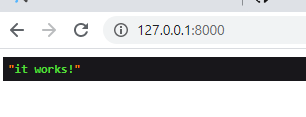
{

dump('it works!');

}

}

**Output:**

****