WORKING WITH RESPONSE FORMATE

Response Type String

```php
 10     function DemoAction(Request $request): array|string|null|int|bool
 11     {
 12         return  "This is string";
 13     }
```
DemoController.php

Response Type Int

```php
 10     function DemoAction(Request $request): array|string|null|int|bool
 11     {
 12         return  100;
 13     }
```
DemoController.php

WORKING WITH RESPONSE FORMATE

Response Type Null

```php
10    function DemoAction(Request $request): array|string|null|int|bool
11    {
12        return null;
13    }
```
DemoController.php

Response Type Boolean

```php
10    function DemoAction(Request $request): array|string|null|int|bool
11    {
12        return true;
13    }
```
DemoController.php

WORKING WITH RESPONSE FORMATE

Response Type Array

```
   DemoController.php

10     function DemoAction(Request $request): array|string|null|int|bool
11     {
12         return  array('A','B','C');
13     }
```

Response Type Associative Array

```
   DemoController.php

10     function DemoAction(Request $request): array|string|null|int|bool
11     {
12         return  array(['name'=>'Jhon','age'=>22]);
13     }
```

WORKING WITH RESPONSE FORMATE

■ Response Type JSON

```php
DemoController.php

10    function DemoAction(Request $request): array|string|null|int|bool
11    {
12        return response()->json(['name' => 'Abigail', 'state' => 'CA']);
13    }
```

■ Response with message,data & code

```php
DemoController.php

10    function DemoAction():JsonResponse{
11        $msg="Success";
12        $data=['name' => 'Abigail', 'state' => 'CA'];
13        $code=200;
14        return response()->json(['msg' => $msg, 'data' =>  $data],$code);
15    }
```

WORKING WITH RESPONSE FORMATE

Response Redirect

```php
DemoController.php

10    function DemoAction(Request $request): array|string|null|int|bool
11    {
12        return redirect('/DemoAction2');
13    }
```

WORKING WITH RESPONSE FORMATE

Binary File Response

```php
// DemoController.php

11    function DemoAction(): BinaryFileResponse
12    {
13        $pathToFile="img.jpg";
14        return response()->file($pathToFile);
15    }
```

File Download Response

```php
// DemoController.php

11    function DemoAction(): BinaryFileResponse
12    {
13        $pathToFile="img.jpg";
14        return response()->download($pathToFile);
15    }
```

WORKING WITH RESPONSE FORMATE

Generating Response Cookie

```php
DemoController.php

9       function DemoAction()
10      {
11          $name="cookie_1";
12          $value="This is my data";
13          $minutes=60;
14          $path="/";
15          $domain=$_SERVER['SERVER_NAME'];
16          $secure=true;
17          $httpOnly=true;
18          return response('Hello World')->cookie(
19              $name, $value, $minutes, $path, $domain, $secure, $httpOnly
20          );
21      }
```

WORKING WITH RESPONSE FORMATE

■ Attaching Headers To Responses

```php
DemoController.php

 9    function DemoAction()
10    {
11        return response("Hello ")
12            ->header('X-Header-One', 'Header Value')
13            ->header('X-Header-Two', 'Header Value');
14    }
```

■ Response View

```php
DemoController.php

 9    function DemoAction()
10    {
11        return view('welcome');
12    }
```

LARAVEL LOG

You may write information to the logs using emergency, alert, critical, error, warning, notice, info and debug

```php
        function DemoAction(Request $request):int
        {
                $sum=$request->num1+$request->num2;
                Log::info($sum);
                Log::emergency($sum);
                Log::alert($sum);
                Log::critical($sum);
                Log::error($sum);
                Log::warning($sum);
                Log::notice($sum);
                Log::debug($sum);
                return $sum;
        }
```
DemoController.php (lines 7–19)

```
1  [2023-04-26 13:50:30] local.INFO: 300
2  [2023-04-26 13:50:30] local.EMERGENCY: 300
3  [2023-04-26 13:50:30] local.ALERT: 300
4  [2023-04-26 13:50:30] local.CRITICAL: 300
5  [2023-04-26 13:50:30] local.ERROR: 300
6  [2023-04-26 13:50:30] local.WARNING: 300
7  [2023-04-26 13:50:30] local.NOTICE: 300
8  [2023-04-26 13:50:30] local.DEBUG: 300
```
laravel.log

LARAVEL LOG

To help you learn more about what's happening within your application, Laravel provides robust logging services that allow you to log messages to files

```php
DemoController.php

7        function DemoAction(Request $request):int
8        {
9             $sum=$request->num1+$request->num2;
10            Log::info($sum);
11            return $sum;
12        }
```

```
laravel.log

1   [2023-04-26 13:46:32] local.INFO: 5
2   [2023-04-26 13:46:35] local.INFO: 5
3   [2023-04-26 13:46:46] local.INFO: 105
4
```

LARAVEL SESSIONS

Sessions provide a way to store information about the user across multiple requests.

```
Route::get('/StoringData/{email}', [DemoController::class,'StoringData']);
Route::get('/RetrievingData', [DemoController::class,'RetrievingData']);
Route::get('/DeletingData', [DemoController::class,'DeletingData']);
Route::get('/FlashData', [DemoController::class,'FlashData']);
```

```
function StoringData(Request $request):bool
{
    $request->session()->put('user_email', $request->email);
    return true;
}
function RetrievingData(Request $request):string
{
    return $request->session()->pull('user_email', 'default');
}
function DeletingData(Request $request):bool
{
    $request->session()->forget('user_email');
    return true;
}
function FlashData(Request $request):bool
{
    $request->session()->flush();
    return true;
}
```