WORKING WITH SIMPLE REQUEST

```php
DemoController.php

7    class DemoController extends Controller
8    {
9        function DemoAction(): string
10       {
11           return "This is my first controller action";
12       }
13   }
```

```php
web.php

3    use App\Http\Controllers\DemoController;
4    use Illuminate\Support\Facades\Route;
5
6    Route::get('/DemoAction', [DemoController::class, 'DemoAction']);
```

Learn with Rabbil

REQUEST URL PARAMETER

```php
DemoController.php

9      function DemoAction(Request $request): string
10     {
11         $name=$request->name;
12         $age=$request->age;
13         return "Name is= ${name}, age is= ${age}";
14     }
```

```php
web.php

3  use App\Http\Controllers\DemoController;
4  use Illuminate\Support\Facades\Route;
5  Route::get('/DemoAction/{name}/{age}', [DemoController::class, 'DemoAction']);
```

REQUEST JSON BODY

```php
class DemoController extends Controller
{
    function DemoAction(Request $request): string
    {
        $name=$request->input('name');
        $age=$request->input('age');
        return "Name is= ${name}, age is= ${age}";
    }
}
```

DemoController.php

```php
<?php
use App\Http\Controllers\DemoController;
use Illuminate\Support\Facades\Route;
Route::post('/DemoAction', [DemoController::class, 'DemoAction']);
```

web.php

REQUEST HEADER

```php
DemoController.php

9       function DemoAction(Request $request): string
10      {
11          $name=$request->header('name');
12          $age=$request->header('age');
13          return "Name is= ${name}, age is= ${age}";
14      }
```

```php
web.php

1   <?php
2   use App\Http\Controllers\DemoController;
3   use Illuminate\Support\Facades\Route;
4   Route::post('/DemoAction', [DemoController::class, 'DemoAction']);
5
```

EXTRACT ALL REQUEST

HEADER BODY & QUERY PROPERTIES

DemoController.php

```php
9       function DemoAction(Request $request): array
10      {
11          return $request->header();
12      }
```

DemoController.php

```php
10      function DemoAction(Request $request): array
11      {
12          return $request->query();
13      }
```

DemoController.php

```php
10      function DemoAction(Request $request): array
11      {
12          return $request->input();
13      }
```

REQUEST MULTIPART/FORM-DATA

```php
DemoController.php

10      function DemoAction(Request $request):array
11      {
12          $img=$request->file('image_file');
13          $filesize=filesize($img);
14          $filetype=filetype($img);
15          $OriginalName=$img->getClientOriginalName();
16          $TempFilename=$img->getFilename();
17          $extension=$img->extension();
18
19          return array(
20              'filesize'=>$filesize,
21              'filetype'=>$filetype,
22              'OriginalName'=>$OriginalName,
23              'TempFilename'=>$TempFilename,
24              'extension'=> $extension
25          );
26
27      }
```

UPLOAD REQUEST FILE MULTIPART/FORM-DATA

■ Store File in Storage Folder

```php
$img=$request->file('image_file');
$img->storeAs('uploads', $img->getClientOriginalName());
```

■ Store File in Public Folder

```php
$img=$request->file('image_file');
$img->move(public_path('uploads'), $img->getClientOriginalName());
```

Learn with Rabbil

REQUEST IP AND CONTENT NEGOTIATION

Content Negotiation

```php
DemoController.php

10    function DemoAction(Request $request): array
11    {
12        return $request->getAcceptableContentTypes();
13    }
```

```php
DemoController.php

10    function DemoAction(Request $request): int
11    {
12        if ($request->accepts(['application/json'])) {
13            return 1;
14        }
15        else{
16            return 0;
17        }
18    }
```

IP Address

```php
DemoController.php

10    function DemoAction(Request $request): string
11    {
12        return $request->ip();
13    }
```

REQUEST COOKIES

Retrieving Cookies From Requests

```php
DemoController.php

10    function DemoAction(Request $request): array|string|null
11    {
12        return  $request->cookie();
13    }
```

```php
DemoController.php

10    function DemoAction(Request $request): array|string|null
11    {
12        return  $request->cookie('Cookie_2');
13    }
```

WORKING WITH RESPONSE FORMATE

■ Response Type String

```php
   DemoController.php

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

■ Response Type Int

```php
   DemoController.php

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

Learn with Rabbil

WORKING WITH RESPONSE FORMATE

Response Type Null

```php
●  ●  ●        DemoController.php

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

Response Type Boolean

```php
●  ●  ●        DemoController.php

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

Learn with
Rabbil

WORKING WITH RESPONSE FORMATE

Response Type Array

```php
DemoController.php

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

Response Type Associative Array

```php
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    }
```

Learn with Rabbil

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

Learn with Rabbil

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

Learn with Rabbil

■ 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

```
● ● ●    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

```
● ● ●    DemoController.php

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

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 LOG

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

DemoController.php

```php
7       function DemoAction(Request $request):int
8       {
9            $sum=$request->num1+$request->num2;
10           Log::info($sum);
11           Log::emergency($sum);
12           Log::alert($sum);
13           Log::critical($sum);
14           Log::error($sum);
15           Log::warning($sum);
16           Log::notice($sum);
17           Log::debug($sum);
18           return $sum;
19       }
```

laravel.log

```
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 SESSIONS

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

```php
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']);
```

```php
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;
}
```