

# By OnlineInterviewQuestions.com

## Laravel 5 interview questions

Laravel is one of the most popular frameworks on Github. On Github, Laravel has 54,843+ stars which is growing day by day. Here you can 100+ Laravel interview questions that are taken from various open-source places like quora, Github, and StackOverflow

### **Q1. List Server Requirements for installing Laravel Framework**

There is quite a requisite for the **installation of Laravel 6**. And, below is the list of the required servers that are needed for the installment of it.

- PHP 7.2.0 version
- Extension of BCMath PHP
- JSON PHP Extension
- XML PHP Extension
- Extension of CType PHP
- Extension of Tokenizer PHP
- Mbstring PHP Extension
- PDO PHP Extension
- OpenSSL PHP

### **Q2. What is Composer? How to install composer**

**Composer** is a tool to manage the **dependency in PHP**. Composer allows declaring the libraries of a project and the composer will install or update them for the user. It is not a package manager but deals with the libraries or packages and install them inside the project directory. The composer is downloaded by the installer in the windows and sets up a path for the environment that can be called from any directory.

The composer can be installed using a command

```
$curl -sS https://getcomposer.org/installer |  
sudo php -- --install-dir=/usr/local/bin --filename=composer
```

### **Q3. How to create a Laravel Project using composer.**

To create a Laravel project using composer you can follow the given steps:

- **Step 1:** Point your browser address bar to <https://getcomposer.org/composer.phar>
- **Step 2:** Create a new Laravel project. While in Terminal, browse where you want to set up your app and run the below command: `composer create-project laravel/laravel demowebsite`.
- **Step 3:** Configure virtual host and system host.

#### Q4. What is Laravel Homestead ?

**Laravel Homestead** helps in providing you a development environment and also that too without the installation of the PHP, HHVM, web server or any such thing in your local machine. It is completely disposable. This gives you the power to destroy if anything goes wrong and then you can just reuse it or recreate the same thing in just a few minutes.

#### Q5. How to start laravel Local Development Server.

We can run the Laravel Development server by typing **PHP artisan serve command** on terminal or command line in Larvel and always make sure it is present in your Laravel's project root directory else change your present working directory.

#### Q6. Explain What is Valet?

**Valet** can be defined as a tool that mainly helps to spin up a demo or those projects that are considered to be discardable as a breeze. The main function of valet is to combine various default software on the operating system with some creative and innovative ideas that are related to file paths and serving of PHP applications. In addition to that, they combine with some extra tools used for eliminating the requirement of modifying the `/etc/hosts` file.

#### Q7. List the steps to Configure Laravel application.

The steps to Configure Laravel application are:

- First, download the Laravel installer using Composer: ***composer global require laravel/installer***
- Create a new directory somewhere on your computer for the Laravel project that you are going to create and type the command: ***composer create-project laravel/laravel --prefer-dist***
- Install the complete Laravel framework: ***composer create-project laravel/laravel test dev-develop***
- Enable the installation of Laravel in the current directory: ***php artisan serve***

**Q8. In order to run Laravel. What permissions are required to be on storage and the bootstrap/cache directories ?**

In order to **run Laravel**, the directories within the storage and the bootstrap/cache directories should be writable by your web server otherwise the Laravel will not run.

**Q9. How to generate Application Key in laravel ? why it is necessary ?**

You can use **PHP artisan key: generate** to generate Application Key in laravel.

**Q10. How to create SEO-friendly URLs in Laravel?**

You can create SEO-friendly URLs in Laravel by adding the following features:

- Website speed must be fast
- Responsive designs to have a significant market share of internet usage
- Keywords must be appropriate
- The URLs should be separated by dashes instead of underscores.

**Q11. In which directory laravel configuration files are kept.**

All of the directory laravel configuration files are kept in *config directory*.

**Q12. How to do Environment Configuration in Laravel.**

Environment Configuration in Laravel can be done by following steps:

- Environment Configuration
- Accessing Configuration Values
- Caching of Configuration
- Maintenance Mode

**Q13. How can you retrieve values for laravel configuration files.**

You can retrieve values for Laravel configuration files by using **env helper** to retrieve values from these variables in your configuration files.

#### Q14. [How to retrieve/get current environment in Laravel.](#)

The entire variable listed in a **(.env)** the file will be loaded in the **\$\_ENV** PHP when a request is received by an application. However, an ENV helper may be implemented to retrieve the values of the variables in a particular configuration file. If a Laravel configuration file is reviewed, several options can be noticed to assist these helpers in the following way:

```
'Debug'=> env ('APP_DEBUG', false),
```

The second value that is passed onto the env function is the default value. This value can be used if there is no environment variable present in the given key.

#### Q15. [How to get/set configuration values in laravel?](#)

To **get/set configuration** values in Laravel, you can simply use **config helper** of Laravel or config facade. You can use the following syntaxes for using get and set configuration values.

- **Get Config Variable Value:** `$url = config('app.url');` OR `$url = Config::get('app.url');`
- **Set Config Variable Value:** `$url = config('app.url','http://interviewmocks.com/');` OR `$url = Config::set('app.url','http://interviewmocks.com/');`

#### Q16. [How to enable or disable maintain mode in Laravel.](#)

The **maintenance mode in a Laravel** can be enabled or disabled by artisan CLI.

Laravel returns a custom view that can be displayed in all requests when the website is in a maintenance mode. Thus, it becomes easier to enable or disable an application or a website while it is still under maintenance or updating.

To enable the maintenance mode in Laravel, the following artisan command should be entered into the command line:

```
php artisan down
```

To disable the maintenance mode in Laravel, the following artisan command should be executed in the command line:

```
php artisan up
```

#### Q17. [Explain the directory structure of laravel 5.6.](#)

The **storage directory** contains files such as, blade templates, sessions, caches, and log files plus app storage

like uploads, etc which are required frequently when a Laravel project is running.

### **Q18. List out the platforms Laravel Homestead supports ?**

**Laravel Homestead** is a pre-packaged and official Vagrant "box", that offers a great improvement environment without needing one to install **HHVM, PHP, web servers** or any distinct server software on your local machine.

It functions on every **Mac, Linux system, Windows, Nginx web servers, PHP 7.2, PHP 7.1, PHP 7.3, PostgreSQL, Redis, MySQL, Node, Memcached** and every other article one requires to improve the outstanding Laravel applications.

### **Q19. List the Software's included in Laravel Homestead ?**

There are numerous Softwares included in Laravel Homestead. Some of these are:

- Ubuntu 14.04.
- PHP 5.6.
- HHVM.
- Nginx.
- MySQL.
- Postgres.
- Node (With Bower, Grunt, and Gulp)
- Redis.

### **Q20. How to install Homestead Vagrant Box ?**

To install Homestead Vagrant Box, you have to unfurl git bash and execute the command "vagrant box add laravel/homestead"

### **Q21. Please provide steps to Configure Homestead ?**

The steps to Configure Homestead are as follows:

- Hardware Virtualization
- Downloading VirtualBox and Vagrant
- Install Git Bash
- Installing Homestead Box
- Cloning Homestead
- Creating ssh keys
- Editing Configuration File

- Editing the Host file
- Starting Homestead
- Creating Laravel project
- Running our Application

## Q22. How to launch Vagrant Box ?

**Vagrant** is a software that is utilized to handle the improvement environment. Via the command line, one can take any available OS, configure it, install it, work within it, run it, and additional functions. Utilizing it one can restore the manufacturing environment of one's website or app.

**To launch Vagrant Box, one has to:**

- Start and initialize the Vagrant Box.
- Select a box then arrange the Vagrant Box.
- Incorporate SSH within the box and Customize It.
- Create the box as tiny as possible.
- Incorporate the box within one's Vagrant install.
- Customize the new Vagrantfile.

## Q23. What is Vagrantfile and Homestead.yaml ,

The **Vagrantfile** is a Ruby file utilized to arrange Vagrant on a per-project way. The chief function of the Vagrantfile is to explain the virtual devices needed for projects with the process of configuration and amenities of the devices.

**Laravel Homestead** happens to be a pre-packaged Vagrant Box that offers one a marvelous development environment san needing one to install, web servers, PHP, or any distinct server software on one's local machine. These Vagrant Boxes are fully disposable.

## Q24. How do you upgrade Homestead?

You can upgrade Homestead with the help of the following steps:

- Destroy your box and remove the old version.
- Update VirtualBox to 5.1.10 with the extensions if needed.
- Update Vagrant to 1.9.1.
- Rename your Homestead directory to Homestead-Old.
- Run `vagrant box add Lravel/homestead`.
- Run `vagrant up` and that should be it.

## Q25. [How to install Installing MariaDB on Homestead ?](#)

You can **Install MariaDB on Homestead** by following the given step:

- Remove MySQL server.
- `sudo apt-get remove --purge MySQL-server MySQL-client MySQL-common`
- `sudo apt-get autoremove`
- `sudo apt-get autoclean`
- After that install MariaDB
- `sudo apt-get install software-properties-common`
- `sudo apt-key adv --recv-keys --keyserver hkp://keyserver.ubuntu.com:80 0xc9cb082a1bb943db`
- `sudo add-apt-repository 'deb [arch=amd64,i386] http://sfo1.mirrors.digitalocean.com/mariadb/repo/10.1/ubuntu trusty main'`
- `sudo apt-get update`
- `sudo apt-get install mariadb-server`
- Now, In your homestead.yaml file, use
- MariaDB: - homestead

## Q26. [Command to connect vagrant via ssh ?](#)

You can add ssh config for your vagrant host to ssh config by the following method:

- Get ssh config for the vagrant machine in the vagrant folder: `vagrant ssh-config`.
- Open `{UserDir}/.ssh/config` and append their result from the previous command.
- Ssh to vagrant: `ssh vagrant`.

## Q27. [How to add Additional Sites in Homestead environment?](#)

You can add an additional site by simply add the site to your `~/homestead/Homestead.YAML` file and then run the `vagrant reload --provision` terminal command from your Homestead directory.

## Q28. [List Site Types supported by Homestead?](#)

**Homestead** assists numerous kinds of sites that permits one to easily execute projects that are not founded on Laravel. The sites supported by Homestead include:

- Symfony4.
- Apache.
- Laravel.

- Proxy.
- Symfony2.
- Statamic.
- SilverStripe.

## Q29. How to configure Cron jobs in Homestead ?

There are numerous ways to configure Cron in Homestead like:

- Make new Artisan command and register it within the Kernel. Then execute the “PHP artisan list” command within the terminal. One will observe the command has been recorded. Command will be implemented with the sign that is kept in “protected \$signature = ‘command: name’”. One will observe the log message within the terminal.
- Utilize call back or closure method: Laravel permits one to close, execute a call back utilizing the “call” process. Laravel also permits one to schedule shell commands to initiate commands to the external applications and operating system.
- Simply after reaching the terminal add SSH to the server and cd within the project and execute the command "cronstab -e". This will ensure its opening and after pasting the code below within the file, save it and exit.

## Q30. List the default ports that are forwarded to your Homestead environment?

Some of the default ports that are forwarded to your Homestead environment are:

- SSH: 2222 ? Forwards To 22.
- ngrok UI: 4040 ? Forwards To 4040.
- HTTP: 8000 ? Forwards To 80.
- HTTPS: 44300 ? Forwards To 443.
- MySQL: 33060 ? Forwards To 3306.
- PostgreSQL: 54320 ? Forwards To 5432.
- MongoDB: 27017 ? Forwards To 27017.

## Q31. How to share your Homestead with others.

You can share your Homestead with others by using Vagrant Cloud by using the **vagrant share command**. It works with single-site setups only.



### Q32. Is Multiple PHP Versions are supported by Homestead ?

Yes, Homestead v6 or above versions support the multiple PHP Versions such that you can automatically get multiple PHP versions installed by default. The latest Laravel-Version with Homestead including PHP 5.6 is Laravel 5.6. These Homestead-Version supports PHP 5.6 to 7.3.

### Q33. How to enable Multiple PHP Versions in Homestead ?

Multiple versions of PHP are supported in Homestead 6 on the same machine. In the **Homestead.yaml** file, the version of PHP needs to be specified before using it. The available versions of PHP are 7.1, 7.2 and 7.3.

Any of the Supported versions of PHP can be used through the CLI in the following way.

- Php7.1 artisan list
- Php7.2 artisan list
- Php7.3 artisan list

### Q34. How to configure network interfaces in Homestead ?

To configure network interfaces in Homestead, you can follow the given steps:

- Run the gist above to prove everything works fine.
- Create a .bashrc file like
- In your after.sh, source the .bashrc:

```
cp /home/vagrant/Code/env/homestead/.bashrc ~/.bashrc
```

```
dos2unix ~/.bashrc
```

```
source ~/.bashrc
```

- Now running vagrant halt and then vagrant reload --provision

### Q35. How can you update Homestead ?

You update Homestead by following the given steps:

- Destroy your box and remove old version.
- Update VirtualBox to 5.1.10 with the extensions if needed.
- Update Vagrant to 1.9.1.
- Rename your Homestead directory to Homestead-Old.

- Run vagrant box add laravel/homestead.
- Run vagrant up and that should be it.

### Q36. List the applications supported by Valet?

There are several applications supported by Valet. Some of these are:

- Laravel
- Lumen
- Bedrock
- CakePHP 3
- Concrete5
- Contao
- Craft
- Drupal

### Q37. Is Valet supports windows or Ubuntu environment?

Basically, Valet is a Laravel development environment for Mac minimalists that means it is a **Mac-only application**. Therefore you can say Valet does not supports windows or Ubuntu environment.

### Q38. List the requirements to install Valet ?

**Before installing Valet fulfill these below requirements:**

- **Composer:** One must install Composer before installing Valet.
- **PHP:** Before installing Valet install PHP if it is not installed.
- Carefully check whether one's system is not executing or running any programs or applications that bind port 80 similar to Nginx or Apache.

Now install Valet internationally with `composer global require cretueusebiu/valet-windows`.

Now configure Valet along with registration of Valet's Daemon. This will launch Valet in an automated way on system setup.

### Q39. What is Homebrew ?

**Homebrew** is an open-source software package management system for macOS and Linux. It helps Linux and macOS users for the installation of software in a simple manner. Homebrew is written in Ruby programming language and free to download.

#### Q40. [How to install Valet ?](#)

You can install Valet by taking the given steps:

**Step 1.** Install Homebrew

**Step 2.** Install PHP

**Step 3.** Install MySQL

**Step 4.** Install Composer

**Step 5.** Install Valet: composer global require laravel/valet

**Step 6.** Serving Sites

#### Q41. [How to uninstall Valet?](#)

You can uninstall the valet by removing the. vendor directory and the valet binary found in */usr/local/bin/valet*. It will permanently remove Valet from your Mac.

#### Q42. [How to configure Vault server to use another domain ?](#)

There are different ways to configure a Vault server depending on the server. These are divided into the following categories:

- **Command line flags** are passed to the vault binary as part of the complete command line at runtime.
- **Configuration files** are read by the vault process at runtime and when reloaded by sending the process a hangup signal (SIGHUP).
- **Environment variables** are set in the environment for the shell of the user that executes the vault process and can only configure a limited set of options.

#### Q43. [How do I start valet in Laravel?](#)

You can start Valet by following the given step:

- Install PHP using Homebrew
- Install the **laravel/valet** Composer package
- **Run** the **valet** install command.

#### Q44. [How to upgrade Valet To Valet 2.0 ?](#)

Upgrading valet to v2 is simple and can be completed in just a few steps. First create a backup of your ~/.valet folder. Then uninstall your original:

- **valet stop**
- **valet uninstall**

Now, upgrade to the latest version of Valet. If you installed Valet via Composer, you should use the following command to update to the latest major version:

- *composer global require laravel/valet*

After the fresh Valet source code has been downloaded, you should run the install command:

- *valet install*
- *valet restart*

#### Q45. [How to serve sites on Valet ?](#)

To **serve sites on Valet**, you will need to first configure a folder where Valet should search for your sites and the park command will register your current directory as a path that Valet should search for your sites.

#### Q46. [What does valet park and link command do?](#)

In Valet, **park and link** are the two commands to help you serve your Laravel sites. park command lets you create a new directory on your Mac by running something like `mkdir ~/Sites`. Whereas The link command may also be used to serve your Laravel sites.

#### Q47. [How to Serve sites with TLS on Valet ?](#)

**Securing sites with TLS** is a function of valet. Basically, Valet serves sites over HTTP. However, if you would like to serve a site over encrypted TLS using HTTP/2, you may use the secure command.

#### **Q48. How to Share Sites with external world on valet ?**

**Valet** includes a command to share your local sites with the world. You can share it by :

- Sharing Sites Via Ngrok
- Sharing Sites Via Expose
- Sharing Sites Via Expose

#### **Q49. List some uses of Valet?**

**Valet** is a Laravel improvement environment for Mac essentialist. One can also share one's sites publicly utilizing local tunnels. It configures one's Mac to continue running Nginx in the framework when one's machine starts. Then utilizing DnsMasq Valet substitutes every appeal on the \*.test domain to depict or point to websites installed in one's local machine.

**Valet supports the following frameworks and cms:**

- Static HTML.
- Zend.
- WordPress.
- Laravel.
- Craft.
- Magento.
- Kirby.
- Slim.
- Katana.
- Bedrock.
- Contao.
- Statamic.
- Symfony.
- Concrete5.
- Jigsaw.
- Joomla.
- Lumen.
- OctoberCMS.

#### **Q50. Explain Request life cycle of laravel.**

Laravel Request Lifecycle is as follows:

- Auto Loader
- Kernel
- Service Providers
- Dispatch Request
- Router

### Q51. What is name of first file that loaded in Laravel?

The **php** file or the *bootstrap/start.php* file is the first file that is loaded in Laravel. It creates the new Laravel Application object, which also serves as an IoC container.

### Q52. What is HTTP / Console Kernels in laravel?

The **HTTP Kernel** is used to process requests that come in through the web (HTTP) whereas the **Console Kernel** is used when you interact with your application from the command line.

### Q53. Explain Service Container ?

A **Service Container** is also known as a dependency injection container which is simply a PHP object that manages the instantiation of services. It is a powerful tool for managing class dependencies and performing dependency injection.

### Q54. How to bind a service container to a service provide.

To bind a service container to a service provide use *App\Services\Transistor; use App\Services\PodcastParser;*  
*\$this->app->bind(Transistor::class, function (\$app) { return new Transistor(\$app->make(PodcastParser::class)); });*

### Q55. What is Contextual Binding ?

Sometimes one can have 2 classes that use a similar interface, but one wishes to administer various implementation within all classes. For instance, two controllers can depend on the various implementation of the **Illuminate\Contracts\Filesystem\Filesystem** Contract.

Laravel offers a simple and fluent assemblage for explaining this behavior. Thus, a contextual binding permits one to bind an execution to a selected class.

### Q56. How to resolve class instance out of the container in Laravel ?

To resolve **class instance** out of the container in Laravel, for readability Laravel also offers an alias to the `App::make` method called `makeWith` which simply calls the `make` method behind the scenes. Know you can start resolving classes from the Container and have all their dependencies resolved automatically for you.

### Q57. Explain Automatic Injection in Laravel ?

In **automatic injection** in laravel, the class dependencies are "injected" into the class via the constructor or, in some cases, "setter" methods.

### Q58. How can you write your own service provider in laravel.

You can write your own **service provider** in laravel with the help of service providers. In `config/app.php` file included with Laravel, you will see a provider array, these are all of the service provider classes that will be loaded for your application.

### Q59. What service provider's Register Method does ?

The **Service Provider Register Method** is used to declare all your service container bindings, with the help of the `boot` method, you can use already registered services via the `register` method.

### Q60. How to register a service provider in laravel ?

To register your service provider in laravel, you just need to take a simple step i.e. add an entry to the array of service providers in the **`config/app.php` file**.

### Q61. What are Deferred Providers in laravel ?

**Deferred Providers in Laravel** are every service provider class that will be packed for your implementation. Although more of these are "deferred" providers, explaining that they will be unloaded on all requests, only during the facilities their activities are required. Hence, if one's provider is only recording bindings in the Service Container, one can wish to defer its record until one of the recorded bindings is truly required. This will enhance the performance of the applications as it is not filled from the filesystem on all requests. To defer the filling of a provider, execute the `\Illuminate\Contracts\Support\DeferrableProvider` interface and explain a `Provides` procedure. The `Provides` process should send back the facility container bindings recorded by the provider.

## Q62. [How to register a service provider via composer in Laravel ?](#)

To register a service provider via composer in Laravel using the register method.

## Q63. [What are Laravel Facades?](#)

**Laravel Facades** act as "static proxies" to the classes that already exist in the classes in the service container, it provides the benefit of an expressive syntax while maintaining more testability and the flexibility that is different from the traditional static methods. Laravel provides many Facades and those Facades provide access to almost all the features of Laravel. All the Facades of Laravel are defined inside the namespace `Illuminate\Support\Facades`.

## Q64. [How Facades are different from Dependency injection ?](#)

**Facades** use dynamic methods to proxy method calls to objects resolved from the service container, and we can test facades just as we would test an injected class instance. Whereas the **dependency injection** is the ability to swap implementations of the injected class.

## Q65. [What getFacadeAccessor method does?](#)

**getFacadeAccessor** method is used to define what to resolve from the container in facade class.

## Q66. [Explain Laravel Contracts?](#)

**Laravel's Contracts** are basically a set of interfaces consists of various functionalities that define the core services provided by the framework.

## Q67. [Difference between Contracts and Facades?](#)

**Facades** use dynamic methods to proxy method calls to objects resolved from the service container, and we can test facades just as we would test an injected class instance. Whereas Laravel's **contracts** are basically a set of interfaces consists of various functionalities that define the core services provided by the framework.

## Q68. [How to do 301 redirects in Laravel ?](#)



You can use `\Redirect::to('/new-url', 301);` for 301 redirects.

**example :**

```
Route::get('old-url', function(){
    return Redirect::to('/bar', 301);
});
```

## **Q69. Explain Named Routes in Laravel ?**

Named routes allow the convenient generation of URLs or redirects for specific routes. You may define a name for a path by chaining the name process onto the path definition.

Example

```
Route::get('user/registration', function () {
    //
})->name('registration');
```

## **Q70. How to generate Urls from named named Routes?**

To generate URLs from named Routes, follow the given steps:

- **//Generating URLs.**
- **\$url= route('student\_details');**
- **//Generating Redirects...**
- **return redirect() -> route('student\_details');**

## **Q71. How to do subdomain routing in Laravel ?**

You can do subdomain routing in Laravel by using routes groups to achieve this:

```
Route::group(array('domain' => '{subdomain}. website.com'), function () { Route::get('/', function ($subdomain) { $name = DB::table('users')->where('name', $subdomain)->get(); dd($name); }); });
```

## **Q72. What are Route Prefixes ?**

Route prefixes are associated with routes by design in attribute routing. It is basically used to set a common prefix for an entire controller. It can be done by: ***Route::namespace('Admin') ->prefix('admin') ->name('admin. ') ->group(function () { Route::resource('users', 'UsersController'); })***

### Q73. [How to return a view from route in Laravel ?](#)

To return a view from route in Laravel you can use: *route. php* `Route::get('/page', function(){ return View::make('dir. page'); });`; *controller. php* `View::make('/page');`

### Q74. [How to configure route cache and clear route cache in Laravel ?](#)

To configure route cache and clear route cache in Laravel, you can use the command: **\$ php artisan route:clear** **Route cache cleared!**. If you want to cache your routes again, simply run the following command: **\$ php artisan route:cache** **Route cache cleared!**.

### Q75. [How to list all routes in Laravel ?](#)

To list all registered routes in Laravel application you can use `php artisan route:list` command or also use `Route::getRoutes()` ; method it will return a `RouteCollection`.

### Q76. [What Route Model Binding ?](#)

**Route Model Binding** provides a mechanism to bind or inject a model instance in the route, and to validates, a model Id injected in the route by injecting a Model instance. In this, the model instance is directly injected bypassing the type hinted variable as a parameter in the closure.

### Q77. [What is difference between Implicit and Explicit route Binding in Laravel ?](#)

In **Implicit Binding**, model instances are directly injected into the route or controller actions whose type-hinted variable names match a route segment name. Whereas in **Explicit Binding** the router's model method specifies the class for a given parameter.

### Q78. [How to get current Url in Laravel ?](#)

You can use `url()->current()` ; method to get the current URL in Laravel Framework. This method returns only the URL without its parameter. If complete URL with parameter is required then use `url()->full()` ;

### Q79. [What are middlewares in Laravel ?](#)

**Middleware** in a Laravel is the bridge between the response and a request. Middleware is a type of mechanism for filtering. Middleware in Laravel verifies the authentication of the user. If the user is genuine, it redirects to the home page. If not, then the user is redirected to the login page. Each and Every middleware should be registered before using it. Parameters can also be passed through the middleware's. The middleware is terminated after the response is sent to the browser.

There are two types of Middleware in Laravel. They are:

- **Global Middle:** It executes every Http request of an application
- **Route Middleware:** It is assigned to a particular route.

### Q80. [How to create custom Middleware in Laravel ?](#)

To create custom Middleware in Laravel you can follow the given steps:

- Create Custom Middleware.
- Add Route.
- Add Method to Controller.

### Q81. [How to register a middleware in Laravel ?](#)

In Laravel, the **middleware** can be registered at app/Http/Kernel which contains two properties \$middleware that is used to register Global Middleware and \$routeMiddleware property that is used to register route specific middleware.

### Q82. [Can we assign multiple middlewares to a route.](#)

Yes, we can assign multiple middlewares to a route.

### Q83. [How to add a middleware in route group?](#)

You can add a middleware in the route group by registering your middleware in the **\$routeMiddleware** instead of the \$middleware array.

### Q84. [What are Terminable Middlewares ?](#)

**Terminable middleware** is implemented after the request has been run and before the response is ready. The response and request object inside a terminate method of the Terminable middleware is received, but the response cannot be changed it has been returned already. When the response is on the way to a browser, terminable middleware allows executing a middleware that is processed.

If a terminate method is defined in middleware and a web server is implying the Fast CGI method, then the terminate method will be automatically executed after the response is sent to a browser.

### **Q85. How to add CSRF Protection in Laravel ?**

**CSRF token** protection can be applied to any HTML form in the Laravel application by specifying a hidden form field of the CSRF token.

### **Q86. How can you Exclude URIs From CSRF Protection in Laravel ?**

You can **exclude URLs** from CSRF Protection in Laravel by defining it's routes outside of the web middleware group that is included in the default routes.php file or by adding the URIs to the \$except property of the VerifyCsrfToken middleware.

### **Q87. What is X-CSRF-TOKEN , How it is different from CSRF Tokens ?**

**X XSRF** token is added to the request header for ajax requests. It is a type of malicious exploit of a website where unauthorized commands are submitted from a user that the web application trusts. Whereas **CSRF** is a one-click attack or session riding.

### **Q88. What are Controllers in Laravel ?**

**Controller in Laravel** is a class in which is responsible for controlling application execution flow. It sits between Model and view. Most of the business logic are written in the controller. In Laravel controllers are kept in the app/Http/Controllers directory.

Example Controller In Laravel

```
<?php
namespace App\Http\Controllers;
use Illuminate\Support\Facades\Redirect;
use Illuminate\Http\Request;
class HomeController extends Controller
{
    /**
     * Create a new controller instance.
```

```

        *
        * @return void
        */
        public function __construct()
        {

        }

        public function index(Request $request){
        }
    }
}

```

### Q89. [How to create a controller in Laravel ?](#)

To create a controller in Laravel by extending the base controller class included with Laravel:  
**App\Http\Controllers\Controller.**

### Q90. [How to register a middlewares in Laravel controller ?](#)

In Laravel, the middleware can be registered at **app/Http/Kernel** which contains two properties **\$middleware** that is used to register Global Middleware and **\$routeMiddleware** property that is used to register route specific middleware.

### Q91. [How to generate a controller with resources in Laravel ?](#)

You can generate a controller with resources in Laravel by typing - PHP artisan makes: controller sharkController --resource. It will create a resource controller with all the methods in it.

### Q92. [How to Localize Resource URIs in Laravel ?](#)

To localize the create and edit action verbs, you may use the Route::resourceVerbs method. This may be done in the boot method of your AppServiceProvider:

### Q93. [How to do Dependency Injection in Laravel Controllers?](#)

You can do Dependency Injection in Laravel Controllers by following the given steps:

- public function \_\_construct(Request \$request)
- composer create-project laravel/laravel DI --prefer-dist.
- DB\_CONNECTION=mysql DB\_HOST=127.0.0.1 DB\_PORT=3306 DB\_DATABASE=di DB\_USERNAME=root DB\_PASSWORD=mysql.

- php artisan make:seeder UsersTableSeeder.

#### **Q94. How to get requested path in Laravel ?**

**Request::getPathInfo()** is used to get requested path in Laravel.

#### **Q95. How to check request is ajax or not ?**

In Laravel, we can use `$request->ajax()` method to check request is ajax or not.

**Example:**

```
public function saveData(Request $request)
{
    if($request->ajax()){
        return "Request is of Ajax Type";
    }
    return "Request is of Http type";
}
```

#### **Q96. How to retrieve all requested data in Laravel ?**

You can retrieve all requested data in Laravel by using `$request->all()`.

#### **Q97. How to check an Input value is present or not in Laravel ?**

You can check whether an Input value is present or not in Laravel by running the following code:

```
if (Input::has('name'))
{
    //
}
```

#### **Q98. What are Cookies ? How to get , set , destroy cookies in Laravel ?**

**Cookies** are text files with small pieces of data that are used to identify your computer when you visit Internet

sites. With the help of **cookies::make()** method to create or set cookies and With the help of **Cookie::forget()** method to delete or destroy cookies in Laravel.

### **Q99. How to upload a file in Laravel 5?**

To upload a file in Laravel 5 you can follow the given steps:

- Define Route
- Add Controller Function
- Add Blade File

### **Q100. What are Macros in Laravel?**

Macros provide a way to add functionality to the classes wherein you do not have that function. It can help you to create a small reusable component that'll be possible to use it everywhere on your application.

### **Q101. How to redirect user with Flashed Session Data in Laravel?**

You can redirect user with Flashed Session Data in Laravel by:

```
Route::post('/user/profile', function () {  
  
    // Update the user's profile  
  
    return redirect('/dashboard')->with('status', 'Profile updated!');  
  
});
```

### **Q102. How much laravel experience do you have?**

Tell the recruiter about the number of years of Laravel experience do you have. You can also tell the number of projects you have done in Laravel.

Please Visit [OnlineInterviewquestions.com](https://www.onlineinterviewquestions.com) to download more pdfs