**Laravel interview questions**

**1.What is Laravel?**

**Laravel** is a free open source "PHP framework" based on the MVC design pattern. Laravel is a [Symfony](https://www.onlineinterviewquestions.com/laravel-interview-questions/) based free open-source PHP web framework. It is created by Taylor Otwell and allows developers to write expressive, elegant syntax. Laravel comes with built-in support for user authentication and authorization which is missing in some most popular PHP frameworks like CodeIgniter, CakePHP.

|  | |
| --- | --- |
| **Laravel** is written In | PHP Programming (PHP 7) |
| **Laravel** is a | PHP Framework for Developing websites and mobile API's. |
| **Laravel** is developed By | Taylor Otwell |
| **Laravel** is Based on | MVC architectural pattern |
| **Laravel** Dependencies | Composer, OpenSSL, PDO, Mbstring, etc. |
| **Laravel** Licence | MIT License |
| **Laravel** Current Stable release | 6.1.4 |

**2. What are pros and cons of using Laravel Framework?**

**Pros of using Laravel Framework**

1. Laravel framework has in-built lightweight blade template engine to speed up compiling task and create layouts with dynamic content easily.
2. Hassles code reusability.
3. Eloquent ORM with PHP active record implementation
4. Built in command line tool “Artisan” for creating a code skeleton , database structure and build their migration
5. Supports MVC pattern of software development.
6. Modular structure.
7. Great Documentation and Number of forums.
8. Dependency can be installed using composer tool.
9. Easy to implement Authentication and Authorization.
10. Inbuilt CSRF protection features.
11. Powerful Artisan Command Line Interface.
12. Featured with blade template engine, templates can be organize.
13. Default Cache driver is configured and also support various cache driver like Redis and Memcached.

**Cons of using laravel Framework**

1. Development process requires you to work with standards and should have real understanding of programming
2. Laravel is new framework and composer is not so strong in compare to **npm (for node.js)**, **ruby gems** and**python pip**.
3. Development in laravel is not so fast in compare to ruby on rails.
4. Laravel is lightweight so it has less inbuilt support in compare to django and rails. But this problem can be solved by integrating third party tools, but for large and very custom websites it may be a tedious task

**3. Explain Events in laravel ?**

An event is an action or occurrence recognized by a program that may be handled by the program or code. Laravel events provides a simple observer implementation, that allowing you to subscribe and listen for various events/actions that occur in your application.

All **Event** classes are generally stored in the app/Events directory, while their listeners are stored in app/Listeners of your application.

**4. Explain validations in laravel?**

In Programming validations are a handy way to ensure that your data is always in a clean and expected format before it gets into your database. Laravel provides several different ways to validate your application incoming data.By default Laravel’s base controller class uses a **ValidatesRequests** **trait** which provides a convenient method to validate all incoming HTTP requests coming from client.You can also validate data in laravel by creating Form Request.

**5. How to install laravel via composer ?**

composer create-project laravel/laravel your-project-name version

**6. List some features of laravel 5.0 ?**

Inbuilt CRSF ([cross-site request forgery](https://en.wikipedia.org/wiki/Cross-site_request_forgery) ) Protection, Inbuilt paginations, [Reverse Routing](https://www.laravelinterviewquestions.com/2017/03/what-is-reverse-routing-in-laravel.html), Query builder, Route caching, Database Migration, IOC (Inverse of Control) Container Or service container.

**7. What is PHP artisan. List out some artisan commands ?**

**PHP artisan** is the command line interface/tool included with Laravel. It provides a number of helpful commands that can help you while you build your application easily. Here are the list of some artisan command:-

php artisan list : list all available artisan commands

php artisan help migrate : display help screen for migrate command

php artisan migrate –env=local : set the configuration environment as local

php artisan –version : display the current version of laravel installation

php artisan help, php artisan tinker, php artisan make, php artisan make model model\_name, php artisan make controller controller\_name

**8. List some default packages provided by Laravel 5.6 ?**

Cashier, Envoy, Passport, Scout, Socialite, Horizon

**9. What are named routes in Laravel?**

**Named routing** is another amazing feature of Laravel framework. Named routes allow referring to routes when generating redirects or Urls more comfortably.  
**You can specify named routes by chaining the name method onto the route definition:**

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

})->name('profile');

**You can specify route names for controller actions:**

Route::get('user/profile', 'UserController@showProfile')->name('profile');

**Once you have assigned a name to your routes, you may use the route's name when generating URLs or redirects via the global route function:**

// Generating URLs...

$url = route('profile');

// Generating Redirects...

return redirect()->route('profile');

**10. What is database migration. How to create migration via artisan ?**

**Migrations** are like version control for your database, that’s allow your team to easily modify and share the application’s database schema. Migrations are typically paired with Laravel’s schema builder to easily build your application’s database schema.

php artisan make:migration create\_users\_table // creating Migration

**11. What are service providers in Laravel ?**

**Service Providers** are central place where all laravel application is bootstrapped . Your application as well all Laravel core services are also bootstrapped by service providers.  
All service providers extend the Illuminate\Support\ServiceProvider class. Most service providers contain a register and a boot method. Within the register method, you should only bind things into the service container. You should never attempt to register any event listeners, routes,

**12. Explain Laravel’s service container ?**

One of the most powerful feature of Laravel is its **Service Container**. It is a powerful tool for resolving class dependencies and performing dependency injection in Laravel.  
**Dependency injection** is a fancy phrase that essentially means class dependencies are “injected” into the class via the constructor or, in some cases, “setter” methods.

**13. What is composer ?**

**Composer**is a tool for managing dependency in PHP. It allows you to declare the libraries on which your project depends on and will manage (install/update) them for you.  
**Laravel**utilizes Composer to manage its dependencies.

**14. What are Laravel Contract’s ?**

Laravel's Contracts are nothing but a set of interfaces that define the core services provided by the Laravel framework.

**15. Explain Facades in Laravel ?**

**Laravel Facades** provides a static like an interface to classes that are available in the application’s service container. Laravel self-ships with many facades which provide access to almost all features of Laravel ’s. Laravel facades serve as “static proxies” to underlying classes in the service container and provide benefits of a terse, expressive syntax while maintaining more testability and flexibility than traditional static methods of classes. All of Laravel’s facades are defined in the Illuminate\Support\Facades namespace. You can easily access a facade like so:

use Illuminate\Support\Facades\Cache;

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

return Cache::get('key');

});

**16. What are Laravel eloquent?**

Laravel's **Eloquent ORM** is simple Active Record implementation for working with your database. Laravel provide many different ways to interact with your database, Eloquent is most notable of them. Each database table has a corresponding “Model” which is used to interact with that table. Models allow you to query for data in your tables, as well as insert new records into the table.

Below is sample usage for querying and inserting new records in Database with Eloquent.

// Querying or finding records from products table where tag is 'new'

$products= Product::where('tag','new');

// Inserting new record

$product =new Product;

$product->title="Iphone 7";

$product->price="$700";

$product->tag='iphone';

$product->save();

**17. How to enable query log in Laravel ?**

DB::connection()->enableQueryLog();

You can get array of the executed queries by using getQueryLog method:

$queries = DB::getQueryLog();

**18. What is reverse routing in Laravel?**

**Laravel reverse routing** is generating URL's based on route declarations. Reverse routing makes your application so much more flexible. It defines a relationship between links and Laravel routes. When a link is created by using names of existing routes, appropriate Uri's are created automatically by Laravel. Here is an example of reverse routing.

Route::get('login', 'users@login'); //// route declaration

Using reverse routing we can create a link to it and pass in any parameters that we have defined. Optional parameters, if not supplied, are removed from the generated link.

{{ HTML::link\_to\_action('users@login') }}

It will automatically generate an Url like http://xyz.com/login in view.

**19. What are traits in Laravel?**

**PHP Traits** are simply a group of methods that you want include within another class. A Trait, like an abstract class cannot be instantiated by itself. Trait are created to reduce the limitations of single inheritance in PHP by enabling a developer to reuse sets of methods freely in several independent classes living in different class hierarchies.

trait Sharable {

public function share($item) {

return 'share this item'; }

}

**You could then include this Trait within other classes like this:**

class Post {

use Sharable;

}

class Comment {

use Sharable;

}

**Now if you were to create new objects out of these classes you would find that they both have the share() method available:**

$post = new Post;

echo $post->share(''); // 'share this item'

$comment = new Comment;

echo $comment->share(''); // 'share this item'

**20. Explain Laravel’s Middleware?**

As the name suggests, Middleware acts as a middleman between request and response. It is a type of filtering mechanism. For example, Laravel includes a middleware that verifies whether the user of the application is authenticated or not. If the user is authenticated, he will be redirected to the home page otherwise, he will be redirected to the login page.

There are two types of Middleware in Laravel.  
Global Middleware: will run on every HTTP request of the application.  
**Route Middleware**: will be assigned to a specific route.

**21. What is Lumen?**

**Lumen** is PHP micro-framework that built on Laravel’s top components.It is created by Taylor Otwell. It is perfect option for building Laravel based micro-services and fast REST API’s. It’s one of the fastest micro-frameworks available. Command   
composer create-project --prefer-dist laravel/lumen blog

**22. Explain Bundles in Laravel?**

In Laravel, bundles are referred to as packages. These packages are used to increase the functionality of Laravel. A package can have views, configuration, migrations, routes, and tasks

**23. How to use custom table in Laravel Modal ?**

You can use **custom table in Laravel** by overriding protected $table property of Eloquent. Below is sample uses

class User extends Eloquent{

protected $table="my\_user\_table"; }

**24. List types of relationships available in Laravel Eloquent?**

* Types of relationship in Laravel Eloquent are: 1) One To One 2) One To Many 3) Many To Many 4) Has Many Through, and 5) Polymorphic Relations.

**25. Why are migrations necessary?**

* Without migrations, database consistency when sharing an app is almost impossible, especially as more and more people collaborate on the web app.
* Your production database needs to be synced as well.

**26. List some Aggregates methods provided by query builder in Laravel ?**

* count(), max(), min(), avg(),sum()

**27. How to check request is ajax or not ?**

public function saveData(Request $request)

{

if($request->ajax()){

return "Request is of Ajax Type";

}

return "Request is of Http type";

}

**28. Explain Inversion of Control, how to implement it.**

**Inversion of control** is a design pattern that is used for decoupling components and layers of a system. Inversion of control(IOC) is implemented through injecting dependencies into a component when it is constructed.

**29. What is routing?**  
Routing is a way to define flexible request URL for application. Laravel offers different routing methdods like get,post,put,delete,patch etc.

**30. What is reverse routing in Laravel?**  
Reverse routing is a process to generate URL from the references given and route declaration at the time.

**31. What is named routing in Laravel?**  
We can specify name for route, this is called Named Routing.We can access the name of a route that is running via the route name method.

**32. What is CSRF protection?**  
CSRF or Cross-Site Request Forgery is an attack that may lead unwanted and harmful action. We need to secure state changing stions from attacks. Laravel provide inbuilt CSRF protection.

**33. What are query scopes?**  
Query scope is feature developed by Laravel where we can reuse similar queries.We don’t need to write similar types of queries again in the application.Once the scope has been defined, you may call the scope methods when querying the model.

**34. What is query builder in laravel?**  
Laravel’s database query builder provides a convenient, fluent interface to creating and running database queries. It can be used to perform most database operations in your application and works on all supported database systems.

**35. How can we run custom and raw query in laravel?**  
Calling raw() functions of DB class:

$result\_data = DB::raw(‘SELECT first\_name,email\_id from users’);

**36. What is HTTP middleware?**

HTTP middleware is a technique for filtering HTTP requests. Laravel includes a middleware that checks whether application user is authenticated or not.

**37. Why use Route?**

Routes are stored inside files under the /routes folder inside the project's root directory. By default, there are a few different files corresponding to the different "sides" of the application ("sides" comes from the hexagonal architecture methodology). .

**38. Explain important directories used in a common Laravel application.**

* App/: This is a source folder where our application code lives. All controllers, policies, and models are inside this folder.
* Config/: Holds the app's configuration files. These are usually not modified directly but instead, rely on the values set up in the .env (environment) file at the root of the app.
* Database/: Houses the database files, including migrations, seeds, and test factories.
* Public/: Publicly accessible folder holding compiled assets and of course an index.php file.

**39. State the difference between get and post method.**

Get method allows you to send a limited amount of data in the header. Post allows you to send a large amount of data in the body.

**40. List default packages of Laravel 5.6.**

Default packages of Laravel 5.6 are: 1) Envoy, 2) Passport, 3) Socialite, 4) Cashier, 5) Horizon, and 6) Scout.

**41 Explain dependency injection and their types.**

It is a technique in which one object is dependent on another object. There are three types of dependency injection: 1) Constructor injection, 2) setter injection, and 3) interface injection.

**42. How can you reduce memory usage in Laravel?**

While processing a large amount of data, you can use the cursor method in order to reduce memory usage.

**43. Name the Template Engine utilized by Laravel.**

Blade is a powerful template engine utilized by Laravel.

**44. Why are migrations important?**

Migrations are important because it allows you to share application by maintaining database consistency. Without migration, it is difficult to share any Laravel application. It also allows you to sync database.

**45. How can you generate URLs?**

Laravel has helpers to generate URLs. This is helpful when you build link in your templates and API response.

**46. Explain fluent query builder in Laravel.**

It is a database query builder that provides convenient, faster interface to create and run database queries.

**47. How to configure a mail-in Laravel?**

Laravel provides APIs to send an email on local and live server.

**48. Explain Auth.**

It is a method of identifying user login credential with a password. In Laravel it can be managed with a session which takes two parameters 1) username and 2) password.

**49. Differentiate between delete() and softDeletes().**

* delete(): remove all record from the database table.
* softDeletes(): It does not remove the data from the table. It is used to flag any record as deleted.

**50. How can you make real time sitemap.xml file in Laravel?**

You can create all web pages of a website to tell the search engine about the organizing site content. The crawlers of search engine read this file intelligently to crawl a website.

**51. Explain faker in Laravel.**

It is a type of module or packages which are used to create fake data. This data can be used for testing purpose.

It is can also be used to generate: 1) Numbers, 2) Addresses, 3) DateTime, 4) Payments, and 5) Lorem text.

**52. How will you check table is exists or in the database?**

Use hasTable() Laravel function to check the desired table is exists in the database or not.

**53. What is the significant difference between insert() and insertGetId() function in Laravel?**

Insert(): This function is simply used to insert a record into the database. It not necessary that ID should be autoincremented.

InsertGetId(): This function also inserts a record into the table, but it is used when the ID field is auto-increment.

**54. Explain active record concept in Laravel.**

In active record, class map to your database table. It helps you to deal with CRUD operation.

**55. List basic concepts in Laravel?**

Following are basic concepts used in Laravel:

* Routing, Eloquent ORM, Middleware, Security, Caching, Blade Templating

**56. Define Implicit Controller.**

Implicit Controllers help you to define a proper route to handle controller action. You can define them in route.php file with Route:: controller() method.

**57. How to use the custom table in Laravel Model?**

In order to use a custom table, you can override the property of the protected variable $table.

**58. What is MVC framework?**

It is Model, View, and Controller:

Model: Model defines logic to write Laravel application.

View: It covers UI logic of Laravel application.

Controller: It is work as an interface between Model, and View. It is a way how the user interacts with an application.

**59. Define @include.**

@include is used to load more than one template view files. It helps you to include view within another view. User can also load multiple files in one view.

**60. Explain the concept of cookies.**

Cookies are small file sent from a particular website and stored on PC by user's browser while the user is browsing.

**61. Which file is used to create a connection with the database?**

To create a connection with the database, you can use .env file.

**62. What is Eloquent?**

Eloquent is an ORM used in Laravel. It provides simple active record implementation working with the database. Each database table has its Model, which used to interact with the table.

**63. Name some Inbuilt Authentication Controllers of Laravel.**

RegisterController, LoginController, ResetPasswordController, ForgetPasswordController

**64. Define Laravel guard.**

Laravel guard is a special component that is used to find authenticated users. The incoming requested is initially routed through this guard to validate credentials entered by users. Guards are defined in ../config/auth.php file.

**65. What is Laravel API rate limit?**

It is a feature of Laravel. It provides handle throttling. Rate limiting helps Laravel developers to develop a secure application and prevent DOS attacks.

**66. Explain collections in Laravel.**

Collections is a wrapper class to work with arrays. Laravel Eloquent queries use a set of the most common functions to return database result.

**67. What is the use of DB facade?**

DB facade is used to run SQL queries like create, select, update, insert, and delete.

**68. What is the use of Object Relational Mapping?**

Object Relational Mapping is a technique that helps developers to address, access, and manipulate objects without considering the relation between object and their data sources.

**69. What is Ajax in Laravel?**

Ajax stands for Asynchronous JavaScript and XML is a web development technique that is used to create asynchronous Web applications. In Laravel, response() and json() functions are used to create asynchronous web applications.

**70. What is a session in Laravel?**

Session is used to pass user information from one web page to another. Laravel provides various drivers like a cookie, array, file, Memcached, and Redis to handle session data.

**71. How to access session data?**

Session data be access by creating an instance of the session in HTTP request. Once you get the instance, use get() method with a "Key" as a parameter to get the session details.

**72. State the difference between authentication and authorization.**

Authentication means confirming user identities through credentials, while authorization refers to gathering access to the system.

**73. Explain to listeners.**

Listeners are used to handling events and exceptions. The most common listener in Laravel for login event is LoginListener.

**74. What are policies classes?**

Policies classes include authorization logic of Laravel application. These classes are used for a particular model or resource.

**75. How to rollback last migration?**

Use need to use artisan command to rollback the last migration.

**76. What do you mean by Laravel Dusk?**

Laravel Dusk is a tool which is used for testing JavaScript enabled applications. It provides powerful, browser automation, and testing API.

**77. Explain Laravel echo.**

It is a JavaScript library that makes possible to subscribe and listen to channels Laravel events. You can use NPM package manager to install echo.

**78. What is make method?**

Laravel developers can use make method to bind an interface to concreate class. This method returns an instance of the class or interface. Laravel automatically inject dependencies defined in class constructor.

**79. Explain Response in Laravel.**

All controllers and routes should return a response to be sent back to the web browser. Laravel provides various ways to return this response. The most basic response is returning a string from controller or route. .

**80. What is namespace in Laravel?**

A namespace allows a user to group the functions, classes, and constants under a specific name.

**81. What is Laravel Forge?**

Laravel Forge helps in organizing and designing a web application. Although the manufacturers of the Laravel framework developed this toll, it can automate the deployment of every web application that works on a PHP server.

**82. State the difference between CodeIgniter and Laravel.**

|  |  |  |
| --- | --- | --- |
| **Parameter** | **CodeIgniter** | **Laravel** |
| Support of ORM | CodeIgniter does not support Object-relational mapping. | Laravel supports ORM. |
| Provide Authentication | It does provide user authentication. | It has inbuilt user authentication. |
| Programming Paradigm | It is component-oriented. | It is object-oriented. |
| Support of other Database Management System | It supports Microsoft SQL Server, ORACLE, MYSQL, IBM DB2, PostgreSQL, JDBC, and orientDB compatible. | It supports PostgreSQL, MySQL, MongoDB, and Microsoft BI, but CodeIgniter additionally supports other databases like Microsoft SQL Server, DB2, Oracle, etc. |
| HTTPS Support | Laravel supports custom HTTPS routes. The programmers can create a specific URL for HTTPS route they have defined. | CodeIgniter partially support HTTPS. Therefore, programmers can use the URL to secure the data transmission process by creating PATS. |

**83. What is an Observer?**

Model Observers is a feature of Laravel. It is used to make clusters of event listeners for a model. Method names of these classes depict the Eloquent event. Observers classes methods receive the model as an argument.

**84. What is the use of the bootstrap directory?**

It is used to initialize a Laravel project. This bootstrap directory contains app.php file that is responsible for bootstrapping the framework.

**85. In which folder robot.txt is placed?**

Robot.txt file is placed in Public directory.

**86. Explain API.PHP route.**

Its routes correspond to an API cluster. It has API middleware which is enabled by default in Laravel. These routes do not have any state and cross-request memory or have no sessions.

**87. what is open source software?**

Open-source software is a software which source code is freely available. The source code can be shared and modified according to the user requirement.

**88. Explain Loggin in Laravel.**

It is a technique in which system log generated errors. Loggin is helpful to increase the reliability of the system. Laravel supports various logging modes like syslog, daily, single, and error log modes.

**89. What is Localization?**

It is a feature of Laravel that supports various language to be used in the application. A developer can store strings of different languages in a file, and these files are stored at resources/views folder. Developers should create a separate folder for each supported language.

**90. Define hashing in Laravel.**

It is the method of converting text into a key that shows the original text. Laravel uses the Hash facade to store the password securely in a hashed manner.

**91. Explain the concept of encryption and decryption in Laravel.**

It is a process of transforming any message using some algorithms in such way that the third user cannot read information. Encryption is quite helpful to protect your sensitive information from an intruder.

Encryption is performed using a Cryptography process. The message which is to be encrypted called as a plain message. The message obtained after the encryption is referred to as cipher message. When you convert cipher text to plain text or message, this process is called as decryption.

**92. How to share data with views?**

To pass data to all views in Laravel use method called share(). This method takes two arguments, key, and value.

Generally, share() method are called from boot method of Laravel application service provider. A developer can use any service provider, AppServiceProvider, or our own service provider.

**93. Explain web.php route.**

Web.php is the public-facing "browser" based route. This route is the most common and is what gets hit by the web browser. They run through the web middleware group and also contain facilities for CSRF protection (which helps defend against form-based malicious attacks and hacks) and generally contain a degree of "state" (by this I mean they utilize sessions).

**How to generate a request in Laravel?**

php artisan make:request UploadFileRequest

#### ****94. What is the one best property of Laravel?****

Laravel supports MVC (model, view, controller) architecture. With MVC, a developer can keep the business logic of UI, and other functions away from each other which makes code clean and enables easy code reading and effortless code maintenance.

#### ****95. What is a Laravel package?****

#### A Laravel package is nothing but the easiest way of adding functionalities in the program. A Laravel package allows developers to achieve a task by writing the easiest code, having minimum lines.

#### ****96. What is the name of the Laravel package library? And how many packages does it store?****

A Laravel package library is called Packalyst and it stores more than 17000 packages.

#### ****97. In which language, Laravel has been written?****

Since Laravel is the PHP framework, it has been written in PHP language.

#### ****98. How you can check the current version of the installed Laravel?****

PHP artisan –version

#### ****99. How to use select, update and delete statements in Laravel?****

For update statement: DB::update(‘update users set user\_id=5 where id=?’,[2025]),

For delete statement: DB::delete(‘delete from users where id =?’, [2525]);

For select statement: $users = DB::select(‘select \* from users where user\_id=?’, 10);

#### ****100. What is the by default database in Laravel and how to change it?****

Mysql is the default database in Laravel and to change it, find config/database.php file. In it, search for ‘default=>’MySQL’. Here, change ‘MySQL’ to your database name.

#### ****101. What is maintenance mode and how to enable it in Laravel?****

Maintenance mode in Laravel can be used when one needs to modify the site. When you active maintenance mode on the site, users see maintenance messages as soon as they open your site. To enable maintenance mode, you just need to give an artisan command. (PHP artisan down). And to disable maintenance mode, give the command, PHP artisan up.

#### ****102. Can someone hack Laravel application?****

No, no one can ever hack Laravel application as Laravel has inbuilt CSRF security and encrypted sessions and cookies.

|  |  |
| --- | --- |
|  |  |

**103. What is Laravel Dusk?**

Laravel Dusk provides an expressive, easy-to-use browser automation and testing API. You’ll love it.

**104. What is Binding?**

A) Within a service provider, we always have access to the container via the $this->app property. We can register a binding using the bind method, passing the class or interface name that we wish to register along with a Closure that returns an instance of the class:

$this->app->bind(‘HelpSpot\API’, function ($app) {  
return new HelpSpot\API($app->make(‘HttpClient’));  
});

**105. Explain Binding A Singleton?**

A) The singleton method binds a class or interface into the container that should only be resolved one time. Once a singleton binding is resolved, the same object instance will be returned on subsequent calls into the container.

**106. Explain Binding Instances?**

A) You may also bind an existing object instance into the container using the instance method. The given instance will always be returned on subsequent calls into the container:

$api = new HelpSpot\API(new HttpClient);

$this->app->instance(‘HelpSpot\API’, $api);

**107. Explain Binding Primitives?**

A) Sometimes you may have a class that receives some injected classes, but also needs an injected primitive value such as an integer. You may easily use contextual binding to inject any value your class may need:

$this->app->when(‘App\Http\Controllers\UserController’)  
->needs(‘$variableName’)  
->give($value);

**108. Explain Extending Bindings?**

A) The extend method allows the modification of resolved services. For example, when a service is resolved, you may run additional code to decorate or configure the service. The extend method accepts a Closure, which should return the modified service, as its only argument:

$this->app->extend(Service::class, function($service) {  
return new DecoratedService($service);  
});

**109. What is the make Method?**

A) You may use the make method to resolve a class instance out of the container. The make method accepts the name of the class or interface you wish to resolve:

$api = $this->app->make(‘HelpSpot\API’);

**110. What is the Boot Method?**

if we need to register a view composer within our service provider? This should be done within the boot method. This method is called after all other service providers have been registered, meaning you have access to all other services that have been registered by the framework.

**111. Where do you** register **service providers?**

All service providers are registered in the config/app.php configuration file. This file contains a providers array where you can list the class names of your service providers.

**112. How do you register service providers?**

To register your provider, add it to the array:

‘providers’ => [  
// Other Service Providers

App\Providers\ComposerServiceProvider::class,  
],

**113. What are the benefits of Facades?**

Facades have many benefits. They provide a terse, memorable syntax that allows you to use Laravel’s features without remembering long class names that must be injected or configured manually. Furthermore, because of their unique usage of PHP’s dynamic methods, they are easy to test.

**114. Difference between Facades Vs. Dependency Injection?**

One of the primary benefits of dependency injection is the ability to swap implementations of the injected class. This is useful during testing since you can inject a mock or stub and assert that various methods were called on the stub.

Typically, it would not be possible to mock or stub a truly static class method. However, since facades use dynamic methods to proxy method calls to objects resolved from the service container, we actually can test facades just as we would test an injected class instance.

**115. What is the difference between Facades** Vs**Helper Functions?**

In addition to facades, Laravel includes a variety of “helper” functions which can perform common tasks like generating views, firing events, dispatching jobs, or sending HTTP responses. Many of these helper functions perform the same function as a corresponding facade.

**116. What are Laravel’s Contracts?**

Laravel’s Contracts are a set of interfaces that define the core services provided by the framework. For example, a Illuminate\Contracts\Queue\Queue contract defines the methods needed for queueing jobs, while the Illuminate\Contracts\Mail\Mailer contract defines the methods needed for sending e-mail.

**117. What is the difference between Contracts Vs Facades?**

Laravel’s facades and helper functions provide a simple way of utilizing Laravel’s services without needing to type-hint and resolve contracts out of the service container. In most cases, each facade has an equivalent contract.

Unlike facades, which do not require you to require them in your class’ constructor, contracts allow you to define explicit dependencies for your classes. Some developers prefer to explicitly define their dependencies in this way and therefore prefer to use contracts, while other developers enjoy the convenience of facades.

**118. Where do you locate Route files?**

All Laravel routes are defined in your route files, which are located in the routes directory.

**119. What is Route Model Binding?**

When injecting a model ID to a route or controller action, you will often query to retrieve the model that corresponds to that ID. Laravel route model binding provides a convenient way to automatically inject the model instances directly into your routes..

**120. What are Middleware Groups?**

Sometimes you may want to group several middleware under a single key to make them easier to assign to routes. You may do this using the $middlewareGroups property of your HTTP kernel.

**121. What is X-CSRF-TOKEN?**

In addition to checking for the CSRF token as a POST parameter, the VerifyCsrfToken middleware will also check for the X-CSRF-TOKEN request header. You could, for example, store the token in a HTML meta tag:

<meta name=”csrf-token” content=”{{ csrf\_token() }}”>

**122. What is X-XSRF-TOKEN?**

Laravel stores the current CSRF token in a XSRF-TOKEN cookie that is included with each response generated by the framework. You can use the cookie value to set the X-XSRF-TOKEN

**123. What is** url **helper?**

A) The url helper may be used to generate arbitrary URLs for your application. The generated URL will automatically use the scheme (HTTP or HTTPS) and host from the current request:

$post = App\Post::find(1);

echo url(“/posts/{$post->id}”);.

**124. What is the render method?**

The render methos is responsible for converting a given exception into an HTTP response that should be sent back to the browser. By default, the exception is passed to the base class which generates a response for you.

**125. What is Monolog library?**

Laravel utilizes the Monolog library, which provides support for a variety of powerful log handlers. Laravel makes it a cinch to configure these handlers, allowing you to mix and match them to customize your application’s log handling.

**126. What is stack channel?**

By default, Laravel will use the stack channel when logging messages. The stack channel is used to aggregate multiple log channels into a single channel..

**127. Does Laravel Support Caching?**

Yes, Its provides.

**128. How To Use Delete Statement In Laravel?**

DB::delete('delete from  users where id = ?', [1015]);

**129. How To Use Update Statement In Laravel?**

DB::update('update users set city\_id = 10 where id = ?', [1015]);

**130. How To Use Insert Statement In Laravel?**

DB::insert('insert into users (id, name, city\_id) values (?, ?)', [1, 'Web technology',10]);

**131. How To Use Select Query In Laravel?**

$users = DB::select('select \* from users where city\_id = ?', 10);

if(!empty($users)){  
    foreach($users as $user){  
    }  
}

**132. How To Enable The Query Logging?**

DB::connection()->enableQueryLog();

**133. How To Set Database Connection In Laravel?**

Database configuration file path is : config/database.php

**134. What Are Bundles,reverse Routing And The Ioc Container ?**

* **Bundles:** These are small functionality which you may download to add to your web application.
* **Reverse Routing:** This allows you to change your routes and application will update all of the relevant links as per this link.
* **IoC container:** It gives you Control gives you a method for generating new objects and optionally instantiating and referencing singletons.

**135. Compare Laravel With Codeigniter?**

**Laravel :**

* Laravel is a framework with expressive, elegant syntax
* Development is enjoyable, creative experience
* Laravel is built for latest version of PHP
* It is more object oriented compared to CodeIgniter
* Laravel community is still small, but it is growing very fast.

**Codeigniter :**

* CodeIgniter is a powerful PHP framework
* Simple and elegant toolkit to create full-featured web applications.
* Codeigniter is an older more mature framework
* It is less object oriented compared to Laravel.
* Codeigniter community is large.