What is Laravel?

Laravel is a Symfony 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.Here you will find Latest Laravel interview questions and answer that helps you crack Laravel Interviews.

Quick Questions About Laravel	
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	6.9.0

Q1. What is Laravel?

Laravel is a free open source "PHP framework" based on the MVC design pattern.
It is created by Taylor Otwell. Laravel provides expressive and elegant syntax that helps in creating a wonderful web application easily and quickly.

Q2. 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.

Q3. 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.

Laravel validation Example

```
$validatedData = $request->validate([
            'name' => 'required|max:255',
            'username' => 'required|alpha_num',
            'age' => 'required|numeric',
        ]);
```

Q4. How to install laravel via composer ?

You can install Laravel via composer by running below command.

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

Also Read Core PHP Interview Questions and Answers for 2019

Q5. List some features of laravel 6 ?

Laravel 6 features

- Inbuilt CRSF (cross-site request forgery) Protection.
- Inbuilt paginations
- Reverse Routing
- Query builder
- Route caching
- Database Migration
- IOC (Inverse of Control) Container Or service container.
- Job middleware
- Lazy collections

Q6. 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
- php artisan help
- php artisan tinker
- php artisan make
- php artisan –versian
- php artisan make model model_name
- php artisan make controller controller_name

Q7. List some default packages provided by Laravel Framework?

Below are a list of some official/ default packages provided by Laravel

- Cashier
- Envoy
- Passport
- Scout
- Socialite
- Horizon
- Telescope

Q8. 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');
```

Q9. 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.

Use below commands to create migration data via artisan.

```
// creating Migrationphp artisan make:migration create_users_table
```

Q10. 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, or any other piece of functionality within the register method.
You can read more about service provider from here

Q11. 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.

Q12. 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.

Q13. What is dependency injection in Laravel ?

In software engineering, dependency injection is a technique whereby one object supplies the dependencies of another object. A dependency is an object that can be used (a service). An injection is the passing of a dependency to a dependent object (a client) that would use it. The service is made part of the client's state.[1] Passing the service to the client, rather than allowing a client to build or find the service, is the fundamental requirement of the pattern.

https://en.wikipedia.org/wiki/Dependency_injection

You can do dependency injection via Constructor, setter and property injection.

Q14. 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.
Read more about laravel Contract’s

Q15. 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');
});
```

Q16. 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();
```

Q17. How to enable query log in Laravel ?

Use the **enableQueryLog** method to enable query log in Laravel

```
DB::connection()->enableQueryLog();
You can get array of the executed queries by using getQueryLog method:
$queries = DB::getQueryLog();
```

Q18. 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 declaration

```
Route::get('login', 'users@login');
```

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.

Q19. How to turn off CRSF protection for specific route in Laravel?

To turn off **CRSF protection in Larave**l add following codes in “app/Http/Middleware/VerifyCsrfToken.php”

```
//add an array of Routes to skip CSRF check
private $exceptUrls = ['controller/route1', 'controller/route2'];
 //modify this function
public function handle($request, Closure $next) {
 //add this condition foreach($this->exceptUrls as $route) {
 if ($request->is($route)) {
  return $next($request);
 }
}
return parent::handle($request, $next);}
```

Q20. 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.

Here is an example of trait.

```
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'
```

Q21. Does Laravel support caching?

Yes, Laravel supports popular caching backends like **Memcached** and **Redis**.
By default, Laravel is configured to use the **file cache driver**, which stores the serialized, cached objects in the file system.For large projects, it is recommended to use Memcached or Redis.

Q22. 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.
Read more about Laravel middlewares

Q23. 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.
You can install Lumen using composer by running below command

```
composer create-project --prefer-dist laravel/lumen blog
```

Q24. Explain Bundles in Laravel?

In Laravel, **bundles** are also called packages. Packages are the primary way to extend the functionality of Laravel. Packages might be anything from a great way to work with dates like Carbon, or an entire BDD testing framework like Behat.In Laravel, you can create your custom packages too. You can read more about packages from here

Q25. 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";
}
```

Q26. List types of relationships available in Laravel Eloquent?

Below are types of relationships supported by Laravel Eloquent ORM.

- One To One
- One To Many

- One To Many (Inverse)
- Many To Many
- Has Many Through
- Polymorphic Relations
- Many To Many Polymorphic Relations

You can read more about relationships in Laravel Eloquent from here

Q27. Why are migrations necessary?

Migrations are necessary because:

- 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.

Q28. Provide System requirements for installation of Laravel framework ?

In order to install Laravel, make sure your server meets the following requirements:

- PHP >= 7.1.3
- OpenSSL PHP Extension
- PDO PHP Extension
- Mbstring PHP Extension
- Tokenizer PHP Extension
- XML PHP Extension
- Ctype PHP Extension
- JSON PHP Extension

Q29. List some Aggregates methods provided by query builder in Laravel ?

- count()
- max()
- min()
- avg()
- sum()

Also Read **Laravel 5 interview questions 2019**

Q30. What is Laravel Vapor?

It is a serverless deployment platform that is powered by AWS.**Laravel Vapor** provides on-demand autoscaling with zero server maintenance.

Q31. 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

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

Points to be remembered while appearing in Laravel Interviews :

- **Laravel** is developed on the MVC (Model-View-Controller) design pattern.
- **Laravel** comes with inbuilt features/ modules like authentication, authorization, localization, models, views, sessions, paginations and routing
- **Laravel** supports advanced concepts of PHP and OOPs like Dependency Injection, traits, Contracts, bundles, Namespaces, Facades
- **Laravel** supports Multiple Databases like MySQL, PostgreSQL, SQLite, SQL Server.
- **Laravel** allows developers to write a clean and modular code.
- **Laravel** supports blade Template Engine
- **Laravel** comes with Official Packages like Cashier, Envoy, Horizon, Passport, Scout, Socialite
- **Laravel** can be used with various popular Javascript Frameworks Like AngularJs, VueJs, ReactJS.

Frequently Asked Laravel Developer Interview Questions.

1. What is Laravel migration?

Migration in the Laravel is used to modify and share the structure of your database. It is paired with the schema builder in the Laravel to build the database for your application. It acts as a version control but for the Laravel database. There are two methods in the migration class, up and down. Up method is used to add tables, columns, to the database. The down method is used to reverse the operations of up. To create a migration, use **make:migration** command. To run the migration, use migrate command. It also offers a rollback feature.

2. Which version of Laravel you have used?

   Answer the versions of Laravel you have worked on like Laravel 4, Laravel 5, Laravel 5.5, Laravel 6

3. What is socialite Laravel?

   Laravel Socialite is an OAuth provider that is used to authenticate social media apps such as Facebook, Twitter, Google, and many more simply and seamlessly. It can be added to your Laravel web application using the dependency manager composer.

4. Why the composer is used in Laravel?

   The **composer in the Laravel** is a dependency management tool that is used to install and manage libraries in your application. It does not install packages globally but on a per-application basis. Define the packages you want in the composer.json file and run the following command to install the packages in your application.

5. How can you update Laravel?

   To upgrade your Laravel framework to the latest version, open the composer.json file and change the version of the Laravel framework to the newest version. Then run the following command to update your Laravel framework.

   ```
   composer update
   ```

6. How to use Laravel tinker?

   Laravel Tinker is a REPL that allows the developer to interact with the Laravel application through the command line. To enter into the Tinker environment, enter the following command

   ```
   php artisan tinker
   ```

   Tinker allows you to run commands such as clear-compiled, down, env, inspire, migrate, optimize, and up by default. To use more commands using Tinker, add the command to the command array that is present in the tinker.php configuration file.

7. How to use where relationship Laravel?

   The **where clause** is a query builder that is used to verify something. Where has three arguments that it must require. The column name is the first argument with an operator as the second argument. The third argument is the value that the first operand must be evaluated using the operator.

   Example

```
$users = DB::table('users')->where('salary', '=', 100)->get();
```

In the above example, the where clause is used to get the users who have the salary column that is equal to 100.

8. What is the difference between Laravel find and where?

   The where clause in the query is used to verify and retrieve values. It has three arguments. First is the column name argument. Second is the operator. The third is the values that need to be checked with the column name. It retrieves the first model that is matching the given query.

   Example

   ```
   $users = DB::table('students')->where('marks', '=', 100)->get();
   ```

   The find method is used to retrieve a single model instance instead of a collection of models. It retrieves the model by its primary key.

   ```
   $users = StudentModel::find(1);
   ```

9. How to hash password in Laravel?

   The Hash::make function is used to create a hash for the password.

   Syntax:

   ```
   Hash::make($password)
   ```

10. What are the Laravel guards?

    The **guard in Laravel** is used to define the user authentication for each request. The Guards also provide a definition of user information storage and retrieval by the system. The session guard is a default guard in Laravel that is used to maintain the state using session storage and cookies. A web guard is used to store and retrieve session information. API guard is used to authenticate users and requests. These are some examples of default Laravel guard, and you can also create your own guard in Laravel.