

BASIC LARAVEL INTERVIEW QUESTIONS AND ANSWERS

1. Can Laravel be used for Full Stack Development (Frontend + Backend)?

[Hide Answer](#)

Laravel is the best choice for creating progressive, scalable full-stack web applications. Full-stack web applications can have a backend written in Laravel and a frontend built with blade files or SPAs built with Vue.js, which is included by default. However, you can also use it simply to provide rest APIs to an SPA application. As a result, Laravel can be used to create full-stack applications and just backend APIs.

2. Tell us about your prior experience as a remote Laravel developer.

[Hide Answer](#)

With such questions, recruiters often look to get a clear understanding of your experience in the remote-first model. If you have prior experience of working in such a role, it will help to improve your chances of getting selected for the role. To increase your chances, you could provide a quick and crisp peek into your daily work life.

3. Have you ever faced any challenges while working in distributed teams?

[Hide Answer](#)

The recruiter may want to know what kind of hurdles have you faced through your journey as a remote developer. You could try answering these questions by stating not just the challenges but also what measures you took to overcome them.

4. What kind of tools did you use for the remote role to stay connected?

[Hide Answer](#)

Companies often look to bring in people who can blend into new operational structures quickly. State the types of applications or software you use on a daily basis along with the purpose of each to shed light on your remote work experience. Mention the types of tools you use on daily basis for - communication, calls, project management, and more.

5. Do you find it difficult to focus on work as a remote developer?

[Hide Answer](#)

The recruiter is trying to understand your capability to cope with the remote structure. You can explain how you manage tasks and deadlines without any monitoring. This will help in creating a good impression on the recruiters.

6. What is the latest version of Laravel?

[Hide Answer](#)

The most latest version of Laravel is version 10.12, which includes the addition of conditional methods to Sleep, a newly introduced job timeout event, and validation parameters for time zones.

7. Mention the databases supported by Laravel.

[Hide Answer](#)

Laravel, through its Eloquent ORM (Object-Relational Mapping), supports various relational database systems, allowing developers to choose the one that best fits their application requirements. The supported databases include:

MySQL: An open-source, widely-used relational database management system.

PostgreSQL: A powerful, open-source object-relational database management system with an emphasis on extensibility and standards compliance.

SQLite: A self-contained, serverless, and zero-configuration database engine, ideal for development and testing environments.

SQL Server: A proprietary database management system developed by Microsoft.

8. Explain what is composer in Laravel.

[Hide Answer](#)

In Laravel, the composer is a tool that works as a package manager for the framework. The composer helps to add fresh new packages and libraries from different sources directly into the application. One of the most common examples used for authentication purposes is a Passport. It generates a composer.json in the project directory to track all linked packages. Passport package can easily be easily installed with composer.

9. What is maintenance mode in Laravel.

[Hide Answer](#)

In Laravel, maintenance mode is a feature that allows you to put your application into a temporary "maintenance mode" in order to perform updates, upgrades, or maintenance on the server without affecting the user experience.

When maintenance mode is enabled, users trying to access your application will see a "503 Service Unavailable" error page, indicating that the site is currently down for maintenance.

To enable maintenance mode in Laravel, you can run the php artisan down command from the command line. This will create a down file in the storage directory, which will trigger the maintenance mode.

10. What is a Route in Laravel?

[Hide Answer](#)

A route in Laravel refers to the mapping of a user-requested URL to a specific action, method, or callback function within the application. In other words, it defines the way an application responds to a client request through a specific URL pattern.

Routes play a crucial role in organizing the flow and handling of HTTP requests in a Laravel application. They can be defined in the route files stored within the routes directory of your project. The primary route files include `web.php`, `api.php`, `console.php`, and `channels.php`.

11. Mention the default route files used in Laravel?

[Hide Answer](#)

In Laravel, the default route files are used to define various types of routes for your application. These files are stored in the routes directory within your project. The main default route files include:

web.php: This file contains routes intended for web applications, which need features like sessions, CSRF protection, and cookies. These web-based routes are assigned the web middleware group.

api.php: This file is dedicated to routes that cater to APIs, where stateless requests are expected. Routes defined in this file are automatically assigned the api middleware group, which includes rate limiting and token-based authentication.

console.php: This file is used for defining Artisan commands; these custom commands can be executed in the command line, making it easier to manage and automate tasks within your application.

channels.php: This file is designed for defining and customizing broadcast channels for your Laravel application. It handles real-time data broadcasting using WebSockets and handles channel authorization.

12. What is soft delete in Laravel?

[Hide Answer](#)

In Laravel, soft delete refers to a feature that allows you to "delete" a record from the database without permanently removing it. When a record is soft deleted, a timestamp is added to the record's `deleted_at` column instead of the record being entirely removed from the table. This way, the record is marked as deleted but is still retained in the database.

Soft delete functionality is useful when you want to allow for the recovery or restoration of deleted records in case of accidental deletions or when displaying a history of deleted items is needed in your application.

13. Explain what are models in Laravel.

[Hide Answer](#)

In Laravel, models are a crucial part of the MVC (Model-View-Controller) architectural pattern. A model represents the data structure and the business logic of your application. It interacts with the database by providing an abstraction layer for working with your data. Models communicate with tables in the database, allowing you to fetch, insert, update, and delete records through an object-oriented syntax.

Laravel uses the Eloquent ORM (Object-Relational Mapping), which is an active record implementation, making it simple to work with database records as objects. With Eloquent, you can easily map your database tables to models within your application. Each model corresponds to a single table in the database, and each instance of that model represents a row or record within the table.

14. What is the process of defining environment variables in Laravel?

[Hide Answer](#)

In Laravel, environment variables are used to store and manage sensitive configuration values, such as API keys, database credentials, or other settings that

may vary between different development, staging, or production environments. By using environment variables, you can keep sensitive data and environment-specific configurations separate from your codebase, ensuring better security and maintainability.

15. Explain migrations in Laravel.

[Hide Answer](#)

Migrations in Laravel are a powerful feature that allows you to manage and version your database schema through code. They are like version control for your database, enabling you to modify your database schema in a clear, structured, and maintainable way.

Migrations in Laravel work by creating schema files that contain instructions to create, update, or delete tables and columns in the database. These schema files are stored in the database/migrations directory and are executed in the order specified by their timestamps.

16. What is the importance of migration in Laravel?

[Hide Answer](#)

Migrations play a crucial role in Laravel development by providing an organized, version-controlled, and collaborative approach to handle database schema changes. The importance of migrations in a Laravel project can be attributed to the following factors:

Version Control: Migrations act like version control systems for your database schema, allowing you to track and manage schema changes in a structured and reusable manner. This makes it easier to roll back to previous schema states if needed.

Collaboration: When working with a team, migrations help keep the database schema consistent across all development environments. Developers can share and sync schema changes with others, minimizing conflicts and ensuring coherence.

Deployment: Migrations make deploying applications to different environments, such as staging and production, more streamlined. You can apply incremental schema changes to the target environment without having to reproduce the entire database manually.

Code-Based Schema Management: Migrations allow you to modify the database schema using PHP and Laravel's schema builder, which makes your schema more readable, maintainable, and versioned along with your code.

Database Agnosticism: Since Laravel migrations use the schema builder, migrating between different database systems (MySQL, PostgreSQL, SQLite, or SQL Server) becomes easier, as you don't need to write raw SQL queries specific to a particular database system.

17. What is the significance of seeders in Laravel?

[Hide Answer](#)

In Laravel, seeders are a valuable feature that facilitates the population of your database tables with sample or default data. Seeders are PHP classes used to define and manage test or initial data for your application. They are especially helpful during the development phase, where quickly filling tables with data aids in testing, presenting, and developing features without manual input.

18. What are 'Bundles' in Laravel?

[Hide Answer](#)

In Laravel 3.x, 'Bundles' were a way of extending the functionality of the core Laravel framework by adding third-party packages or modules. These bundles were similar to what we now know as packages in the modern Laravel ecosystem.

However, since Laravel 4.x introduced the improved package management system based on the composer, the term 'Bundles' is no longer officially used. Instead, packages are used for incorporating additional features or functionality into your Laravel projects. These packages can be installed and managed through the

Composer dependency management tool, making it easier to include and maintain third-party code in your applications.

19. Mention some essential directories used in Laravel applications.

[Hide Answer](#)

Laravel applications have a well-structured directory hierarchy to organize and manage different components of the application. Here are some essential directories used in Laravel applications:

app: This directory contains the core components of a Laravel application, including models, providers, console commands, and mail classes. It also houses the `Http` directory, where controllers, middleware, and other HTTP handling logic are stored.

config: This directory stores all configuration files related to the application, such as database connection settings, caching drivers, mail configurations, and so on.

database: This directory contains all files related to the application's database structure, such as migration files, seeders, and factories.

public: This directory acts as the public-facing entry point for your web application, holding assets like compiled CSS, JavaScript files, and images. It also contains the `index.php` file, which is the entry point for incoming HTTP requests.

20. Explain what is a controller?

[Hide Answer](#)

A controller is an essential component in the Model-View-Controller (MVC) architectural pattern, which most modern web frameworks, including Laravel, adhere to. In the context of web applications, a controller acts as an intermediary between the model (data) and the view (presentation). It receives input from the user (typically in the form of HTTP requests) and processes that input based on the application's business logic.

Controllers are responsible for the following tasks:

- Handling and processing incoming HTTP requests.

- Interacting with models to retrieve, store, update, or manipulate data.
 - Processing the data to prepare it for displaying in the view, if needed.
 - Rendering the appropriate view and returning the response to the user.
-

21. Can you explain reverse routing in Laravel?

[Hide Answer](#)

In Laravel, the process used for generating URL based on symbols or names is referred to as reverse routing. It allows developers to create links to specific routes within their application without hard-coding the URL. This can be particularly useful when working with dynamic or changing URLs, as it allows the URL to be generated automatically based on the name of the route.

22. What are contracts in Laravel?

[Hide Answer](#)

In Laravel, contracts are a set of interfaces that define the core services provided by the framework. Contracts offer a standardized and clearly defined API for these services, which can include authentication, caching, mail, queue, session, and many others. By adhering to their corresponding contracts, different implementations of the services can be used without altering the core Laravel code.

The primary purpose of contracts in Laravel is to ensure code flexibility, maintainability, and robustness. Contracts promote better code organization and ease the process of creating testable, decoupled, and interchangeable implementations of various services.

23. What are events?

[Hide Answer](#)

Events in Laravel are a way to implement the event-driven programming paradigm, where actions and communication between different components of your

application are orchestrated through events and event listeners. Events act as hooks or points of interest within your application's lifecycle where certain actions or behaviors should be triggered. Event listeners, on the other hand, are responsible for handling specific events by executing code in response to those events.

24. State some of the advantages of selecting Laravel.

[Hide Answer](#)

Laravel is a popular and feature-rich PHP framework that offers several advantages for building modern web applications:

Eloquent ORM: Laravel's built-in Object Relational Mapping (ORM) provides an elegant, expressive, and easy-to-use way to interact with databases using an object-oriented approach.

MVC Architecture: Laravel follows the Model-View-Controller (MVC) architectural pattern, promoting a clear separation of concerns for better code organization, maintainability, and scalability.

Routing: Laravel's simple, yet powerful routing system makes it easy to define and manage routes for your application, resulting in clean and expressive URL structures.

Blade Templating Engine: Laravel includes the Blade templating engine, which provides an intuitive, convenient, and secure way to build user interfaces and render views with reusable components.

Built-in Authentication: Laravel offers out-of-the-box user authentication and authorization, making it easy to set up secure and functional systems for user registration, login, password management, and access control.

Artisan Console: Laravel's Artisan CLI tool comes with numerous built-in commands for common tasks such as generating migrations, controllers, models, and more. It also supports the creation of custom commands to automate repetitive tasks.

Database Migrations & Seeding: Laravel's migrations and seeders allow you to manage, version, and automate your database schema and sample data in a unified and clear way.

25. In short, how would you reduce memory usage in Laravel?

[Hide Answer](#)

Developers often use the ‘cursor method’ while processing large amounts of data on Laravel. The method is well regarded for its exceptional speed and reduced memory usage. When you use the cursor method, Laravel retrieves the results in small chunks instead of loading the entire dataset into memory, which helps to avoid memory issues when dealing with large datasets.

The cursor method uses PHP generators to yield a single record at a time, instead of loading the entire dataset into memory all at once. By yielding one record at a time, the cursor method allows you to process extensive datasets in a memory-efficient manner, without consuming a significant amount of RAM.

26. Mention the types of relationships found in Laravel Eloquent.

[Hide Answer](#)

Laravel Eloquent provides a simple way to define relationships between your models, making it easy to manage related data in your database. The following are the primary types of relationships supported by Eloquent:

One-to-One: This relationship is used when one record in a table is associated with exactly one record in another table. To define a one-to-one relationship, use the `hasOne` and `belongsTo` Eloquent methods in their respective models.

One-to-Many: This relationship represents a situation when one record in a table is associated with multiple records in another table. Use the `hasMany` method in the parent model and the `belongsTo` method in the related model to define a one-to-many relationship.

Many-to-One: This is the inverse of a one-to-many relationship, where multiple records in one table are related to a single record in another table. The `belongsTo` method is used to establish a many-to-one relationship.

Many-to-Many: In this relationship, multiple records in one table are associated with multiple records in another table. To define a many-to-many relationship, use the

belongsToMany method in both models. Additionally, an intermediary table (pivot table) is required to store the associations.

Has-One-Through: This is an indirect one-to-one relationship where a relationship exists between two distant tables through an intermediary table. Use the hasOneThrough method in the initial model to define this relationship.

Has-Many-Through: This is an indirect one-to-many relationship where one record in a table is associated with multiple records in another table via an intermediary table. Use the hasManyThrough method in the initial model to define this relationship.

Polymorphic Relationships: This type of relationship allows a model to be associated with multiple types of related models on a single association. Laravel supports both one-to-many and many-to-many polymorphic relationships, using the morphTo, morphOne, morphMany, morphToMany, and morphedByMany methods.

Many-to-Many Polymorphic Relationships: This is an extension of polymorphic relationships where many-to-many relationships can occur between different types of related models. Use the morphToMany and morphedByMany methods in the relevant models to define this relationship.

27. What is Lumen?

[Hide Answer](#)

Lumen is a micro-framework created by Taylor Otwell, the same person who developed Laravel. Often referred to as a smaller, faster, and lighter version of Laravel, Lumen is built for speed and is specifically designed for building microservices, APIs, and simple web applications.

Lumen retains the familiar Laravel syntax and core components, but it removes some built-in features that are not crucial to building small to medium-sized applications, resulting in better performance and a smaller footprint.

28. What does Laravel use to generate URLs?

[Hide Answer](#)

Laravel provides a built-in URL generation system that makes it easy to generate URLs for your application's routes and assets. The main components used by Laravel to generate URLs are:

URL Helpers: Laravel includes several global helper functions such as `url()`, `asset()`, `route()`, and `action()`, which help generate URLs based on specified routes, controller actions, or asset files. These functions consider your application's base URL, making it easy to generate the correct URL structure for your application.

Named Routes: Named routes in Laravel allow you to assign a name to a specific route, making it easier to generate URLs for that route without hardcoding the actual URI. You can use the `route()` helper function with the assigned name to generate a URL for that specific route.

Route Parameters: When defining routes with parameters, Laravel allows you to generate URLs that include those parameters using the same URL helpers, such as `route()` and `action()`.

URL Generator Class: Laravel also includes an `Illuminate\Contracts\Routing\UrlGenerator` contract that provides a fluent interface for generating URLs. You can inject this contract into your classes, or use it through the `app('url')` service container binding.

29. What is socialite in Laravel?

[Hide Answer](#)

Laravel Socialite is a quick method for authenticating users using the OAuth providers. It allows developers to provide a quick link or simple button for users to click and initiate the authentication process from the home page. It simplifies the process of authentication with popular social networks such as Facebook, Google, Twitter, LinkedIn, and GitHub.

30. Explain the Laravel cursor method.

[Hide Answer](#)

The cursor method in Laravel is a memory-efficient solution for processing large datasets retrieved from your database, particularly when using Eloquent ORM. Rather than loading the entire dataset into memory at once, the cursor method allows you to process the records one by one, using PHP generators to yield each record individually.

31. In Laravel, which class is used for handling exceptions?

[Hide Answer](#)

In Laravel, the `App\Exceptions\Handler` class is responsible for handling exceptions. It extends the `Illuminate\Foundation\Exceptions\Handler` class, which is provided by the Laravel framework. The Handler class is located in the `app/Exceptions` directory.

The Handler class contains two primary methods for handling exceptions:

report: This method is used to log exceptions or send them to external services, such as error monitoring tools. By default, it logs exceptions to the file specified in the `config/logging.php` configuration file in the 'log' channel.

render: This method is responsible for converting the given exception into an HTTP response that will be sent to the user. It can also handle different exception types, such as 404 Not Found or 500 Internal Server Errors, and render appropriate error pages or JSON responses based on the application context (web or API).

32. Mention some commonly used artisan commands in Laravel.

[Hide Answer](#)

PHP artisan down: This command puts the application into maintenance mode, which allows you to perform maintenance tasks without affecting your users.

PHP artisan up: This command takes the application out of maintenance mode, allowing users to access the site once again.

PHP artisan make:controller: This command generates a new controller class for your application.

PHP artisan make:model: This command generates a new Eloquent model class for your application.

PHP artisan make:migration: This command generates a new database migration file, which allows you to modify your database schema.

PHP artisan make:middleware: This command generates a new middleware class for your application, which can be used to modify HTTP requests and responses.

Looking for remote developer job at US companies?

Work at Fortune 500 companies and fast-scaling startups from the comfort of your home

Apply Now

INTERMEDIATE LARAVEL DEVELOPER INTERVIEW QUESTIONS AND ANSWERS

1. What are CSRF tokens in Laravel?

[Hide Answer](#)

CSRF (Cross-Site Request Forgery), is a unique value. This value is generated from the server side of the application and forwarded to the client. The tokens help to safeguard web apps and its users from intrusions, i.e CSRF attacks.

The following is an example of how CSRF tokens are incorporated within forms in Laravel.

```
<form action="/user" method="POST">  
@csrf  
...  
</form>
```

2. What is the process of sharing data with views?

[Hide Answer](#)

In Laravel, you can share data with views through different approaches, allowing you to pass variables or objects from the controller to the view. The most common methods of sharing data with views are:

Passing Data Directly: The simplest and most direct way of sharing data with views is passing an array of values while rendering the view using the view function or helper.

```
return view('my_view', ['key' => 'value']);
```

Using compact Function: You can use the compact function to pass multiple variables as an associative array to the view.

```
$key1 = 'value1';  
$key2 = 'value2';  
return view('my_view', compact('key1', 'key2'));
```

Using with Method: You can chain the with method when rendering the view to pass data to the view. The with method takes a key-value pair as arguments.

```
return view('my_view')->with('key', 'value');
```

View Creators and Composers: View creators and composers are used when you need to share data with a view every time it is loaded, irrespective of the controller. You define these in a service provider, and they take the form of callbacks or class methods.

```
// Using closures  
View::composer('my_view', function ($view) {  
    $view->with('key', 'value');  
});  
  
// Using a class  
View::composer('my_view', MyViewComposer::class);  
  
class MyViewComposer  
{  
    public function compose(View $view)  
    {  
        $view->with('key', 'value');  
    }  
}
```


Sharing Global Data: You can use the share method on the View façade to share data globally with all views in your application.

```
View::share('key', 'value');
```

3. What is web.php route?

[Hide Answer](#)

In Laravel, the web.php route file is used to define web routes, which are the URL patterns and corresponding actions for your web application. The web.php file is stored in the routes directory within your Laravel project.

Routes defined in the web.php file are intended specifically for web-based applications that require features like sessions, cookies, and CSRF protection. Laravel automatically assigns the web middleware group to these routes to provide these necessary functionalities.

4. What is the simplest method for generating a request in Laravel?

[Hide Answer](#)

In Laravel, developers can easily generate a request using the command:
php artisan make:request UploadFileRequest. This command would generate a new UploadFileRequest class in the app/Http/Requests directory. You can then define validation rules for this request in the rules method of the generated class.

5. What is the common location of model files in Laravel apps?

[Hide Answer](#)

In Laravel applications, the common location for storing model files is the app directory, which is located at the root level of your project. Within the app directory,

you'll find the Models folder in Laravel 8 and newer versions. In older versions (before Laravel 8), model files were placed directly inside the app directory.

6. What is MVC architecture?

[Hide Answer](#)

MVC architecture refers to the a design pattern popularly used for building web application and is highly regarded for speeding up the process. It comes prebuilt with three different components - Model, View, and Controller.

The Model acts as the central component of the design and is responsible for managing the in-app data. View is responsible for displaying data to users and Controllers are used for handling user requests.

7. What are the server requirements for installing Laravel 8?

[Hide Answer](#)

To install Laravel 8, ensure that your server meets the following requirements:

PHP: Laravel 8 requires PHP 7.3 or later, although it is recommended to have the latest PHP version (8.x).

BCMath PHP Extension: Required for arbitrary precision mathematics.

Ctype PHP Extension: Necessary for character validation and manipulation.

JSON PHP Extension: Mandatory for handling JSON objects efficiently.

Mbstring PHP Extension: Supports multi-byte string handling.

OpenSSL PHP Extension: Required for establishing secure connections and encrypting data.

PDO PHP Extension: Enables access to databases using PHP's Data Objects' abstraction.

Tokenizer PHP Extension: Required for parsing PHP scripts, specifically tokenizing the parsed syntax.

XML PHP Extension: Required for processing and manipulating XML documents.

Fileinfo PHP Extension: Allows for detecting and working with files' various properties.

You also need to have the dependency manager, Composer, installed for managing Laravel packages and dependencies.

8. State some differences between Laravel and CodeIgniter.

[Hide Answer](#)

Laravel	CodeIgniter
Supports relational object oriented framework	Only supports object oriented framework
Provides support for custom HTTPS routes	Does not provide full support for HTTPS routes
Comes with authentication class features	Lacks authentication features
Provides robust testing framework including unit testing	No unit testing features in place
Uses blade templates	Does not support blade templates
Steeper learning curve	Easier to master
Easy REST API development	Inefficient REST API development process
Supports ORM	Doesn't support ORM out of the box.

Learn more about [Laravel vs CodeIgniter](#).

9. What is a Query builder?

[Hide Answer](#)

The query builder in Laravel provides direct access to the database while also acting as an alternative to Eloquent ORM. Developers do not require written SQL queries but rather offer a set of classes and methods. These can be used for building queries programmatically. It allows developers to easily and quickly build complex database queries while still maintaining the flexibility to use raw SQL when necessary. The query builder is a powerful tool for building efficient and maintainable database-driven applications in Laravel.

10. What kind of template is used by Laravel engine?

[Hide Answer](#)

Laravel uses the Blade templating engine for creating templates and rendering views. Blade is a powerful, flexible, and user-friendly templating system that comes built-in with the Laravel framework.

Blade offers features such as template inheritance, layout, and convenient control structures to help you write clean and maintainable template files. Blade templates have a .blade.php file extension and are stored in the resources/views directory.

11. What is the process of clearing cache in Laravel?

[Hide Answer](#)

Clearing cache in Laravel is essential for ensuring that your application uses the latest data and configurations. You can clear various cache types, such as configuration, views, routes, or application cache, using the built-in Artisan CLI commands. Here's a list of common cache-clearing commands:

Configuration Cache: To clear the configuration cache, which is created using php artisan config:cache, run:

```
php artisan config:clear
```

Routes Cache: To clear the routes cache generated using php artisan route:cache, execute:

```
php artisan route:clear
```

Compiled Views Cache: To clear the compiled views cache created using php artisan view:cache, run:

```
php artisan view:clear
```

Application Cache: To clear the application cache, which includes the cache created by using the Cache façade or the caching features provided by Laravel, execute:

```
php artisan cache:clear
```

Event Cache: To clear the cached events and listeners created using php artisan event:cache, run:

```
php artisan event:clear
```

12. Does Laravel support caching?

[Hide Answer](#)

Yes, Laravel supports caching and provides a unified API for various caching systems. Caching in Laravel is essential for improving the performance and speed of your application by storing the results of expensive operations or frequently accessed data, reducing the need for redundant computations or database queries.

Laravel's caching system is built on top of the Illuminate\Cache\CacheManager class and uses the Cache facade to interact with different cache drivers. The caching configuration can be found in the config/cache.php file, which allows you to define the default cache driver, lifetime, and configuration for supported cache systems.

13. Explain Middleware in Laravel.

[Hide Answer](#)

Middleware in Laravel is a mechanism that allows you to filter and manipulate HTTP requests and responses within the application's request lifecycle. Middleware acts as a bridge or layer between the initial HTTP request and the final response, enabling you to modify the request or response, or terminate the request entirely.

Common use cases for middleware include:

Authentication: Middleware is often used to handle user authentication, ensuring that the user is logged in and has the necessary privileges to access specific routes or resources within the application.

Authorization: Providing fine-grained access control, middleware can check if a user has specific roles or permissions before allowing them to perform certain operations or access specific routes.

CSRF Protection: Middleware helps protect your application against Cross-Site Request Forgery (CSRF) attacks by validating CSRF tokens.

14. What is the process of creating a route?

[Hide Answer](#)

In Laravel, routes are created using controllers or by adding codes directly into routes. The example below should help to understand the technical process involved:

Example: Replacing codes in routes/web.php file with the following codes.

```
<?php
use Illuminate\Support\Facades\Route;

Route::get('/', function () {
    return "Welcome!";
});
```

Once the updates are made, open a browser and run the project. You should see a 'Welcome!' message as an output.

You can also create routes in Laravel using controllers to handle requests and responses or define named routes for better organization and flexibility.

15. Mention the routing files in Laravel.

[Hide Answer](#)

In Laravel, routing files are used to define the various types of routes that cater to different aspects of your application. These routing files are stored within the routes directory of your Laravel project. The default routing files include:

web.php: This routing file is primarily used for web routes that require functionalities such as sessions, cookies, and CSRF protection. Routes defined in this file are assigned the web middleware group by default.

api.php: This routing file serves the purpose of defining routes for APIs or stateless routes that expect token-based authentication and JSON-based responses. Routes in this file are assigned the api middleware group by default, which includes rate limiting and stateless behavior.

console.php: This routing file is intended for defining custom Artisan console commands that help automate various tasks within your Laravel application.

channels.php: This file is used for defining and configuring real-time broadcasting channels for your Laravel application, which handle WebSocket-based communication and channel authorization.

16. What are Facades in Laravel?

[Hide Answer](#)

Facades in Laravel are a convenient and elegant way to access various services and components provided by the framework. They act as a static interface to the underlying classes available in Laravel's IoC (Inversion of Control) container. Facades provide a simple, easy-to-use syntax for using Laravel's features without the need to access an instance of a class directly.

17. How to find and identify blade templates?

[Hide Answer](#)

In Laravel, Blade templates are used for creating views and defining the structure and design of your application's user interface. To find and identify Blade templates, follow these steps:

Locate the Blade templates: Blade template files are stored in the resources/views directory within your Laravel project. This directory contains template files organized into subdirectories according to the application's structure and requirements.

Look for the file extension: Blade templates have a specific file extension, .blade.php, which distinguishes them from regular PHP files.

Inspect the Blade syntax: Blade templates use a unique syntax for variables, control structures, template inheritance, and other features. Blade directives such as {{ \$variable }}, @if, @foreach, @extends, and @section indicate that the file is a Blade template.

18. Mention the loops provided by the Blade templating engine.

[Hide Answer](#)

The Blade templating engine in Laravel provides several looping constructs that make it easy to iterate through data or repeat code snippets within your templates. Some common loops provided by Blade include:

@for: The @for loop executes a block of code a specified number of times. Usage in Blade is similar to a standard PHP for loop.

@foreach: The @foreach loop is used to iterate through an array or collection, executing the loop body for each item. The Blade @foreach loop resembles the PHP foreach construct.

```
@foreach ($users as $user)
    User: {{ $user->name }}
@endforeach
```

@while: The @while loop continues executing the loop body as long as the specified condition remains true. It resembles the PHP while loop.

```
@php($count = 0)
@while($count < 5)
    Count: {{ $count }}
    @php($count++)
@endwhile
```


@forelse: The @forelse loop is an enhanced version of @foreach that allows you to specify a default block of code to execute if the array or collection being iterated is empty.

```
@forelse ($users as $user)
    User: {{ $user->name }}
@empty
    No users found.
@endforelse
```

Loop Variables: Blade also exposes a \$loop variable when using loops like @foreach and @forelse, which provides useful information about the current iteration, such as index, iteration, remaining, count, first, last, even, and odd. This allows you to handle different scenarios within your loop more effectively.

```
@foreach ($items as $item)
    @if ($loop->first)
        This is the first item.
    @endif
    Item: {{ $item }}
    @if ($loop->last)
        This is the last item.
    @endif
@endforeach
```

19. Explain dd() function in Laravel.

[Hide Answer](#)

The dd() function (short for "dump and die") is a helpful debugging tool in Laravel for quickly inspecting variables, objects, or any expression's value during the development process. The dd() function outputs the value of the given expression in an organized, human-readable format and immediately halts the further execution of the script.

The dd() function leverages the symfony/var-dumper package to provide more comprehensive output compared to a simple var_dump or print_r. It renders the

output with syntax highlighting and collapsible structures, making it easier to explore and visualize complex objects or arrays.

Here's an example of using `dd()` with an array:

```
$array = [  
    'name' => 'John',  
    'occupation' => 'Developer',  
];  
  
dd($array);
```

When the `dd()` function is called, it will dump the array's content and stop the script from executing any further. This function can also be used within Blade templates or controller methods for debugging during the application development process.

20. What is PHP artisan? Explain some common artisan commands.

[Hide Answer](#)

The PHP artisan is a CLI/tool that comes preloaded in Laravel and offers several useful commands. These can assist to build applications faster and with more efficiency. Some of the most common examples of artisan commands include the likes of:

PHP artisan list: Used to view the entire list of available artisan commands.

PHP artisan help: Used to display and describe commands available arguments and options.

PHP artisan tinker: Used for writing write actual PHP code using the command line.

PHP artisan-version: Used to view the current version of Laravel.

PHP artisan make model model_name: Used for creating models 'model_name.php' under the 'app' directory.

PHP artisan make controller controller_name: Used to build new controller files in app/Http/Controllers folder.

21. What is the process of using custom table in Laravel?

[Hide Answer](#)

Using a custom table in Laravel primarily involves creating a new Eloquent model and defining the custom table name within the model. Here is a step-by-step guide to using a custom table in Laravel:

Create a new Eloquent model: Generate a new Eloquent model using the Artisan command:

```
php artisan make:model CustomTable -m
```

This command creates a new model file named CustomTable.php in the app/Models directory (or app directory for Laravel 7 or older), and the -m flag generates an associated migration file in the database/migrations folder.

Define the custom table name in the model: Open the CustomTable.php model file and set the protected \$table property to your custom table name:

```
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class CustomTable extends Model
{
    use HasFactory;

    protected $table = 'custom_table_name'; // Specify your custom table name
}
```

Modify the migration file: In the database/migrations folder, edit the migration file generated for your custom table and define the table schema as per your requirements:

```

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateTableExample extends Migration
{
    public function up()
    {
        Schema::create('custom_table_name', function (Blueprint $table) {
            $table->id();
            // Define other columns as needed
            $table->timestamps();
        });
    }

    public function down()
    {
        Schema::dropIfExists('custom_table_name');
    }
}

```

Run migration: Execute the migration to create the custom table in your database:

```
php artisan migrate
```

Perform operations using the custom table: You can now perform CRUD operations, relationships, and queries using the Eloquent model for your custom table.

22. What is the process of creating a helper file in Laravel?

[Hide Answer](#)

Helper files can be created by simply following the steps using the composer.

First, create a new file "app/helpers.php" within the app folder.

Then define your helper functions inside the helpers.php file.

Next, you need to autoload the helpers.php file in your Laravel project. To do this, open the composer.json file located in the root directory of your Laravel project and add the following code to the autoload section:

```
"files": [  
    "app/helpers.php"  
]
```

23. Explain the process of implementing a package in Laravel.

[Hide Answer](#)

Implementing a package in Laravel is pretty simple and can be completed following the mentioned process below:

- Create a new package folder and name it
 - Create a composer.json file for the same package
 - Load the package using the main composer.json and PSR-4
 - Make a new Service Provider
 - Set up a new Controller for the package
 - Make a Routes.php file and you are done
-

24. What is the process to check logged-in user info in Laravel?

[Hide Answer](#)

In Laravel, you can use the Auth facade to check logged-in user information and perform various tasks related to authentication. To access the information of the currently logged-in user, utilize the user() method provided by the Auth facade.

Here's an example of how to fetch the information for the logged-in user:

```
use Illuminate\Support\Facades\Auth;

// Access logged-in user information
$user = Auth::user();

// Access individual attributes of the user
$user_id = Auth::id();
$user_name = $user->name;
$user_email = $user->email;
```

The `Auth::user()` method returns an instance of the authenticated user model, which allows you to access the user's attributes directly like you would with any Eloquent model.

25. What is Guarded Attribute in a Laravel model?

[Hide Answer](#)

Guarded attributes work as the opposite of fillable attributes. These attributes are used for specifying fields that are not mass assignable.

Example:

```
class User extends Model {
protected $guarded = ['role'];
// All fields inside the $guarded array are not mass-assignable
}
```

26. What are Closures in Laravel?

[Hide Answer](#)

Closures are anonymous functions that are not a part of any object or class. They are often used as a callback function. Closures can be used to modify data, filter results, or perform any other operations that can be expressed as a function. These can also be used as a parameter in a function allowing passing parameters into a Closure. The

changes can be done by just altering the function of the handle() method for providing parameters to it. A Closure can also access relevant variables outside the scope of the variable.

27. Explain factories in Laravel.

[Hide Answer](#)

Factories in Laravel are an essential tool that enables the creation of fake or sample data for your application's models, which can be helpful during development, testing, or even seeding data. Factories allow you to generate model instances with random or pre-defined values, ensuring that you create consistent and realistic sets of data for various purposes.

Laravel model factories use the powerful Faker library for generating random sample data. Factories are defined as classes that extend the “Illuminate\Database\Eloquent\Factories\Factory” class and are typically stored in the “database/factories” directory of your Laravel project.

28. What are some of the Aggregates methods provided by query builder in Laravel?

[Hide Answer](#)

The aggregates functions are used where values from more than one row are grouped together as input. These are often based on a single criterion and form a single value for a specific meaning or measurement.

Some of the aggregates methods provided by the query builder in Laravel include:

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

- min()
-

29. What is the process for enabling query log in Laravel?

[Hide Answer](#)

The steps mentioned below should help in enabling query log in Laravel:

- `DB::connection()->enableQueryLog();`
 - Post query, place it
 - `$querieslog = DB::getQueryLog();`
 - Then, place it
 - `dd($querieslog)`
-

30. What is Dependency injection in Laravel?

[Hide Answer](#)

Dependency injection in Laravel is a design pattern that promotes loose coupling of components and helps manage their dependencies efficiently. It involves providing the required dependencies to an object, rather than creating them within the object itself. By doing so, it becomes easier to test, maintain and update the code.

In Laravel, the dependency injection mechanism is primarily facilitated by the Service Container, also referred to as the IoC (Inversion of Control) container. It is a powerful tool for managing class dependencies, which can automatically resolve and inject dependencies by analyzing their constructor signatures. This reduces the need for manual instantiation and promotes adherence to the SOLID design principles.

Example:


```
public function __construct(UserRepository $data)

{

    $this->userdata = $data;

}
```

31. What is the process for using skip() and take() in Laravel Query?

[Hide Answer](#)

In Laravel, the skip() and take() methods are used to paginate and limit the results of database queries, providing an effective way to handle large datasets by fetching specific chunks of data.

skip() method is used to skip a certain number of records from the beginning of the resultset, while take() method is used to limit the number of records returned. By using these methods in combination, you can easily implement custom pagination for your queries.

Here's an example of using skip() and take() with an Eloquent query:

```
use App\Models\User;

$page = 1; // Assume this is the current page number
$perPage = 10; // Records per page to be fetched

$users = User::query()
    ->orderBy('id') // Or any other sort order you'd like to impose
    ->skip(($page - 1) * $perPage)
    ->take($perPage)
    ->get();
```

In this example, we fetch 10 records per page, starting from the first record of the current page. By changing the \$page variable, you can paginate through the records and fetch the desired chunk of data

32. Explain the repository pattern in Laravel.

[Hide Answer](#)

The repository pattern works to decouple data access layers and business logic in an application. The pattern enables objects without knowing the persistence of objects. The business logic does not require an understanding of how data is being retrieved because it relies on repositories to fetch the correct data.

33. What are the advantages of Queue?

[Hide Answer](#)

Queues in Laravel is one of the best approaches for handling tedious and lengthy processes. These allow developers to offload work from the web server, minimizing the waiting period for a response from APIs before loading the next page.

These are extremely handy if an application is using multiple servers which will be in use at the same time, having jobs interfere any internal processes.

34. Explain responses in Laravel.

[Hide Answer](#)

In Laravel, responses are used to send the output of an application back to the client (e.g., a browser or API client) as an HTTP response. Responses can contain different types of data, like HTML, JSON, plain text, or files, depending on the requirements of the application. Laravel offers various ways to create and manipulate responses, allowing developers to customize and fine-tune the data being sent to the client.

35. What is a REPL?

[Hide Answer](#)

A Read-Eval-Print Loop (REPL) is an interactive programming environment that enables developers to input code, evaluate it, and receive immediate output or feedback. REPL allows for easy experimentation, quick prototyping, and learning language syntax and features. It dynamically executes code without the need to compile or write a complete program, which helps developers to test small code snippets or debug their code in real-time.

36. What is the process of stopping Artisan service in Laravel?

[Hide Answer](#)

If developers are facing any kind of problem with artisan service in Laravel, the following steps should help to terminate the service.

Start by pressing Ctrl + Shift + ESC to call up the Windows task manager. Look for the PHP system walking artisan process and end the process tree. Then, reopen the command line and restart the server.

One can also skip using the task manager and try to kill the PHP process by pressing Ctrl+C in the command line.

37. Explain yield in Laravel.

[Hide Answer](#)

In Laravel, yield is a Blade directive used within templates to define placeholders or sections for content that will be injected later by the child templates or views. It is a key part of Blade's template inheritance system, allowing you to create master layouts with specific content sections that can be filled in by the child templates.

The yield directive is used primarily in layout files or master templates to specify where the content from the child templates will be injected.

38. What is the process of hashing passwords in Laravel?

[Hide Answer](#)

In Laravel, hashing passwords is a crucial security practice used to store passwords securely in the database. Laravel uses the Bcrypt hashing algorithm and, more recently, the Argon2 algorithm (introduced in PHP 7.2) for password hashing. To hash passwords, Laravel provides a helpful wrapper around these algorithms using the Hash facade.

Here's a simple example of how to hash a password in Laravel:

```
use Illuminate\Support\Facades\Hash;

$password = 'plain_text_password';
$hashed_password = Hash::make($password);
```

In this example, the plain text password is passed to the Hash::make method, which creates a hashed password using the configured hashing algorithm.

39. Explain Laravel Vapor.

[Hide Answer](#)

Laravel Vapor is a serverless deployment platform built specifically for Laravel applications and powered by AWS Lambda. Laravel Vapor offers a fully managed, scalable, and reliable environment to deploy and manage Laravel applications without the need to manage servers or infrastructure.

Vapor takes care of the underlying server management, scaling, and deployment, allowing you to focus on building your application's features and functionality. Laravel Vapor integrates seamlessly with popular AWS services, such as RDS, S3, and SQS, to provide a comprehensive ecosystem for supporting Laravel applications.

40. Explain collections in Laravel.

[Hide Answer](#)

Collections in Laravel are a powerful and convenient programming tool that provides a convenient way to work with arrays and data manipulation. Laravel Collections use a fluent, object-oriented interface, which makes it easy to chain methods and transform datasets in an expressive and efficient manner.

Collections in Laravel are essentially a wrapper around PHP arrays, offering additional functionality for mapping, filtering, sorting, and reducing data, making array manipulation more intuitive and readable.

41. Explain the function of Console Kernel.

[Hide Answer](#)

In Laravel, the Console Kernel is responsible for handling and managing the Artisan command-line interface (CLI) and scheduled tasks. It is a key component of Laravel's command handling and task scheduling system.

The Console Kernel is a class named Kernel, which is located in the app/Console directory and extends the Illuminate\Foundation\Console\Kernel class. The Console Kernel is responsible for the following tasks:

- **Registering Artisan Commands:** In the commands property of the Console Kernel, you'll register all custom Artisan commands that your application needs. By doing this, the commands become available for use when invoking Artisan.
 - **Scheduling Tasks:** The Console Kernel is responsible for scheduling tasks that need to run periodically, such as data imports, database cleanups, or sending email notifications. You'll define the scheduled tasks within the schedule method of the Kernel, utilizing Laravel's powerful and expressive task scheduler.
-

42. What is Nova?

[Hide Answer](#)

Laravel Nova is an administration panel for Laravel applications, developed and maintained by the Laravel team. Nova is a beautifully designed, highly customizable, and powerful admin dashboard that allows you to manage your application's data and resources with minimal effort.

Nova is designed to work seamlessly with your existing Laravel application, using your existing Eloquent models and relationships to generate a complete administration panel without writing any additional code.

Some key features of Laravel Nova include:

Resource Management: Nova automatically generates admin panels for managing Eloquent models and their relationships, enabling you to create, read, update, and delete records directly from the panel.

Actions: Perform custom actions on model resources through the admin panel, such as bulk updates, exporting data, or running admin-specific tasks.

Filters: Create custom filters to narrow down the display of model records in the admin panel based on specific conditions or attributes.

Metrics: Display various data metrics, such as value, trend, and partition charts, on the dashboard to get insights into your application's data.

Lenses: Use lenses to create custom data views for your resources, allowing you to display and query data differently than in the default resource views.

Customization: Easily extend and customize the appearance and functionality of the admin panel, making it adaptable to your project's specific requirements.

Authorization: Integrates with Laravel's built-in policy system to secure your admin panel, providing fine-grained access control for different user roles.

43. Explain how the process of request validation is processed in Laravel.

[Hide Answer](#)

Developers can choose either to create a request validation class or a controller method for requesting validation. You can check the below example using the controller method for a better idea.

Example

```
public function store (Request $request)

{

    $validated = $request->validate([ 'title' => 'required|unique:posts|max"255', 'body' =>
    'required',]);

}
```

44. What is register and boot methods in the service provider class?

[Hide Answer](#)

In Laravel, the register and boot methods are essential methods within a service provider class that are used to define and customize the service registration and initialization process.

register method: The primary purpose of the register method is to bind services, classes, or components into the application's service container. Within the register method, you can define the bindings, resolve dependencies, instantiate classes, or register services. It's important to note that you should not attempt to access or use other services within this method, as there is no guarantee that they have been loaded into the container yet.

```
public function register()
{
    $this->app->singleton(SomeService::class, function ($app) {
        return new SomeService($app->make('someDependency'));
    });
}
```

boot method: The boot method is called after all other service providers have been registered, making it the right place to perform actions that depend on other services being registered in the container. You can perform any necessary configuration, customization, or setup tasks in this method, such as registering event listeners, defining view composers, or publishing configuration files.

```
public function boot()
{
    view()->composer('view_name', ViewComposer::class);
}
```

45. What are accessors and mutators in Laravel?

[Hide Answer](#)

In Laravel, accessors are used for changing data after they are fetched from the database. For example, you may want to format a date or concatenate two fields together.

Whereas mutators are used for modifying data before saving it in the database. For example, you may want to encrypt a password or format a date.

Using accessors and mutators in your Laravel application can help you keep your code organized and make it easier to work with your models.

46. Explain throttling and how to implement it in Laravel.

[Hide Answer](#)

In Laravel, throttling is a perfect approach for rate-limiting requests from specific IPs and is also capable enough to prevent DDOS attacks. The framework also provides a middleware that is compatible with not just routes but global middleware as well. Developers can configure throttling following the steps.

You can implement throttling as below:


```
Route::middleware('auth:api', 'throttle:60,1')->group(function () {  
  
    Route::get('/user', function () {  
  
        //  
  
    });  
  
});
```

In this example, the throttle middleware is being applied to the /user endpoint and is set to allow 60 requests per minute (60,1). This means that if a client makes more than 60 requests to this endpoint within a minute, they will be blocked for a period of time before being allowed to make further requests.

47. What is logging in Laravel?

[Hide Answer](#)

Logging in Laravel is often used to track down bugs but at times can be a bit tricky to use. It also helps to understand log messages using a user-friendly logging system. Developers can also write log messages to files, system logs, Slack, etc.

Moreover, logging in Laravel is based on channels where every different unique channel represents a different way of writing log information.

48. What is the process of extending login time in Auth?

[Hide Answer](#)

Extending the login expiration time on Laravel is pretty easy and can be done with the config/session.php. Developers only need to update the lifetime value mentioned in the variable. The value of the variable usually is set to 120 by default, which can be altered based on requirements.

49. Explain the process of making a constant for global use.

[Hide Answer](#)

Developers can create constant.php pages directly in the config folder if not available already. Enter a constant variable with a corresponding value and use the command

```
Config::get('constants.VariableName');
```

Here's an example for better understanding

Example

```
return [  
    'ADMINEMAIL' => 'info@bestinterviewquestion.com',  
];  
Now we can display with  
Config::get('constants.ADMINEMAIL');
```

Looking for remote developer job at US companies?

Work at Fortune 500 companies and fast-scaling startups from the comfort of your home

[Apply Now](#)

ADVANCED LARAVEL DEVELOPER INTERVIEW QUESTIONS AND ANSWERS

1. Explain the process of disabling CSRF protection on specific routes.

[Hide Answer](#)

In Laravel, the CSRF (Cross-Site Request Forgery) protection middleware is enabled by default, but there may be cases where you need to disable it for specific routes or URLs.

To disable CSRF protection for specific routes, developers can add the URL or route to the '\$except' variable. The variable is readily available from the path `app\Http\Middleware\VerifyCsrfToken.php` file. Check out the example below to get a better understanding of the same.

Example

```
class VerifyCsrfToken extends BaseVerifier {  
    protected $except = [  
        'Pass here your URL',  
    ];  
}
```

2. Explain Service Containers.

[Hide Answer](#)

Service containers are exceptionally powerful tools that can help to manage dependencies over classes and can also assist in dependency injections. Service containers also offer different advantages like.

- Easy class dependency management for creating objects
 - Services contained as a registry
 - Allows binding of interfaces to concrete classes
-

3. How are delete() and softDeletes() different?

[Hide Answer](#)

`delete()` is a method in Laravel's Query Builder that deletes one or more records from a database table permanently. Whereas using the `delete()` command removes the target entries permanently, the `softDeletes()` is a feature in Laravel that provides a way to "soft delete" records instead of permanently deleting them. Soft deleting is a mechanism for flagging records as deleted instead of physically deleting them from the database.

Example

delete()	softDeletes()
<code>\$delete = Post::where('id', '=', 1)->delete();</code>	<code>use SoftDeletes;</code> <code>protected \$dates = ['deleted_at'];</code> then , both cases are usable. <code>\$softDelete = Post::where('id', '=', 1)->delete();</code>

4. Mention the process of using cookies in Laravel.

[Hide Answer](#)

In Laravel, cookies can be used to store small amounts of data on the client-side and retrieve them at a later time. Laravel supports handling and managing cookies via the `Illuminate\Http\Request` and `Illuminate\Http\Response` objects. To work with cookies in Laravel, follow these steps:

Creating Cookies: To create a cookie, use the cookie helper function or the Cookie facade. This generates a new `Illuminate\Cookie\CookieJar` instance representing the cookie, with options such as name, value, duration, path, domain, secure, and HTTP only.

```
// Using the cookie helper function
$cookie = cookie('cookie_name', 'cookie_value', $minutes);

// Using the Cookie facade
use Illuminate\Support\Facades\Cookie;
$cookie = Cookie::make('cookie_name', 'cookie_value', $minutes);
```

Attaching Cookies to Responses: To send the created cookie to the client, attach it to your response object using the `withCookie` method.

```
use Illuminate\Http\Response;

// In your controller or route closure
public function handleRequest(Request $request)
{
    $cookie = cookie('cookie_name', 'cookie_value', $minutes);
    return (new Response('Hello'))->withCookie($cookie);
}
```

Retrieving Cookies: To access the values of cookies sent by the client, use the cookie method on the Illuminate\Http\Request object.

```
// In a controller or route closure
public function handleRequest(Request $request)
{
    $cookieValue = $request->cookie('cookie_name');
}
```

Encryption: By default, Laravel encrypts and signs all cookies, ensuring data confidentiality and integrity. If you need to set a cookie that should not be encrypted, add the cookie's name to the except array in the config/cookie.php configuration file.

5. What is with() function in Laravel?

[Hide Answer](#)

The with() function in Laravel is used to eager load relationships for the retrieved models. It's typically used when retrieving models that have relationships with other models to avoid the N+1 query problem. The with() function can be used after the first command if data is being fetched from the database using only a single query. The function can help to improve the user experience as it minimizes the number of queries required to fetch results.

6. Is it possible to run a Laravel project in xampp?

[Hide Answer](#)

Yes, xampp can help to run Laravel projects. Once you have xampp installed in the system folder, follow these steps:

- Make a new Laravel folder in htdocs under the xampp
 - Use the command prompt for redirecting to Laravel and use the below mentioned command:
composer create-project laravel/laravel first-project --prefer-dist
Redirect to localhost/laravel/first-project/public/.
-

7. How to use insert statement function in Laravel?

[Hide Answer](#)

In Laravel, you can use the Query Builder or Eloquent ORM to execute an INSERT statement and insert a new record into your database. Here is how you can do it using both methods:

Query Builder: To insert data using Laravel's Query Builder, you can use the insert method on the DB facade:

```
use Illuminate\Support\Facades\DB;

DB::table('your_table')->insert([
    'column1' => 'value1',
    'column2' => 'value2',
    // ...
]);
```

Replace 'your_table' with the name of the table you want to insert data into, and provide an array containing your column names as keys and the corresponding values to insert.

Eloquent ORM: To insert a new record using Eloquent ORM, create a new model instance, set the desired attributes, and call the save method:

```
use App\Models\YourModel;  
  
$record = new YourModel;  
$record->column1 = 'value1';  
$record->column2 = 'value2';  
// ...  
$record->save();
```

Replace YourModel with the name of the Eloquent model representing the table you want to insert data into, and set the corresponding attributes with the values you want to insert.

Both approaches allow you to use Laravel's built-in functionality to insert new records into the database easily and efficiently.

8. How to rollback a migration?

[Hide Answer](#)

Developers can rollback a migration, by accessing the migrations table. Every migration table comes with a unique batch number and helps to roll back the last batch. You can use the command 'php artisan migrate:rollback --step=1' to rollback the one batch of migration

9. How to run test cases?

[Hide Answer](#)

You can run test cases in Laravel using the PHPUnit or even the artisan test command. Check out the example below to get a better understanding.

Example

```

namespace Tests\Unit;
use PHPUnit\Framework\TestCase;
class ExampleTest extends TestCase
{
    * @return void
    public function testBasicTest()
    {
        $this->assertTrue(true);
    }
}

```

To run this test, you can execute the phpunit command in the terminal from the root directory of your Laravel application. PHPUnit will automatically detect and run all the tests in the tests/ directory. You can also run a specific test file or a specific test method using the --filter option.

10. How to use the updateOrCreate() method in Laravel Query?

[Hide Answer](#)

Developers use the 'updateOrCreate()' function for updating existing records in the database for matching conditions or creating one if there is no existing matching record. The return type is usually Boolean.

Syntax

DB::table('blogs')->updateOrCreate([Conditions],[fields with value]);

Example

```

DB::table('blogs')->updateOrCreate(
    ['email' => 'info@bestinterviewquestion.com', 'title' => 'Best Interview Questions'],
    ['content' => 'Test Content']
);

```

11. How to check if a column exists or not in a table?

[Hide Answer](#)

To check whether a column exists in a table, you can use the following command

```
if(Schema::hasColumn('admin', 'username')) //check whether admin table has  
username column  
{  
    // write your logic here  
}
```

12. Explain what are gates in Laravel?

[Hide Answer](#)

Gates in Laravel offer a high-end mechanism that helps to authorize user activities using available resources. As the implementation of models are not defined by gates, it offers users the freedom of writing different types of use cases based on choice. Gates allow you to define granular permissions for specific actions or resources within your application. Gates can be used to check if a user is authorized to perform a particular action, such as viewing a specific page or editing a particular resource.

Gates are defined as callback functions that return a boolean value indicating whether or not the user is authorized to perform the requested action. Gates can be defined in a service provider or in a gate provider class, and can be used throughout your application to check user permissions.

13. What is forge in Laravel?

[Hide Answer](#)

Laravel Forge is a server management and deployment platform designed specifically to streamline the deployment and hosting of Laravel applications. Forge simplifies the provisioning, management, and monitoring of servers, enabling you to focus on your application's features and functionality rather than server configuration and maintenance tasks.

Forge provides an intuitive interface for deploying and managing Laravel applications on popular Infrastructure as a Service (IaaS) providers like AWS, DigitalOcean, Linode, or custom VPS providers.

14. How is redirecting from the controller to the view file done?

[Hide Answer](#)

Anybody can redirect from controllers to view using the following commands:

- `return redirect('/')->withErrors('You can type your message here');`
 - `return redirect('/')->with('variableName','You can type your message here');`
 - `return redirect('/')->route('PutRouteNameHere');`
-

15. How can you use Laravel's container to bind an interface to a concrete implementation?

[Hide Answer](#)

In Laravel, you can bind an interface to a concrete implementation using the Service Container's bind method. This enables dependency injection and makes the application more extensible and testable. Here's an example:

```
// In a service provider's boot or register method.  
$this->app->bind(\App\Contracts\PaymentGateway::class,  
    \App\Services\StripePaymentGateway::class);
```

16. Explain how to create and dispatch jobs in Laravel with delayed execution.

[Hide Answer](#)

To create a job in Laravel, first, run `php artisan make:job ProcessTask`. This command will generate a Job class in `app/Jobs/`. Edit the handle method to include the job's

logic. To dispatch a job with delayed execution, use the dispatch function with the delay method. For example:

```
$job = new App\Jobs\ProcessTask($data);  
$delay = now()->addMinutes(10);  
dispatch($job)->delay($delay);
```

17. How do you create a middleware in Laravel that checks for a specific HTTP header?

[Hide Answer](#)

To create a middleware in Laravel, first, run the command `php artisan make:middleware CheckHttpHeader`. This will generate a middleware class in `app/Http/Middleware`. Edit the `handle` method with the HTTP header check logic:

```
    public function handle(Request $request, Closure $next)  
{  
    if (!$request->hasHeader('X-Specific-Header')) {  
        return response('Forbidden. X-Specific-Header is missing.', 403);  
    }  
  
    return $next($request);  
}
```

Then, register the middleware in the `app/Http/Kernel.php` file and use it in your routes.

18. Explain how to define two different authentication guard systems in Laravel.

[Hide Answer](#)

To define two different authentication guards, you should edit the `config/auth.php` configuration file. Add new guards and providers specific to the different

authentication methods. For example:

```
// In config/auth.php
'guards' => [
    'web' => [
        'driver' => 'session',
        'provider' => 'users',
    ],
    'api' => [
        'driver' => 'token',
        'provider' => 'users',
        'hash' => false,
    ],
    'admin' => [
        'driver' => 'session',
        'provider' => 'admins',
    ],
],
'providers' => [
    'users' => [
        'driver' => 'eloquent',
        'model' => App\Models\User::class,
    ],
    'admins' => [
        'driver' => 'eloquent',
        'model' => App\Models\Admin::class,
    ],
],
```

19. Explain how to implement rate limiting for certain routes in Laravel.

[Hide Answer](#)

To implement rate limiting in Laravel, edit the app/Http/Kernel.php file to add ThrottleRequests middleware to the \$routeMiddleware array:

```
protected $routeMiddleware = [  
    // ...  
    'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,  
];
```

Then, apply the throttle middleware to the desired routes in your route definition files, specifying the rate limit and the time interval:

```
Route::middleware(['throttle:60,1'])->group(function () {  
    Route::get('/rate-limited-route', 'Controller@action');  
});
```

Looking for remote developer job at US companies?

Work at Fortune 500 companies and fast-scaling startups from the comfort of your home

Apply Now