Laravel Developer Interview Questions and Answers

1. What are service providers in Laravel? How are they used?

Service providers are the central place of all Laravel application bootstrapping. They bind services into the service container and are used to register application services, event listeners, and more.

2. Explain the lifecycle of a Laravel request.

The Laravel request lifecycle includes the following steps: (1) HTTP request hits public/index.php. (2) Autoload files are loaded via Composer. (3) Kernel handles the request by bootstrapping services, processing middleware, and routing. (4) The response is sent back to the browser.

3. What is middleware, and how do you create and apply custom middleware in Laravel?

Middleware acts as a bridge between a request and a response, performing actions like authentication. To create custom middleware, use 'php artisan make:middleware MiddlewareName'. Register it in 'app/Http/Kernel.php' and apply it in routes or controllers.

4. What is Eloquent ORM, and how does it differ from other ORMs?

Eloquent ORM is Laravel's default ORM, providing a simple ActiveRecord implementation. It simplifies database operations using models and relationships. Unlike other ORMs, Eloquent provides an expressive, chainable query syntax.

5. How do you handle database migrations in Laravel?

Migrations are version control for your database. Use 'php artisan make:migration' to create migrations and 'php artisan migrate' to apply them. They help in syncing database structure across environments.

6. What are Policies and Gates in Laravel? When would you use them?

Policies and Gates provide authorization logic for actions. Gates are simple closures, while Policies organize logic for a specific model. Use them to restrict user actions based on roles or permissions.

7. What are Observers in Laravel, and how are they used?

Observers are classes that group event handlers for a model. Use 'php artisan make:observer ObserverName' to create one. They listen to events like creating, updating, or deleting a model and perform related tasks.

8. How does Laravel handle dependency injection?

Laravel's service container resolves dependencies automatically. Simply type-hint a class or interface in a constructor or method, and Laravel injects the appropriate dependency.

9. What is CSRF protection, and how is it implemented in Laravel?

CSRF (Cross-Site Request Forgery) protection prevents unauthorized commands from being performed by a malicious actor. Laravel includes a CSRF token in forms and validates it on submission.

10. How would you optimize database queries in Laravel?

Optimization techniques include using Eloquent relationships effectively, eager loading to avoid N+1 query problems, indexing database columns, caching results, and using query scopes.