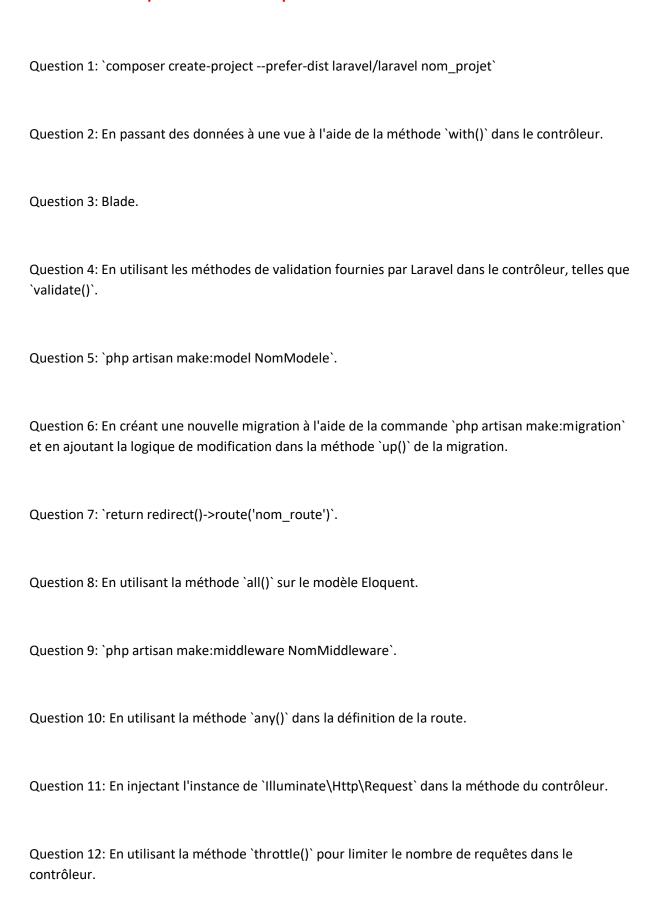
Voici les réponses aux questions du QCM sur Laravel :



Question 13: En utilisant la classe 'Mail' de Laravel pour envoyer un email. Question 14: `php artisan serve`. Question 15: En utilisant la méthode 'request()' dans le contrôleur ou en utilisant la méthode `request()` de la facade `Request`. Question 16: En utilisant la méthode `foreign()` dans la méthode `up()` de la migration. Question 17: En utilisant la planification de tâches cron dans le fichier `app/Console/Kernel.php`. Question 18: En utilisant la méthode `validate()` dans le contrôleur et en utilisant les redirections automatiques en cas d'erreur. Question 19: En utilisant la méthode 'hasMany()' dans le modèle parent. Question 20: En utilisant la méthode `load()` sur le modèle avec une condition. Question 21: En définissant une méthode 'getNomAttributAttribute()' dans le modèle. Question 22: Tinker est une interface en ligne de commande de Laravel qui permet d'interagir avec l'application et de manipuler les données. Question 23: En publiant les assets du package à l'aide de la commande `php artisan vendor:publish`. Question 24: En utilisant la commande 'php artisan key:generate'. Question 25: `php artisan make:event NomEvenement && php artisan make:listener NomListener`.

Question 26: En implémentant une classe qui implémente l'interface `Illuminate\Contracts\Cache\Store`. Question 27: `config/services.php`. Question 28: En utilisant les directives `@slot` et `@component` dans les vues Blade. Question 29: Un trait est une manière de réutiliser du code dans plusieurs classes de manière horizontale. On peut l'utiliser en ajoutant le trait dans la classe avec le mot-clé `use`. Question 30: En définissant une méthode 'getNomAttributAttribute()' dans le modèle. Question 31: En utilisant les méthodes 'beginTransaction()', 'commit()' et 'rollback()' sur l'instance de la base de données. Question 32: En utilisant l'option `encrypt` dans le modèle Eloquent ou en utilisant le mutateur `setAttribute`. Question 33: En utilisant Laravel Mix dans le fichier 'webpack.mix.js' et en exécutant la commande 'npm run dev'. Question 34: En retournant un tableau ou un objet JSON à l'aide de la méthode 'json()' dans le contrôleur. Question 35: En créant des écouteurs d'événements et en les enregistrant dans le fichier

Question 36: En définissant les politiques dans le dossier `app/Policies` et en les enregistrant dans le

`app/Providers/EventServiceProvider.php`.

fichier `app/Providers/AuthServiceProvider.php`.

Question 37: En configurant les files d'attente dans le fichier `config/queue.php`.

Question 38: En créant un événement personnalisé à l'aide de la commande `php artisan make:event` et en le diffusant.

Question 39: En étendant la classe du service provider existant et en ajoutant les fonctionnalités supplémentaires dans la méthode `register()` ou `boot()`.

Question 40: En utilisant Composer pour installer des packages Laravel et en les intégrant dans l'application en fonction des besoins.