

Question 1 : Quelle commande crée un nouveau projet Laravel ?

- Réponse : **composer create-project --prefer-dist laravel/laravel nom_projet**

Question 2 : Comment passer des données à une vue ?

- Réponse : En utilisant la méthode **with()** ou en passant un tableau associatif à la méthode **view()**

Question 3 : Quel est le nom du moteur de template utilisé par Laravel ?

- Réponse : Blade

Question 4 : Comment peut-on valider les données de requête dans un contrôleur ?

- Réponse : En utilisant la méthode **validate()** sur l'objet Request dans le contrôleur

Question 5 : Quelle commande utilise-t-on pour créer un modèle ?

- Réponse : **php artisan make:model NomModele**

Question 6 : Comment ajoute-t-on une colonne à une table existante via une migration ?

- Réponse : En modifiant le fichier de migration correspondant puis en exécutant **php artisan migrate**

Question 7 : Quelle méthode utilise-t-on pour faire une redirection vers une autre route ?

- Réponse : **return redirect()->route('nom_de_la_route');**

Question 8 : Comment récupère-t-on toutes les données d'une table avec Eloquent ?

- Réponse : En utilisant la méthode **all()** sur le modèle Eloquent correspondant

Question 9 : Quelle commande crée un middleware ?

- Réponse : **php artisan make:middleware NomMiddleware**

Question 10 : Comment définit-on une route qui accepte n'importe quelle méthode HTTP ?

- Réponse : En utilisant **Route::any('url', 'Controller@methode');**

Question 11 : Comment accède-t-on à l'instance de la requête actuelle dans un contrôleur ?

- Réponse : En l'injectant dans la méthode du contrôleur ou en utilisant la méthode statique **request()**

Question 12 : Quelle méthode permet de limiter le nombre de requêtes qu'un utilisateur peut faire ?

- Réponse : En utilisant la fonction **throttle()** dans les routes

Question 13 : Comment envoie-t-on un email en utilisant Laravel ?

- Réponse : En utilisant la classe **Mail** de Laravel et en créant un Mailable

Question 14 : Quelle commande permet de lancer le serveur de développement de Laravel ?

- Réponse : **php artisan serve**

Question 15 : Comment accède-t-on aux paramètres d'URL dans une route ?

- Réponse : En utilisant la méthode **request()** ou en injectant **Request** dans la méthode de la route

Question 16 : Comment fait-on pour ajouter une contrainte de clé étrangère dans une migration ?

- Réponse : En utilisant la méthode **foreign()** dans le schéma de migration

Question 17 : Comment peut-on créer une tâche planifiée (scheduled task) dans Laravel ?

- Réponse : En utilisant la planification de tâches dans le fichier **app/Console/Kernel.php**

Question 18 : Quelle méthode permet de valider une requête entrante et rediriger automatiquement en cas d'erreur ?

- Réponse : En utilisant la méthode **validate()** dans le contrôleur

Question 19 : Comment peut-on créer une relation un-à-plusieurs dans Eloquent ?

- Réponse : En définissant la relation dans le modèle parent avec **hasMany()**

Question 20 : Quelle méthode Eloquent utilise-t-on pour charger les relations d'un modèle de manière conditionnelle ?

- Réponse : **with()**

Question 21 : Comment peut-on ajouter un attribut muté à un modèle Eloquent ?

- Réponse : En définissant un mutateur dans le modèle

Question 22 : Qu'est-ce que le Tinker et à quoi sert-il ?

- Réponse : Tinker est un REPL (Read-Eval-Print Loop) interactif pour Laravel, utilisé pour tester et interagir avec l'application en utilisant l'artisan.

Question 23 : Comment peut-on publier les assets d'un package ?

- Réponse : En utilisant la commande **php artisan vendor:publish**

Question 24 : Comment génère-t-on une clé d'application dans Laravel ?

- Réponse : En utilisant la commande **php artisan key:generate**

Question 25 : Quelle commande permet de créer un événement et son listener associé ?

- Réponse : **php artisan make:event NomEvenement && php artisan make:listener NomListener**

Question 26 : Comment implémente-t-on une stratégie de cache personnalisée ?

- Réponse : En implémentant une classe personnalisée qui implémente l'interface **CacheContract**

Question 27 : Quel fichier de configuration détermine les services externes que l'application utilise ?

- Réponse : Le fichier **config/services.php**

Question 28 : Comment peut-on utiliser les slots et les composants dans les vues Blade ?

- Réponse : En utilisant **@slot** et **@component** dans les vues Blade

Question 29 : Qu'est-ce qu'un trait dans Laravel et comment peut-on l'utiliser ?

- Réponse : Un trait est un regroupement de méthodes utilisé pour organiser et réutiliser du code dans Laravel. On peut l'utiliser en l'incluant dans les classes pertinentes à l'aide du mot-clé **use**.

Question 30 : Comment peut-on définir un accesseur dans un modèle Eloquent ?

- Réponse : En définissant des méthodes **getNomAttributAttribute()** dans le modèle Eloquent

Question 31 : Comment peut-on utiliser les transactions de base de données dans Laravel ?

- Réponse : En utilisant les méthodes **beginTransaction()**, **commit()** et **rollback()** sur l'objet de connexion de la base de données

Question 32 : Quelle méthode permet de crypter les données avant de les sauvegarder dans la base de données ?

- Réponse : En utilisant la méthode **encrypt()** sur les données

Question 33 : Comment peut-on utiliser Laravel Mix pour compiler les assets ?

- Réponse : En définissant les configurations dans le fichier **webpack.mix.js** puis en exécutant **npm run dev**

Question 34 : Quelle méthode permet de créer une réponse JSON dans un contrôleur ?

- Réponse : En retournant un tableau associatif avec **return response()->json([...])**

Question 35 : Comment peut-on utiliser les événements de modèle dans Laravel ?

- Réponse : En utilisant les événements fournis par le modèle Eloquent, tels que **created**, **updated**, **deleted** ...