**Тестовый контроль «Основы Laravel»**



**1-15. Вопросы с выбором**

1. **Что такое Laravel?**
   1. A) Библиотека для работы с API
   2. B) Фреймворк для разработки на PHP
   3. C) Язык программирования
   4. D) CMS-система

**Ответ:** b. B) Фреймворк для разработки на PHP

1. **Какая команда используется для создания нового проекта Laravel?**
   1. A) php artisan create
   2. B) composer create-project
   3. C) laravel new
   4. D) php artisan new

**Ответ:** c. C) laravel new

1. **Какой файл конфигурации используется для настройки базы данных в Laravel?**
   1. A) config/database.php
   2. B) database/config.php
   3. C) app/database.php
   4. D) config/app.php

**Ответ:** a. A) config/database.php

1. **Какой командой можно запустить сервер Laravel на локальной машине?**
   1. A) php artisan serve
   2. B) laravel serve
   3. C) php artisan run
   4. D) serve

**Ответ:** a. A) php artisan serve

1. **Что такое Artisan в Laravel?**
   1. A) Командный интерфейс для работы с базой данных
   2. B) Веб-сервер
   3. C) Инструмент для автоматизации задач
   4. D) Среда для тестирования

**Ответ:** c. C) Инструмент для автоматизации задач

1. **Какая директива Blade используется для вставки данных в представление?**
   1. A) {{ }}
   2. B) @include
   3. C) {{ include }}
   4. D) {!! !!}

**Ответ:** a. A) {{ }}

1. **Что такое миграции в Laravel?**
   1. A) Скрипты для работы с пользовательским интерфейсом
   2. B) Инструменты для работы с базой данных и создания таблиц
   3. C) Команды для обновления фреймворка
   4. D) Методы для оптимизации кода

**Ответ:** b. B) Инструменты для работы с базой данных и создания таблиц

1. **Какой метод используется для возврата JSON-ответа в контроллере Laravel?**
   1. A) json()
   2. B) response()->json()
   3. C) jsonResponse()
   4. D) jsonResponse()->send()

**Ответ:** b. B) response()->json()

1. **Какую команду использовать для выполнения миграций в Laravel?**
   1. A) php artisan migrate
   2. B) php artisan make:migration
   3. C) php artisan db:migrate
   4. D) php artisan db:make-migration

**Ответ:** a. A) php artisan migrate

1. **Что такое маршруты (routes) в Laravel?**

* A) Это статические пути для доступа к файлам
* B) Это объекты для работы с базой данных
* C) Это пути, которые определяют, как обрабатываются HTTP-запросы
* D) Это таблицы базы данных

**Ответ:** C) Это пути, которые определяют, как обрабатываются HTTP-запросы

1. **Какой шаблон используется в Laravel для создания представлений?**

* A) Blade
* B) Twig
* C) PHP
* D) Smarty

**Ответ:** A) Blade

1. **Какой метод используется для создания нового ресурса в Laravel?**

* A) php artisan make:model
* B) php artisan make:controller
* C) php artisan make:resource
* D) php artisan make:entity

**Ответ:** C) php artisan make:resource

1. **Как установить Laravel через Composer?**

* A) composer install
* B) composer create-project --prefer-dist laravel/laravel
* C) composer laravel
* D) composer setup laravel

**Ответ:** B) composer create-project --prefer-dist laravel/laravel

1. **Что такое Eloquent в Laravel?**

* A) Это система шаблонов для представлений
* B) Это библиотека для работы с базой данных
* C) Это фреймворк для работы с формами
* D) Это система для работы с файлами

**Ответ:** B) Это библиотека для работы с базой данных

1. **Какой метод используется для загрузки аутентифицированного пользователя в Laravel?**

* A) Auth::user()
* B) User::auth()
* C) Auth::login()
* D) Auth::check()

**Ответ:** A) Auth::user()

**16-25. Открытые вопросы**

1. **Объясните, что такое middleware в Laravel и как его можно использовать.**

**Открытый ответ:**

**Middleware в Laravel**

**Middleware** в Laravel — это механизм, который позволяет фильтровать HTTP-запросы, выполняемые вашим приложением. Middleware может использоваться для выполнения различных задач, таких как аутентификация, логирование, проверка прав доступа и т.д.

**Пример использования middleware:**

// Создание middleware

php artisan make:middleware CheckAge

// Регистрация middleware в ядре приложения (app/Http/Kernel.php)

protected $routeMiddleware = [

'check.age' => \App\Http\Middleware\CheckAge::class,

];

// Применение middleware к маршруту

Route::get('user/profile', function () {

//

})->middleware('check.age');

1. **Какая разница между get и post запросами в контексте маршрутов Laravel?**

**Открытый ответ:**

**Разница между GET и POST запросами в Laravel**

* **GET-запросы** используются для получения данных с сервера. Они кэшируются, могут быть закладками и сохраняются в истории браузера. Данные передаются через URL.
* **POST-запросы** используются для отправки данных на сервер. Они не кэшируются, не могут быть закладками и не сохраняются в истории браузера. Данные передаются в теле запроса.

**Пример маршрутов:**

Route::get('/user', 'UserController@index');

Route::post('/user', 'UserController@store');

1. **Что такое контроллеры в Laravel и для чего они используются? Приведите пример.**

**Открытый ответ:**

**Контроллеры в Laravel**

**Контроллеры** в Laravel используются для группировки логики обработки запросов. Они помогают организовать код и сделать его более читаемым и поддерживаемым.

**Пример контроллера:**

// Создание контроллера

php artisan make:controller UserController

// Пример метода в контроллере

public function index()

{

$users = User::all();

return view('users.index', compact('users'));

}

1. **Как в Laravel настроить связь между моделями "один ко многим"? Приведите пример.**

**Открытый ответ:**

**Связь "один ко многим" в Laravel**

Связь "один ко многим" устанавливается, когда одна модель имеет множество связанных моделей.

**Пример:**

// Модель Post

class Post extends Model

{

public function comments()

{

return $this->hasMany(Comment::class);

}

}

// Модель Comment

class Comment extends Model

{

public function post()

{

return $this->belongsTo(Post::class);

}

}

1. **Что такое зависимость инъекций (Dependency Injection) в Laravel? Приведите пример, как это используется в контроллерах.**

**Открытый ответ:**

**Зависимость инъекций (Dependency Injection) в Laravel**

**Dependency Injection** — это паттерн, который позволяет внедрять зависимости в класс через конструктор или методы. В Laravel это используется для упрощения тестирования и управления зависимостями.

**Пример:**

// Внедрение зависимости в контроллер

public function \_\_construct(UserRepository $users)

{

$this->users = $users;

}

1. **Как создать кастомную миграцию в Laravel? Опишите процесс создания и применения миграций.**

**Открытый ответ:**

**Создание кастомной миграции в Laravel**

Миграции используются для управления структурой базы данных.

**Пример создания и применения миграции:**

// Создание миграции

php artisan make:migration create\_users\_table

// Пример миграции

public function up()

{

Schema::create('users', function (Blueprint $table) {

$table->id();

$table->string('name');

$table->string('email')->unique();

$table->timestamps();

});

}

// Применение миграции

php artisan migrate

1. **Объясните, как можно валидировать данные в Laravel при помощи встроенных правил валидации. Приведите пример.**

**Открытый ответ:**

**Валидация данных в Laravel**

Валидация данных в Laravel может быть выполнена с помощью встроенных правил валидации.

**Пример:**

public function store(Request $request)

{

$validated = $request->validate([

'name' => 'required|max:255',

'email' => 'required|email|unique:users',

]);

// Сохранение данных

}

1. **Как работают события и слушатели (Events & Listeners) в Laravel? Опишите, как их можно использовать для обработки асинхронных задач.**

**Открытый ответ:**

**События и слушатели (Events & Listeners) в Laravel**

События и слушатели позволяют реализовать асинхронную обработку задач.

**Пример:**

// Создание события

php artisan make:event UserRegistered

// Создание слушателя

php artisan make:listener SendWelcomeEmail --event=UserRegistered

// Регистрация слушателя в EventServiceProvider

protected $listen = [

'App\Events\UserRegistered' => [

'App\Listeners\SendWelcomeEmail',

],

];

1. **Как в Laravel можно реализовать аутентификацию пользователей? Опишите основные шаги.**

**Открытый ответ:**

**Аутентификация пользователей в Laravel**

Аутентификация пользователей в Laravel может быть реализована с помощью встроенных механизмов.

**Основные шаги:**

1. Установка пакета аутентификации: composer require laravel/ui
2. Генерация аутентификационных маршрутов и представлений: php artisan ui vue --auth
3. Настройка базы данных и миграций.
4. Регистрация маршрутов аутентификации в web.php.
5. **Как можно использовать пагинацию для вывода результатов в Laravel? Приведите пример использования пагинации в Eloquent.**

**Открытый ответ:**

**Пагинация в Laravel**

Пагинация позволяет разбивать большие наборы данных на страницы.

**Пример использования пагинации в Eloquent:**

// Получение данных с пагинацией

$users = User::paginate(15);

// Отображение данных в представлении

return view('users.index', compact('users'));

// В представлении

{{ $users->links() }}

**Вопросы на сопоставление**

1. **Сопоставьте компоненты Laravel с их назначением.**

**Сопоставьте компоненты Laravel с их назначением.**

**Термины:**

1. Middleware
2. Eloquent ORM
3. Blade
4. Миграции
5. Контроллеры

**Назначения:**

A) Управляет взаимодействием с базой данных и предоставляет активные модели для работы с таблицами.  
B) Обрабатывает HTTP-запросы, выполняя дополнительные действия до или после основного запроса.  
C) Шаблонный движок, используемый для генерации HTML в представлениях.  
D) Позволяет управлять схемой базы данных, создавая или обновляя таблицы через версии.  
E) Управляют бизнес-логикой приложения и обрабатывают запросы, возвращая ответы.

**Ответ:**

**Вопрос 26: Сопоставьте компоненты Laravel с их назначением.**

1. **Middleware**
   * **B) Обрабатывает HTTP-запросы, выполняя дополнительные действия до или после основного запроса.**
2. **Eloquent ORM**
   * **A) Управляет взаимодействием с базой данных и предоставляет активные модели для работы с таблицами.**
3. **Blade**
   * **C) Шаблонный движок, используемый для генерации HTML в представлениях.**
4. **Миграции**
   * **D) Позволяет управлять схемой базы данных, создавая или обновляя таблицы через версии.**
5. **Контроллеры**
   * **E) Управляют бизнес-логикой приложения и обрабатывают запросы, возвращая ответы.**
6. **Сопоставьте компоненты Laravel с их функциональностью.**

**Термины:**

1. **Artisan**
2. **Service Providers**
3. **Routes**
4. **Eloquent Relationships**
5. **Validation**

**Функциональность:**

A) Обеспечивает механизм для управления различными действиями в приложении через командный интерфейс.  
B) Определяет, какие URL-адреса обрабатывает ваше приложение и какой контроллер или метод будет вызван для обработки запроса.  
C) Механизм для работы с базой данных, позволяющий строить связи между моделями, такими как "один к одному", "один ко многим" и другие.  
D) Отвечает за выполнение логики проверки данных, например, проверку email на правильность или наличие обязательных полей.  
E) Это классы, которые помогают интегрировать и настраивать различные компоненты приложения и сторонние пакеты.

**Ответ:**

**Вопрос 27: Сопоставьте компоненты Laravel с их функциональностью.**

1. **Artisan**
   * **A) Обеспечивает механизм для управления различными действиями в приложении через командный интерфейс.**
2. **Service Providers**
   * **E) Это классы, которые помогают интегрировать и настраивать различные компоненты приложения и сторонние пакеты.**
3. **Routes**
   * **B) Определяет, какие URL-адреса обрабатывает ваше приложение и какой контроллер или метод будет вызван для обработки запроса.**
4. **Eloquent Relationships**
   * **C) Механизм для работы с базой данных, позволяющий строить связи между моделями, такими как "один к одному", "один ко многим" и другие.**
5. **Validation**
   * **D) Отвечает за выполнение логики проверки данных, например, проверку email на правильность или наличие обязательных полей.**