

## Тестовый контроль «Основы Laravel»



### 1-15. Вопросы с выбором

#### 1. Что такое Laravel?

- a. A) Библиотека для работы с API
- b. B) Фреймворк для разработки на PHP
- c. C) Язык программирования
- d. D) CMS-система

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

#### 2. Какая команда используется для создания нового проекта Laravel?

- a. A) php artisan create
- b. B) composer create-project
- c. C) laravel new
- d. D) php artisan new

**Ответ:** c. C) laravel new

#### 3. Какой файл конфигурации используется для настройки базы данных в Laravel?

- a. A) config/database.php

- b. B) database/config.php
- c. C) app/database.php
- d. D) config/app.php

**Ответ:** a. A) config/database.php

**4. Какой командой можно запустить сервер Laravel на локальной машине?**

- a. A) php artisan serve
- b. B) laravel serve
- c. C) php artisan run
- d. D) serve

**Ответ:** a. A) php artisan serve

**5. Что такое Artisan в Laravel?**

- a. A) Командный интерфейс для работы с базой данных
- b. B) Веб-сервер
- c. C) Инструмент для автоматизации задач
- d. D) Среда для тестирования

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

**6. Какая директива Blade используется для вставки данных в представление?**

- a. A) {{ }}
- b. B) @include
- c. C) {{ include }}
- d. D) {!! !!}

**Ответ:** a. A) {{ }}

**7. Что такое миграции в Laravel?**

- a. A) Скрипты для работы с пользовательским интерфейсом
- b. B) Инструменты для работы с базой данных и создания таблиц
- c. C) Команды для обновления фреймворка
- d. D) Методы для оптимизации кода

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

**8. Какой метод используется для возврата JSON-ответа в контроллере Laravel?**

- a. A) json()
- b. B) response()->json()
- c. C) jsonResponse()
- d. D) jsonResponse()->send()

**Ответ:** b. B) response()->json()

**9. Какую команду использовать для выполнения миграций в Laravel?**

- a. A) php artisan migrate
- b. B) php artisan make:migration
- c. C) php artisan db:migrate
- d. D) php artisan db:make-migration

**Ответ:** a. A) php artisan migrate

**10. 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

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

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

**Ответ:** A) Blade

**12. 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

**13. 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

**14. 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

**15. 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**

**16. 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');
```

## 17. 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');
```

## 18. 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'));
```

```
}
```

## 19. 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);
    }
}
```

## 20. 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;
}
```

## 21. 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
```

**22. 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  
}
```

**23. 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',  
    ],  
];
```

## **24. 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`.

## **25. 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**

**26. 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:**

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

#### **Ответ:**

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

### **1. Middleware**

- В) Обрабатывает HTTP-запросы, выполняя дополнительные действия до или после основного запроса.

### **2. Eloquent ORM**

- А) Управляет взаимодействием с базой данных и предоставляет активные модели для работы с таблицами.

### **3. Blade**

- С) Шаблонный движок, используемый для генерации HTML в представлениях.

### **4. Миграции**

- D) Позволяет управлять схемой базы данных, создавая или обновляя таблицы через версии.

### **5. Контроллеры**

- Е) Управляют бизнес-логикой приложения и обрабатывают запросы, возвращая ответы.

## **27. Сопоставьте компоненты Laravel с их функциональностью.**

### **Термины:**

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

### **Функциональность:**

А) Обеспечивает механизм для управления различными действиями в приложении через командный интерфейс.

В) Определяет, какие URL-адреса обрабатывает ваше приложение и какой контроллер или метод будет вызван для обработки запроса.

С) 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.

Е) Это классы, которые помогают интегрировать и настраивать различные компоненты приложения и сторонние пакеты.

### **Ответ:**

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

**1. Artisan**

- А) Обеспечивает механизм для управления различными действиями в приложении через командный интерфейс.

**2. Service Providers**

- Е) Это классы, которые помогают интегрировать и настраивать различные компоненты приложения и сторонние пакеты.

**3. Routes**

- В) Определяет, какие URL-адреса обрабатывает ваше приложение и какой контроллер или метод будет вызван для обработки запроса.

**4. Eloquent Relationships**

- С) 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.