

Web-программирование

Месон 01

Карасов С.В.

09.03.01

ИВМ-3, 3 курс

Тест 51

Тестовый контроль „ Основы Laravel “

1.-15. Вопросы с выбором

1. б. В) Фреймворк для разработки на PHP

2. с. С) laravel new

3. а. А) config / database . php

4. а. А) PHP artisan serve

5. с. С) Инструмент для автоматизации задач

6. а. А) {{ }}

7. б. В) Инструменты для работы с БД и создания таблиц.

8. б. В) response () → json ()

9. а. А) php artisan migrate

10. С) Это пути, которые определяют, как обрабатываться

HTTP-запросы.

11. A) Blade

12. C) php artisan make:resource

13. B) composer create-project --
prefer-dist laravel/laravel

14. B) Это средство для
работы с БД;

15. A) Auth::user()

Вопросы на соответствие

26.

1 B

2 A

3 C

4 D

5 E

27.

1 A

2 E

3 B

4 C

5 D

16.-25. Открытые вопросы

16.

Open 16

17.

Open 17

18.

Open 18

19.

Open 19

20.

Open 20

21.

Open 21

22.

Open 22

23.

Imbern 23

24.

Imbern 24

25.

Imbern 25

Бед

Мечта 51

Ковалев С. В.

3 куст

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');
```

Дмитрий 16

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');
```

Итого 14

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'));
}
```

Verben 18

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);
    }
}
```

Андрей 19

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;
}
```

Done 20

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
```

Данная 21

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
}
```

Amber 22

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',  
    ],  
];
```

Домашн 23

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`.

Artem 24

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() }}
```

Пример 25