



Framework Laravel

Frontend



Проверить, идет ли запись

Меня хорошо видно && слышно?



Ставим “+”, если все хорошо
“-”, если есть проблемы

Работа с Frontend В Laravel



Сергей Бакланов

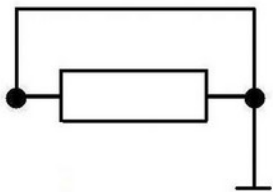
25 лет опыта работы программистом
и предпринимателем в IT

Владелец и ведущий специалист
в php-cat.com

<https://vk.com/phpcatcom>

Пишите, звоните:

89-222-6-222-89



**СОПРОТИВЛЕНИЕ
БЕСПОЛЕЗНО!!!**

YAPLAKAL.COM

Правила вебинара



Активно
участвуем



Off-topic обсуждаем
в Телеграмм



Задаем вопрос(ы)
в чат



Вопросы вижу в чате,
могу ответить не сразу



Карта курса

1 Установка

3 Хранилище для Laravel-продукта

5 Маршруты и представления

7 DI

2 Frontend

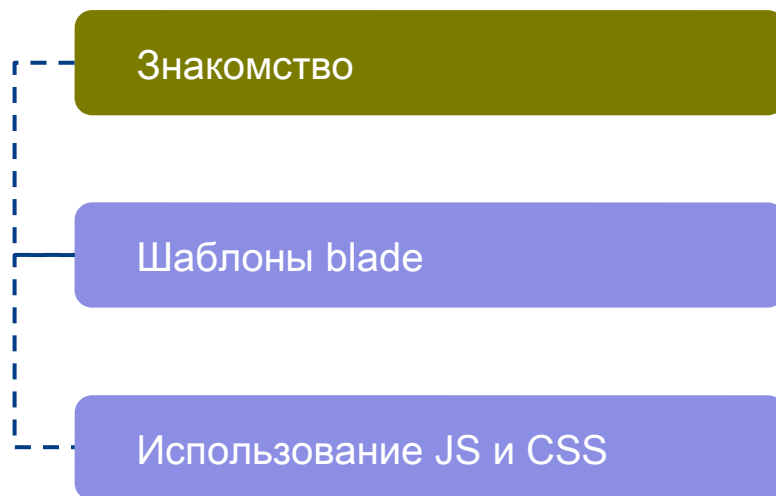
4 Eloquent

6 Контроллеры

8 Авторизация и аутентификация

9 Тестирование

Маршрут вебинара



Цели вебинара

К концу занятия вы сможете

1 ▪	Создавать страниц, используя шаблонизатор blade
2 ▪	Подключать и использовать JS CSS
3 ▪	Использовать mix для сборки js css в обобщённые и минимизированные файлы

Blade шаблонизатор для работы с html страницами

Blade - это шаблонизатор Laravel, помогающий создавать переиспользуемые view

- **расширение файла .blade.php**
- **располагаются в resources/views**

Синтаксис blade состоит из:

**директив, которые начинаются на @
переменных, они окружены {{ }}**

Композиция шаблонов

- `resources/views` - главный каталог व्युшек
- `resources/views/layouts` - каталог для layout'ов
- `resources/views/blocks` - каталог с общими элементами, компонентами
- `resources/views/blocks/navbar/index`
- `resources/views/blocks/alerts/[info|danger|...]`
- `resources/views/auth/[layout|login|register|remember|...]`
- `resources/views/home/[layout|index|components|...]`
- `resources/views/cms/posts/[index|create|edit|show|...]`
-

Композиция шаблонов

Далее в новом файле наследуют шаблон и определяют все необходимые блоки:

```
@extends('layouts.app')
```

```
@section('title', 'Page Title')
```

```
@section('sidebar')
```

```
    @parent
```

```
    <p>This is appended to the master sidebar.</p>
```

```
@endsection
```

```
@section('content')
```

```
    <p>This is my body content.</p>
```

```
@endsection
```

Отображение данных

{{ \$varname }} - экранирует html теги (htmlspecialchars), безопасен для нефильтрованного пользовательского ввода

{!! \$varname !!} - ничего не экранирует, нельзя применять для нефильтрованного пользовательского ввода, возможна XSS (php аналог <?= \$varname ?>)

{{-- comment --}} - комментарий, который не будет рендериться внутри скобок может находиться любой валидный PHP код

Экранирование директив

Просто добавьте @ в начало: _x0003_

@{{ \$varname }}

@@if()

Либо для многострочного экранирования:

```
@verbatim
  {{ $varname }}
  @if(true)
  @endif
@endverbatim
```


Композиция шаблонов

Шаблон разбивают на логические блоки,
и потом их переиспользуют

Сначала создается главный шаблон,
в котором указываются секции

много строчный

```
@section('name')  
default markup  
@show --
```

однострочный

```
@yield('name', 'default')
```

Директивы циклов

```
{ eat()  
  sleep()  
  code()  
  repeat()
```

```
@for($i = 0; $i < 10; $i++)  
    The current value is {{ $i }}  
@endfor
```

```
@foreach($users as $user)  
    <p>This is user {{ $user->id }}</p>  
@endforeach
```

```
@forelse($users as $user)  
    <li>{{ $user->name }}</li>  
@empty  
    <p>No users</p>  
@endforelse
```

```
@while(true)  
    <p>I'm looping forever.</p>  
@endwhile
```

```
@foreach ($users as $user)  
    @continue($user->type == 1)  
    <li>{{ $user->name }}</li>  
    @break($user->number == 5)  
@endforeach
```

```
@foreach ($users as $user)  
    @if ($loop->first)  
        This is the first iteration.  
    @endif
```

```
    @if ($loop->last)  
        This is the last iteration.  
    @endif
```

```
<p>This is user {{ $user->id }}</p>  
@endforeach
```


Директивы разметки

```
@php
    $isActive = false;
    $hasError = true;
@endphp

<span @class([
    'p-4',
    'font-bold' => $isActive,
    'text-gray-500' => !$isActive,
    'bg-red' => $hasError,
])></span>
```

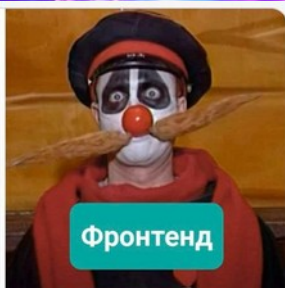
```
<input @checked(old('active', $user->active)) />
<option @selected(old('version') == $version)>
<button @disabled($errors->isEmpty())>
```



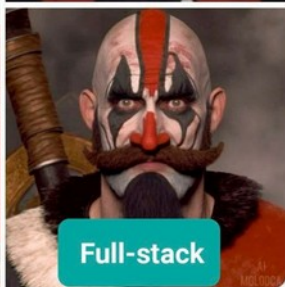
Кеширование шаблонов



Бэкенд



Фронтенд



Full-stack

На проде нужно обязательно кэшировать скомпилированные шаблоны, чтобы сократить время ответа

> `php artisan view:cache`

> `php artisan view:clear`

JS + CSS

**работа с компоновкой
и минимизацией
ЭТИХ ТИПОВ ДАННЫХ**

Подключение JS CSS

Vite используется для сборки ресурсов проекта в Laravel 9+
Работает только входными файлами (app.css, app.js)

Вставка скомпилированного js файла в код .blade.php страницы

```
@vite('resources/js/app.js')
```

и подключение файлов стилей (рекомендуют подключать css импортом в js, а не грузить отдельным файлом в html странице)

```
@vite('resources/js/app.css')
```

Запуск сборки в терминале

```
> npm run dev # разработка
```

(с автообновлением сайта при изменении исходников)

```
> npm run build # сборка
```

(минимизированная сборка для «прода»)

Домашнее задание

На базовой странице welcome, удалить содержимое и расположить таблицу, где 10 строк, в каждой строке цифра

в первой строке «2»
и в каждой следующей строке цифра увеличивается на 2

использование полученных знаний
и функционала blade приветствуется





С какими впечатлениями уходите с вебинара?



Какие элементы будете применять чаще в практике из того что узнали на вебинаре ?



**Заполните,
пожалуйста,
опрос о занятии
по ссылке в чате**