Framework Laravel

Веб-фреймворк, который сделает вашу работу интереснее, проще и быстрее

Длительность курса: 168 академических часов

1 Developing

1	**Парадигмы программирования**	рассказать о необходимости применения стандартизированных и общепринятых подходов поговорить о разных парадигмах, обсудить плюсы и минусы сделать акцент на том, какие парадигмы применяются в современных PHP-фреймворках

2 **Архитектура кода**

перечислить фундаментальные принципы программирования

Домашние задания

1 Анализ своих проектов

Цель: Привести пример своего проекта, который был разработан до курса. Проанализировать код проекта на предмет пройденного материала по парадигмам и архитектурам. Указать места, требующие доработки.

В качестве ДЗ нужно прислать ссылку на репозиторий, а также файл с результатами анализа.

---

3 **Паттерны**

вспомнить или узнать основные паттерны в веб-программировании
обсудить как их применять, когда и зачем
читать UML-схемы

4	**Практики хорошего кода**	учимся важной части командной работы - практикам хорошего кода поговорим о рекомендация "дядюшки Боба" Домашние задания 1 Продолжение анализа Цель: Продолжить работу с примером из прошлого ДЗ. Проанализировать код проекта на предмет пройденного материала по паттернам и коду. 1. Для паттернов: показать места применения изученных паттернов или места, где их можно было бы применить 2. Для практик хорошего кода: указать места на доработку
5	**Linux**	объяснить актуальность Linux систем (почему сервера в основном работает на этой ОС) рассмотреть процессы, потоки обсудить пользователей, привилегии научиться использовать ряд полезных утилит (top, grep etc) научиться использовать bash скрипты

6 **Виртуализация, контейнеры**

обсудить ситуации, когда использования «железных» серверов недостаточно
поговорить про понятие виртуализации
рассказать о разных типах и инструментах виртуализации
познакомиться с контейнерами
собрать рабочее окружение для выполнения простого PHP-скрипта
рассмотреть облачных провайдеров

Домашние задания

1 Построение среды разработки

Цель: Собрать среду разработки приложения на PHP 1. Виртуальная машина - NGINX - PHP-FPM - MySQL/Postgres - Redis/Memcached 2. Docker - Контейнер NGINX - Контейнер FPM - Контейнер БД - Контейнер Memcached - Volume с кодом проекта

Виртуальную машину можно собрать с применением VirtualBox + Vagrant
Контейнеры должны запускаться через docker-compose

7 **PHP in CLI**

вспомнить, что PHP-скрипты можно вызывать не только из браузера
объяснить, зачем нужны служебные скрипты
создать выполнение скрипта по расписанию
создать скрипт-демон, рассказать про особенности демонизации в PHP (zts, память и тд)
IPC

Домашние задания

1 PHP-сокеты

Цель: Написать два PHP скрипта, который запускаются на одной машине и обмениваются сообщениями через unix-сокеты

Один из скриптов является сервером. При запуске раз в несколько секунд он шлёт случайное сообщение клиенту.
Второй скрипт - клиент. Он принимает сообщение от сервера, выводит его на экран и отправляет ответ серверу "Принято"
Сервер выводит ответ на экран: "Сообщение %текст сообщения% принято клиентом"

2 Знакомство с фреймворком. Пишем базовый функционал

1 **Установка и «Hello, world»**

изучить способы установки приложения (напрямую через Composer, Homestead/Valet);
изучить структуру файлов и реализацию MVC;
запустить первый приветственный контроллер;
Бегло познакомимся с простой маршрутизацией;
Обсудить применимость фреймворка;
установить Laravel IDE Helper

Домашние задания

1 Рабочая среда для фреймворка

Цель: Необходимо подготовить среду для дальнейшей разработки. Разрешается использовать Homestead или Laradock.

Развернуть фреймворк у себя в выбранной среде
В качестве ДЗ можно сдавать скриншот рабочей среды

2 Идея курсового проекта

Цель: Выбрать цель для курсового проекта из нижепредложенных или предложить свою Новостной портал Доска объявлений Сервис ToDo Telegram-бот поиска информаций по конструкциям выбранного языка (PHP, Go, C и т.д.) Сайт знакомств 1 занятие

Выбрать проект
Оповестить преподавателя
Обосновать выбор

2 **Фронтэнд**

поработать с CSS и JS в Laravel;
познакомиться с Elixir;
познакомиться с Blade;
узнать о совместимости Laravel с Vue и Bootstrap;
изучить принципы работы приложения с фронтэнд-фреймворками.

Домашние задания

1 Создать несколько фронт-страниц

Цель: Начинаем работу над частями нашего проекта. Для выбранной Вами темы нужно создать несколько страниц под управлением фронтэнд инструментов Laravel 1. Главная 2. Страница пользователя 3. Страница регистрации (прототип, без самой регистрации) 4. Абстрактная статическая страница

---

3 **Хранилище для Laravel-продукта**

познакомиться с Artisan;
научиться накатывать миграции;
познакомимся с Eloquent;
создать базовые модели;
узнать про Namespacing и MVC в Laravel

Домашние задания

1 Сформировать модели

Цель: Продолжаем работу над проектом, создавая бизнес-сущности.

сформировать модели, которые потребуются в продукте:
- модель пользователя и его ролей
- модель данных разных частей страницы
- например, модель задачи для ToDo

4	**Маршруты и представления**	углубиться в логику маршрутизации изучить правила обработки запросов научиться писать кастомные правила
5	**Контроллеры**	углубимся в работу с контроллерами узнаем, что такое FSUC и почему он плох познакомиться с понятиями и реализацией посредников Домашние задания 1 Администраторский интерфейс Цель: Начинаем создание администраторского интерфейса для управления проектом Интерфейс базируется на простых CRUD-действиях, но допускает усложнение аспектами бизнес-логики
6	**DI**	вспомнить суть концепции IoC изучить реализацию DI в Laravel научиться использовать DI-контейнеры

7	**Авторизация и аутентификация**	знать правила и порядок авторизации и аутентификации пользователей в Laravel добавлять к своему приложению разграничение прав реализовывать авторизацию по токему Домашние задания 1 Учим приложение распознавать пользователя Цель: 1. Написать логику для авторизации и аутентификации 2. Разделить права на доступ к админ-интерфейсу 3. Дать пользователям возможность редактировать созданные ими же функции
8	**Тестирование**	выполнять тесты в Laravel; покрывать приложение тестами при помощи различных инструментов
9	**Логирование и работа с файловой системой**	научиться собирать информацию о работе приложения узнать, как и куда можно собирать техническую информацию Домашние задания 1 Тесты и логирование Цель: Проверить покрытие тестами Вашего проекта при помощи php-code-coverage (https://github.com/sebastianbergmann/php-code-coverage) Постараться увеличить покрытие (нормой считается 80%)

3 Продолжаем усложнять логику проекта

1 **Полезные функции, встроенные во фреймворк**

узнать, что ещё умеет фреймворк, чтобы не писать велосипеды

Домашние задания

1 Кэширование

Цель: Перенести сессии в Memcached

---

2 **Middleware**

ознакомиться с идеей middleware
повторить и расширить знания о посредниках

---

3 **Кэширование**

ускорить работу с данными
применять различные механизмы кэширования
писать логику для работы с ними
поддерживать сервисные процессы кэширования

Домашние задания

1 Применить кэширование

Цель: применить кэширование; замерить производительность с кэшем и без него; реализовать очистку и прогрев кэша

---

4 **Пакеты**

упаковывать функционал в packages
создавать полноценные модули

5	**DDD в Laravel**	познакомиться с Domain Driven Development применять концепцию в своих проектах на Laravel Домашние задания 1 Доменная модель Цель: Смоделировать сущности наработанного функционала в виде доменов (можно использовать блок-схемы)
6	**Команды и шина**	отделять групп атомарных действий в команды
7	**Очереди**	сделать следующий шаг в развитии механизма команд Домашние задания 1 Асинхронные процессы Цель: Настраиваем пайплайн (в зависимости от проекта сущности будут меняться) 1. Пользователь выполняет действие (создание задачи, размещение новости и т.п.) 2. В очереди публикуется событие 3. На событие реагирует 1 и более слушателей (примеры: телеграм-оповещение, email, push-уведомление, создание лога и т.д.)

4 API

1	**Контракты и фасады**	углубиться в контракты и фасады научиться лучше применять техники инкапсуляции функционала и его структурирования
2	**Envoy и развёртывание**	научиться деплоить приложение на Laravel применять для деплоя встроенный функционал Домашние задания 1 Deployment Цель: Развернуть имитацию Production-среды по подобию Dev Организовать автоматическую выкатку приложения
3	**Scout и полнотекстовый поиск**	узнать про понятие полнотекстового описка научить приложение быстро искать информацию
4	**Переходим к Stateless**	вспомнить про различия Stateless и Stateful начать работать с API Домашние задания 1 API Цель: Выделить часть приложения, которую можно вынести в API (например, создание задачи в ToDo) Создать CRUD-API для внешней системы Задокументировать его (*swagger)

5	**Passport API**	научиться улучшенной авторизации на уровне API изучить Passport Домашние задания 1 Научить API работать с авторизацией Цель: Для доступа к API клиенты должны получить токен
6	**Lumen**	познакомиться с light-weight версией фреймворка узнать об аспектах реализации SOA при помощи Lumen Домашние задания 1 Создать простой API для отображения личного кабинета пользователя в мобильном приложении; Учесть аспекты авторизации
7	**Тестирование API**	научиться тестировать интеграционные составляющие Домашние задания 1 Автотесты для API Цель: покрыть тестами API обратить внимание на тестирование функционала авторизации API помнить про нормы покрытия

8 **Планирование задач**

организовывать работу систем обслуживания приложения
планировать технические и системные скрипты

Домашние задания

1 Регулярные задачи

Цель: Вынести задачи кэширования в cron (сбор статистики, бэкапы и т.п.) Учесть возможность запуска приложения на N серверах в кластере

5 Проектная работа

1	**Вводное занятие**	определиться с целями проектов Домашние задания 1 Проектная работа Цель: Выбрать проект, описать его цели и пошаговый план работ. Собрать команду (если есть желание работать в группе) Выбрать подход к разработке и контролю проекта (если есть желание работать в группе)
2	**Консультационное занятие**	задать вопросы по проектам
3	**Защита проектов**	финал обучения