

Framework Laravel

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

Длительность курса: 136 академических часов

1 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;
познакомиться с простой маршрутизацией;
обсудить применимость фреймворка;
установить Laravel IDE Helper.

Домашние задания

- 1** Рабочая среда для фреймворка и будущий проект

Цель: Сделать самые первые шаги в освоении нового инструмента

Развернуть рабочую среду для будущих заданий

Определиться с целевым проектом на время изучения

1. 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. Выбрать цель для курсового проекта из нижепредложенных или предложить свою

Новостной портал

Доска объявлений

Сервис ToDo

Telegram-бот поиска информации 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 Mix;
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: Начать работу над частями нашего проекта

Создать базовые статические модули

Освоиться в разработке под Laravel

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;
узнать про Namespacing и MVC в Laravel.

Домашние задания

- 1 Спроектировать базы данных и создать миграции

Цель: Продолжить работу над проектом, создав хранилище данных

Изучить аспекты менеджмента БД в Laravel

Научиться работать с миграциями

1. Спроектировать базу данных
 2. Создать миграции для таблиц
 3. Воспроизвести требуемую структуру через миграции
-

4 Eloquent ORM

познакомиться с Eloquent;
создать базовые модели;
разобрать связи между моделями

Домашние задания

- 1 Создание моделей

Цель: Создать модели для БД

Создать фабрики и сиды для моделей

сформировать модели, которые потребуются в продукте:

- модель пользователя и его ролей
 - модель данных разных частей страницы
 - например, модель задачи для ToDo
- написать фабрики и сиды для моделей
-

5 Маршруты и представления

углубиться в логику маршрутизации;
изучить правила обработки запросов;
писать кастомные правила.

6 Контроллеры

углубиться в работу с контроллерами;
узнать, что такое FSUC и почему он плох;
познакомиться с понятиями и реализацией посредников.

Домашние задания

1 Администраторский интерфейс

Цель: Проект должен быть управляем. И админка - один из базовых элементов для этого. Начинаем создание администраторского интерфейса для управления проектом. Интерфейс базируется на простых CRUD-действиях, но допускает усложнение аспектами бизнес-логики

1. Написать логику управления страницами
 2. Написать логику управления одной-двумя wybranными сущностями
-

7 DI

вспомнить суть концепции IoC;
изучить реализацию DI в Laravel;
использовать DI-контейнеры.

8 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: Разделить приложение на зоны с различным уровнем доступа.

Как минимум, админка должна быть доступна только ограниченному кругу лиц.

1. Написать логику для авторизации и аутентификации
2. Разделить права на доступ к админ-интерфейсу
3. Дать пользователям возможность редактировать созданные ими же функции

9 Тестирование

выполнять тесты в Laravel; покрывать приложение тестами при помощи различных инструментов.

Домашние задания

1 Тестируем приложение

Цель: Автоматические тесты - залог спокойного сна. И Laravel предоставляет для их организации всё необходимое.

Нарастить покрытие тестами Вашего проекта
Постараться увеличить покрытие (нормой считается 80%)

2 Продолжаем усложнять логику проекта

1 Логирование и полезные функции фреймворка

собрать информацию о работе приложения; объяснить что умеет фреймворк, чтобы не писать велосипеды.

Домашние задания

1 Логирование в Slack

Цель: Инциденты в приложении требуют изучения. Для этого нужна исходная информация. В её роли часто выступают логи. Научится сообщать об ошибках напрямую в Slack

Создать свой workspace в Slack
Описать свой обработчик ошибок
Отправлять сообщения об ошибках в Slack канал

2 Middleware

познакомиться с идеей middleware; расширить знания о посредниках.

Домашние задания

1 Создаем своего посредника

Цель: Научится создавать и работать с посредниками

Создайте посредника для своего приложения. Для примера, можно создать посредника, который будет ставить нужную локаль (ru, en) в зависимости от переданного параметра в урле

- /en/dashboard - Locale EN
- /ru/dashboard - Locale RU

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: Научиться использовать один из базовых инструментов ускорения работы приложения
Познакомиться с библиотеками Laravel для кэширования

1. Применить кэширование на уровне логики
 2. Перенести хранение сессий в Memcached
 3. * Замерить производительность с кэшем и без него;
 4. * Реализовать 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: Закрепить понимание доменной модели на практике

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: Научиться реализовывать асинхронное взаимодействие на базе Laravel

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.)

1 **Контракты и фасады** углубиться в контракты и фасады; применять техники инкапсуляции функционала и его структурирования.

2 **Envoy и развёртывание** деплоить приложение на Laravel; применять для деплоя встроенный функционал.

Домашние задания

1 Deployment

Цель: Научиться доставлять код в автоматическом режиме

1. Развернуть имитацию Production-среды по подобию Dev
 2. Организовать автоматическую выкатку приложения
-

3 **Scout и полнотекстовый поиск** узнать про понятие полнотекстового описки; научить приложение быстро искать информацию.

4 **Переходим к Stateless** вспомнить про различия Stateless и Stateful; работать с API.

5 **Lumen** познакомиться с light-weight версией фреймворка; узнать об аспектах реализации SOA при помощи Lumen.

6 Passport API

научиться улучшенной авторизации на уровне API; изучить Passport.

Домашние задания

1 Написание API

Цель: Научиться работать в Stateless парадигме с API

1. Выделить часть приложения, которую можно вынести в API (например, создание задачи в ToDo)

Создать CRUD-API для внешней системы
Задokumentировать его (*swagger)

2. Добавить в API функционал авторизации клиентов по токену

7 Тестирование API

тестировать интеграционные составляющие.

Домашние задания

1 Работа с API

Цель: Продолжить работу с API

1. Создать простой API для отображения личного кабинета пользователя в мобильном приложении. Учесть аспекты авторизации

2. Покрыть тестами API. Обратит 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: Автоматизировать рутинные задачи средствами фреймворка

1. Вынести задачи кэширования в cron (прогрев, сброс и т.п.)
2. * Учесть возможность запуска приложения на N серверах в кластере

4 Проектная работа

- 1 Выбор темы и организация проектной работы**

выбрать и обсудить тему проектной работы;
спланировать работу над проектом;
ознакомиться с регламентом работы над проектом.

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 Консультация по проектам и домашним заданиям**

получить ответы на вопросы по проекту, ДЗ и по курсу.

- 3 Защита проектных работ**

защитить проект и получить рекомендации экспертов.

- 4 Подведение итогов курса**

узнать, как получить сертификат об окончании курса, как взаимодействовать после окончания курса с OTUS и преподавателями, какие вакансии и позиции есть для выпускников (опционально - в России и за рубежом) и на какие компании стоит обратить внимание.