



Vue.js разработчик

Быстрый JS-фреймворк для эффективной frontend-разработки

Длительность курса: 106 академических часов

1 **Введение в курс, современная веб-разработка**

Цели занятия:

использовать основные инструменты современной разработки;
объяснить принципы работы webpack;
инициализировать проект npm;
различать ES5 и ES6 - старый и современный синтаксис JS.

Краткое содержание:

используемые технологии в современном фронтенде;
структура проекта, поиск нужной информации;
Webpack и babel;
основы работы с npm, webpack, babel, npm, package.json.

Цели занятия:

объяснить преимущества Vue;
применять инструменты для быстрого развертывания проекта Vue;
создавать и настраивать новый проект с Vuecli;
объяснить отличия Vue от React и других фреймворков

Краткое содержание:

почему правильно выбрать Vue для фронтенд разработки в 2021;
инструменты и ресурсы для разработчика;
как установить необходимое, как начать новый Vue проект;
Vue CLI, Vue developer tools, Vetur (VSC extension).

Домашние задания

1 Создать проект с Vue-cli, настроить конфиги

Цель: Научиться разворачивать проект с нуля, разобраться какие какие есть настройки в vue.config.js

Сетап проекта Vue с нуля, кастомизация настроек базового проекта (расширение webpack);
Альтернативные варианты начального проекта на Vue.

1. Создать репозиторий на github
 2. Сделать проект на vue с использованием vue-cli
 3. Добавить кастом конфиг (опционально)
 4. прислать ссылку на репозиторий
-

3 Шаблоны

Цели занятия:

использовать встроенные директивы шаблонов Vue;
различать v-if и v-show;
написать переключения и списки в шаблонах;
переключать class и style и другие атрибуты у элементов в шаблонах vue;

Краткое содержание:

существующий синтаксис шаблонизатора;
встроенные директивы и свойства - v-if, v-for, v-show, v-bind, v-for.

4 Компоненты props и события

Цели занятия:

создавать компоненты Vue; применять SFC;
создавать связанные компоненты;
организовывать обмен данными между компонентами;
объяснить принципы разделения приложений на компоненты;
использовать события в Vue для передачи данных.

Краткое содержание:

почему компонентный подход популярен в современном фронтенде;
как писать компоненты на Vue;
как организовывать взаимоотношения данных и событий между ними.

5 Формы, слоты, фильтры, директивы

Цели занятия:

написать форму для получения данных из шаблона;
объяснить возможности расширения v-model;
применять SFC подход.

Краткое содержание:

примените SFC подход;
организация приложения из многих взаимодействующих

компонентов;
кастомные директивы, v-model, custom filters.

Домашние задания

1 Реализовать несколько форм и списков

Цель: Научится создавать формы и работать с данными в приложении

Какие скиллы тренируем в ДЗ: работа с формами, передача данных между компонентами

- У Вас уже создан репозиторий на Github и проект с использованием Vue-cli

- Определитесь с типом приложения, рекомендуется витрина книжного магазина либо приложение для ведения списка кинофильмов.

API для витрины книжного магазина:

<https://developers.google.com/books/docs/v1/using#request>

Можете выбрать другое приложение CRUD.

Готовые rest API найдете здесь:

<https://github.com/public-apis/public-apis>

Создайте CRUD-приложение с двумя связанными сущностями (например, авторы и книги)

Реализовать страницы:

1. список книг
2. список авторов
3. добавление автора
4. добавление книги

Формы:

1. добавления книг;
2. добавления авторов;
3. поиск книг по нескольким параметрам - название, год выпуска, категория, цена

Прочий функционал:

1. кнопка удаления книг

Модель автора включает: ФИО, ДР, пол, страну проживания, написанные книги,

Модель книги включает: ISBN, название, автор/ список авторов, картинку, краткое описание, теги, цена, категория

Задачи со звездочкой:

(ДЗ можно сдать и без них, но они позволят вам больше практиковаться и нам дать вам больше фидбека):

1 ★ - Использовать CSS-препроцессор (на ваш выбор)

1 ★ - Опубликовать проект (netlify, heroku, github pages, AWS)

2 ★ - Написать unit-тесты

2 ★ - использовать TypeScript

2 ★ - использовать Composition API из Vue v3

6 Использование CSS во Vue

Цели занятия:

использовать альтернативные шаблонизаторы - less/sass/scss, pug.

Краткое содержание:

история вопроса, inline styles, селекторы CSS, порядок применения селекторов, атрибут !important, наиболее важные атрибуты геометрии: display, margin, padding, float, заливки, border vs outline, упомянуть про border-box, работа со шрифтами, выравнивание текста.

1 Vue router

Цели занятия:

использовать роутинг в Vue;
написать конфигурацию роутера;
использовать хуки роутера в компонентах.

Краткое содержание:

что такое URL;
как с ним работать JS;
как добавить навигацию в SPA;
использование роутера для создания многостраничных приложений;
популярные хуки для выполнения функций до или после переключения - vue-router.

Домашние задания

1 Реализовать роутинг в приложении

Цель: Научится применять роутинг в приложении

Какие скиллы тренируем в ДЗ: Vue router, взаимодействие компонентов и роутера, программная навигация, компонент router-link

У Вас уже созданы

- Репозиторий на github
- Проект на vue с использованием vue-cli
- Страницы со списком сущностей (книг/авторов)
- Страницы с добавлением сущностей (книг/авторов)

- добавить роутер (vue-cli/npm)

Реализовать роутинг

- главная страница
- страница со списком книг
- страница со списком авторов
- страница добавления книги
- страница добавления авторов

Реализовать переходы

- с главной страницы на страницы со списком
- со страницы со списком на страницы-формы
- из форм обратно на страницы списков

Прочий функционал

- страницы-редакторы книг и авторов, переходы к ним из списка, возвращение на страницы списков

Задачи со звездочкой:

(ДЗ можно сдать и без них, но они позволят вам больше практиковаться и нам дать вам больше фидбека):

- 1 ★ - Использовать программную навигацию
- 1 ★ - Использовать именованные маршруты
- 1 ★ - Использовать параметры, передаваемые, как props
- 2 ★ - Использовать динамическую подгрузку компонентов
- 2 ★ - Использовать named chunks
- 3 ★ - Добавить анимацию в компоненты/роуты
- 3 ★ - Использовать хуки на определенных маршрутах

2 VueX работа с данными

Цели занятия:

различать преимущества центрального управления стейтом;
добавить State Management приложению Vue;
написать гетер, экшен и мутацию;
писать данные в State из компонентов.

Краткое содержание:

самый популярный стейт менеджер для Vue - Vuex;
почему централизованное управление стейтом популярно в современном фронтенде;
паттерны и используемые методы.

Домашние задания

1 Добавить стейт в приложение

Цель: Научиться использовать vuex знать что такое гетеры, мутации и экшены

Какие скиллы тренируем в ДЗ:

- работа с глобальным стейтом приложения;
- получение данных из стейта в компонентах; вызов экшенов и мутаций из компонентов;

У Вас уже созданы

- Репозиторий на github
- Проект на vue с использованием vue-cli
- Страницы со списком сущностей (книг/авторов)
- Страницы с добавлением сущностей (книг/авторов)
- Имеется рутинг в приложении

Что делаем

- Нужно установить и добавить vuex в приложение (npm i, Vue.use)
- В приложение из предыдущего ДЗ добавить в централизованную работу с данными.
- Необходимо добавить начальный стейт, мутации экшены и геттеры
- Добавить чтение стейта и изменения стейта в компонентах

Задачи со звездочкой:

(ДЗ можно сдать и без них, но они позволят вам больше практиковаться и нам дать вам больше фидбека):

- 1 ★ - использование namespaced Vuex modules
 - 2 ★ - сохранение стейта в LocalStorage
 - 2 ★ - использование Vuex как event-bus (publisher/subscriber)
-

3 Тестирование Vue приложений

Цели занятия:

различать виды тестов;
писать юнит-тесты для Vue приложений;
использовать особенности библиотеки jest.

Краткое содержание:

виды тестирования;
основные библиотеки для тестирования - jest, Phantomjs, cypress.

Домашние задания

1 Добавить тесты к готовому приложению

Цель: Разбираться в видах инструментов для тестов, конфигурировать их и писать сами тесты

Варианты средств расширения vue-приложений;
написание плагинов

Какие скиллы тренируем в ДЗ

- инструменты и методики тестирования
- конфигурация тестов для фронтенд приложения;
- написание unit тестов
- написание e2e тестов

Взять приложение из предыдущего ДЗ

1. Добавить к нему jest и cypress или puppeteer
2. Написать e2e и unit-тесты как минимум для двух компонентов

--

Задачи со звездочкой:

(ДЗ можно сдать и без них, но они позволят вам больше практиковаться и нам дать вам больше фидбека):

- 1 ★ - Добиться покрытия кода тестами свыше 90%
 - 2 ★ - Использовать параметризованное тестирование testEach
 - 2 ★ - Протестировать асинхронный код
-

Цели занятия:

объяснить, что такое MVVM
объяснить особенности change detection в vue;
применять подход Vue.set для обновления сложных
моделей.

Краткое содержание:

анализ работы computed, watchers;
методы инстанса Vue (.set / .delete / .watch);
свойства: computed, watchers.

Домашние задания**1** Реализовать несколько форм и списков

Цель: Научится создавать формы и работать с
данными в приложении

Какие скиллы тренируем в ДЗ: работа с формами,
передача данных между компонентами

1. Создать репозиторий на github
2. Сделать проект на vue с использованием vue-cli
3. Создать CRUD приложение для по типу книжный магазин. В приложении будет 2 типа сущностей: авторы и книги.
4. Сделать MR и прислать ссылку

Реализовать страницы:

1. список книг
2. список авторов
3. добавление автора
4. добавление книги

Формы:

1. добавления книг;
2. добавления авторов;
3. поиск книг по нескольким параметрам - название, год выпуска, категория, цена

Прочий функционал:

1. кнопка удаления книг

Модель автора включает: ФИО, ДР, пол, страну проживания, написанные книги,

Модель книги включает: ISBN, название, автор/

список авторов, картинку, краткое описание, теги, цена, категория

Задачи со звездочкой:
(ДЗ можно сдать и без них, но они позволят вам больше практиковаться и нам дать вам больше фидбека):

1 ★ - Использовать CSS-препроцессор (на ваш выбор)

1 ★ - Опубликовать проект (netlify, heroku, github pages, AWS)

2 ★ - Написать unit-тесты

2 ★ - использовать TypeScript

2 ★ - использовать Composition API из Vue v3

5 Продвинутый JavaScript

Цели занятия:

без прямой привязки в Vue разобрать сложные и набирающие популярность концепты языка.

Краткое содержание:

proxies, generators, ES modules.

1 Интеграции Vue p1: GraphQL, OAuth 2.0

Цели занятия:

объяснить возможности GraphQL;
различать работу через REST vs GraphQL;
использовать техники авторизации веб приложений;
добавлять GraphQL к Vue приложению;
добавление Vue в legacy-проекты.

Краткое содержание:

несколько популярных интеграций для работы с данными / авторизации.

2 Интеграции Vue p2 - FireBase, web sockets

Цели занятия:

объяснить возможности Firebase;
использовать техники авторизации веб приложений;
добавлять Firebase к Vue приложению;
добавлять web sockets
добавление Vue в legacy-проекты.

Краткое содержание:

продолжения обзора популярных интеграций для работы с данными / авторизации.

3 SSR с Nuxt

Цели занятия:

объяснить преимущества SSR;
настроить prerendering страниц frontend приложению Vue;
создать с нуля Nuxt в приложение с Vue;
добавить страницы и роутер в Nuxt приложении.

Краткое содержание:

разбор Nuxt фреймворк для написания SSR приложений на Vue.

4 Vue3 работа с библиотеками

Цели занятия:

Рассмотрим функционал который добавляют популярные библиотеки для работы с composition API

Краткое содержание:

useState, useRouter, vueuse library

5 Typescript

Цели занятия:

объяснить особенности и основные преимущества typescript;
добавить typescript к существующему проекту на Vue;
писать компоненты и сервисы Vue приложение на typescript;
создать интерфейсы, модули и .td файлы.

Краткое содержание:

преимущества TS;
создание проекта;
добавление основных решений с использованием Typescript, vue-property-decorator.

6 Организация кода и паттерны Vue

Цели занятия:

структурировать файлы и код в приложении по мере роста кодовой базы: service pattern, fabric pattern, axios.
- provide/inject

Краткое содержание:

структурирование файлов и код в приложении по мере роста кодовой базы;
что такое service pattern, fabric pattern, axios, <component> tag.
provide/inject

7 Консультация по дипломному проекту

Цели занятия:

- обсудить идеи студентов для финального проекта и возможные этапы реализации
- предложить готовые направления для разработки итогового проекта.

Краткое содержание:

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

Домашние задания

1 Описать планируемый проект

Цель: - сформулировать над чем будем работать в проекте модуле №4
- описать технические детали реализации
- описать мотивацию и customer value проекта
- уточнить по объём работы - и этапы реализации

8 **Десктоп приложения с Electron**

Цели занятия:

применять инструменты разработки приложение для десктопа с использованием Vue;
создать приложение для десктоп с нуля;
добавить элементы интерфейса;
сохранить пользовательские настройки;
получить данные по сети.

Краткое содержание:

создание приложения для десктопа, используя Vue + electron.

Домашние задания

1. Выбрать одну из 3х технологий (Nuxt или GraphQL или Electron) и реализовать мини проект.

Цель: Создавать приложения для на десктопа или мобильного на Vue.

Какие скиллы тренируем в ДЗ:

- создавать кросс-платформенные приложения с Vue; особенности сборки приложения с Electron; работа с SSR/Nuxt; работа с GraphQL.

1. Создать репозиторий на github
 2. Создать новый проект для мобильного или для десктопа или веб-проект для Nuxt - в зависимости от вашего выбора
 3. Добавить ряд страниц
 4. Добавить бизнес-логику
 5. Создать МР и прислать ссылку
-

9 **Инструменты современной разработки, cloud, CI/CD**

Цели занятия:

рассмотреть популярные технологии для работы с IT проектами - от планирования до деплоя.

Краткое содержание:

npm publish, git, trello, github actions, figma, postman, heroku, netlify, CI/CD

10 **Анимация Vue**

Цели занятия:

добавить анимацию к спискам;
добавить анимацию между роутами приложения;
создать собственную анимацию;
управлять анимацией из свойств CSS;
знать популярные библиотеки готовых эффектов.

Краткое содержание:

встроенные решения для анимации в Vue;
примеры и выполнение нескольких самостоятельных работ;
transition tag, CSS animations, keyframes, besier curve.

Домашние задания

1 Добавить стейт и роутер в приложение

Цель: Подключать и сконфигурировать роутер, добавлять чтение параметров адреса внутри компонентов вызвать хуки при переходе на определенный route.

Научиться использовать vuex знать что такое гетеры, мутации и экшены

Какие скиллы тренируем в ДЗ:

- работа с глобальным стейтом приложения; получение данных из стейта в компонентах; вызов экшенов и мутаций из компонентов;
- работа с vue-router, добавление вложенных роутов, знание хуков роутера

Использовать репозиторий из прошлого ДЗ.

1. Добавить роутер (npm i, Vue.use)
2. Сконфигурировать адреса для соответствующих страниц
- 2а. использовать Composition API / Vue v3
3. Нужно установить и добавить vueх в приложение (npm i, Vue.use)
4. В приложение из предыдущего ДЗ добавить в централизованную работу с данными.
5. Необходимо добавить начальный стейт, мутации экшены и геттеры
6. Добавить чтение стейта и изменения стейта в компонентах

Задачи со звездочкой:

(ДЗ можно сдать и без них, но они позволят вам больше практиковаться и нам дать вам больше фидбека):

- 1 ★ - Добавить анимацию в компоненты/роуты
- 1 ★ - Использовать CSS-препроцессор (на ваш выбор)
- 1 ★ - Опубликовать проект (netlify, heroku, github pages, AWS)
- 2 ★ - Написать unit-тесты
- 2 ★ - использовать TypeScript

11 Web components + Vue

Цели занятия:

ориентироваться в веб спецификациях Custom Elements и Shadow DOM;

Краткое содержание:

Custom Elements:

Standalone Elements;

vue-CLI - сборка в виде веб-компонента

12 QA / Code review

Цели занятия:

анализировать код проекта;
предложить улучшения архитектуры.

Краткое содержание:

разбор сделанных ДЗ за все предыдущие время -
ответы на вопросы.

13 Функциональное программирование и продвинутый Vue

Цели занятия:

знать отличие и преимущества функционального
программирования;
переиспользовать код с использованием
функциональных паттернов;
применять функциональный подход в Vue.

Краткое содержание:

НОС;
отличия функционального программирования;
сильные стороны функционального
программирования;
написание функционального код на JS.

1 Выбор темы и организация проектной работы

Цели занятия:

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

Краткое содержание:

правила работы над проектом и специфика проведения итоговой защиты;
требования к результату проекта и итоговой документации.

Домашние задания

1 Проект

Цель: Какие навыки тренируем в ДЗ:

- планирование проекта
- декомпозиция реализации
- выбор технологий и архитектуры
- оценка сложности проекта

1. определиться что хотим получить в итоге (отсечь лишнее чтобы сделать проект выполняемым);
 2. план работ;
 3. список возможных сложностей и вещей которые могут повлиять на реализацию
 4. декомпозиция проекта на модули которые будут реализованы;
-

2 **Консультация по проектам и домашним заданиям**

Цели занятия:

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

Краткое содержание:

вопросы по улучшению и оптимизации работы над проектом;

затруднения при выполнении ДЗ;

вопросы по программе.

3 **Защита проектных работ**

Цели занятия:

защитить проект и получить рекомендации экспертов.

Краткое содержание:

презентация проектов перед комиссией;

вопросы и комментарии по проектам.