

Описание функциональных характеристик программного обеспечения и информация, необходимая для установки и эксплуатации программного обеспечения

Оглавление

Словарь терминов	3
О Course Editor	4
Бизнес задачи	4
Описание Course Editor	4
Как работает Course Editor	5
Пользователи Course Editor	5
ПОЛЬЗОВАТЕЛЬСКИЙ ФУНКЦИОНАЛ COURSE EDITOR	6
Главная страница	6
Создание структуры курса	7
Редактирование структуры курса	9
Создание урока	9
Настройки	10
Публикация курсов	11
Превью курсов	11
Инструкция по установке ПО «Course Editor»	12
Системные требования	12
Описание дистрибутива	12
Настройка API	12
Настройка Front окружения	13
Установка Node.js	13
Установка Vuetify, Lodash, Sentry	14
Запуск Course Editor с сервера	14

Словарь терминов

Термин	Определение
CourseEditor, Сервис	Конструктор для разработки электронных курсов
Дистанционное обучение	Способ обучения на расстоянии, при котором преподаватель и обучающиеся физически находятся в различных местах Средство обучения, использующее аудио, видеотехнику, интернет и спутниковые каналы связи
Курс	Электронный образовательный материал определенной тематики
Глава (Раздел)	Контентная единица курса, внутри которой хранятся сущности определенной тематики
Регистрация	Процесс авторизации в Course Editor
Ученик	Человек, имеющий доступ к курсу, для прохождения обучения
Пользователь	Человек, работающий в Course Editor
Контент	Информация, расположенная внутри сущности или курса
Структура курса	Совокупность всех сущностей курса любых уровней вложенности
TinCan API, xAPI	Универсальный стандарт дистанционного обучения, который предполагает доставку и хранение учебных активностей (стейтментов) в хранилище учебных активностей (Learning Record Store)
Информационное окно	Всплывающее окно, содержащее текстовое описание внутри
Навигационная панель	Область на странице, где расположены ссылки на все основные разделы и страницы
Практический контент, Практические сущности	Тип сущности «Практика» — кейсы, тесты, тренажеры–практика
Лонгрид	Страница с нефиксированной высотой, содержащая в себе текстовые, интерактивные, мультимедийные, интерактивные и проверочные элементы
LMS (СДО)	Система дистанционного обучения, в которой ученик проходит обучение на электронном курсе
События (факт, активность, стейтмент)	Заданный формат данных, который хранит всю информацию о каждом действии обучающихся в курсе. Данные стейтмента не могут быть изменены

О Course Editor

Course Editor это конструктор электронных курсов. Его задача — упростить и ускорить разработку курса.

Используя Course Editor для сборки курсов, появляется возможность создавать такие элементы как лонгрид, состоящий из текстовых блоков, картинок, роликов, интерактивных механик и тестов, или любой из этих элементов в отдельности.

Любой курс, созданный в Course Editor, можно экспортировать в современные форматы TinCan, xAPI и SCORM, которые гарантировано будут работать с вашей СДО. Есть возможность настройки прямой публикации курса в систему дистанционного обучения.

Сервис позволяет быстро и просто создавать обучающие курсы, экономя время на разработку и трудозатраты.

Бизнес задачи

Сервис разработки электронных курсов выполняет следующие бизнес-задачи:

1. Позволяет собирать обучающие дистанционные курсы без затрат на разработку и дизайн;
2. Быстрое обучение сотрудников компании;
3. Дизайн-шаблоны электронных курсов могут разрабатываться под требования компании;
4. Сбор BigData для аналитики путем использования стандарта TinCan API в дистанционных курсах.

Описание Course Editor

Сервис является онлайн-платформой по созданию и редактированию контента в обучающих электронных курсах. Пользователь может использовать сервис как рабочее пространство для составления контента онлайн курса, а может использовать свои материалы, для переноса обучения в электронную версию. Теория может быть подкреплена проверкой знаний.

Готовую версию дистанционного курса пользователь может экспортировать в необходимом для него стандарте, либо предоставляет ссылку на просмотр для согласования проекта.

Сервис состоит из нескольких модулей:

- Главная страница со списком курсов
- Модуль создания структуры
- Модуль создания урока
- Модуль настройки курса
- Модуль превью курса
- Модуль публикации курса

При разработке онлайн курса пользователь может использовать вышеперечисленные модули, необходимые для его целей. После использования модуля публикации курса и получения архива созданного курса, обучение может быть запущено.

Как работает Course Editor

С технической точки зрения, основой курса, созданной в Course Editor, является структура курса в JSON формате, которую задает пользователь. К этой основе, дополнительными модулями привязан фундаментальный функционал, такой как переход по элементам структуры, отправка статистики и визуальный интерфейс курса.

Визуальный интерфейс – это шаблон со стилями, на который накладывается структура собранного онлайн курса. Выбирая для курса разные темы при публикации, на выходе можно получить программы с разным дизайном.

Отправка статистики основана на передаче событий из курса в систему дистанционного обучения (СДО). В зависимости от выбранного формата, количество генерируемых данных может меняться, так как например для SCORM события генерируются на курс ([см. Публикация курсов](#)), в то время как для TinCan API события генерируются на каждый элемент структуры, что увеличивает суммарное количество событий, относительно SCORM.

Пользователи Course Editor

В Сервисе создана роль – разработчик (пользователь). Это человек, который занимается сборкой курса.

Разработчик имеет доступ ко всем страницам Course Editor, в пределах своей компании.

ПОЛЬЗОВАТЕЛЬСКИЙ ФУНКЦИОНАЛ COURSE EDITOR

Сервис Course Editor разработан в едином стиле. Каждая страница конструктора интуитивно понятна, а для работы с сервисом не требуется дополнительного обучения. Конструктор включает в себя следующие страницы:

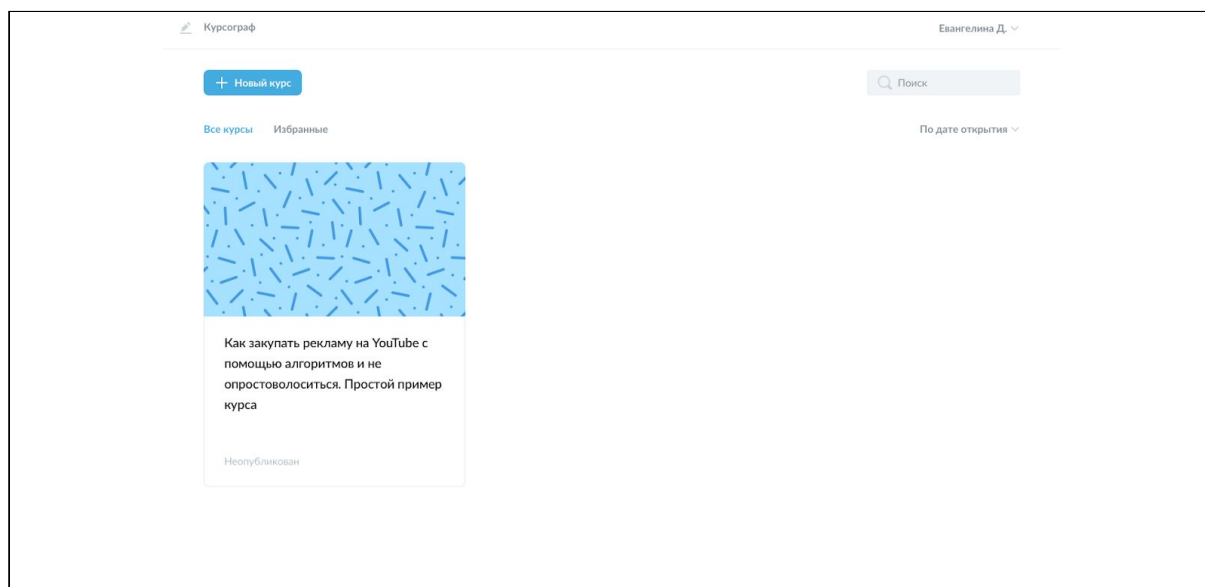
- Главная страница;
- Избранные курсы;
- Создание и редактирование курса;
- Профиль пользователя.

Главная страница

Начальная страница Course Editor выводит полный перечень курсов, разработанных в Course Editor. У пользователя на текущей странице есть два пути: либо редактировать ранее разработанные курсы, либо создать новый курс по клику на кнопку “Новый курс”.

По желанию, разработанные курсы могут быть сгруппированы в избранные курсы. Помимо этого, список курсов может быть отфильтрован по нужному параметру.

Также, для удобства пользователя, на главной странице организован поиск по названию курсов и по элементам структуры.



На странице выведен список курсов, где каждый курс имеет уникальное название и

краткую статистическую информацию:

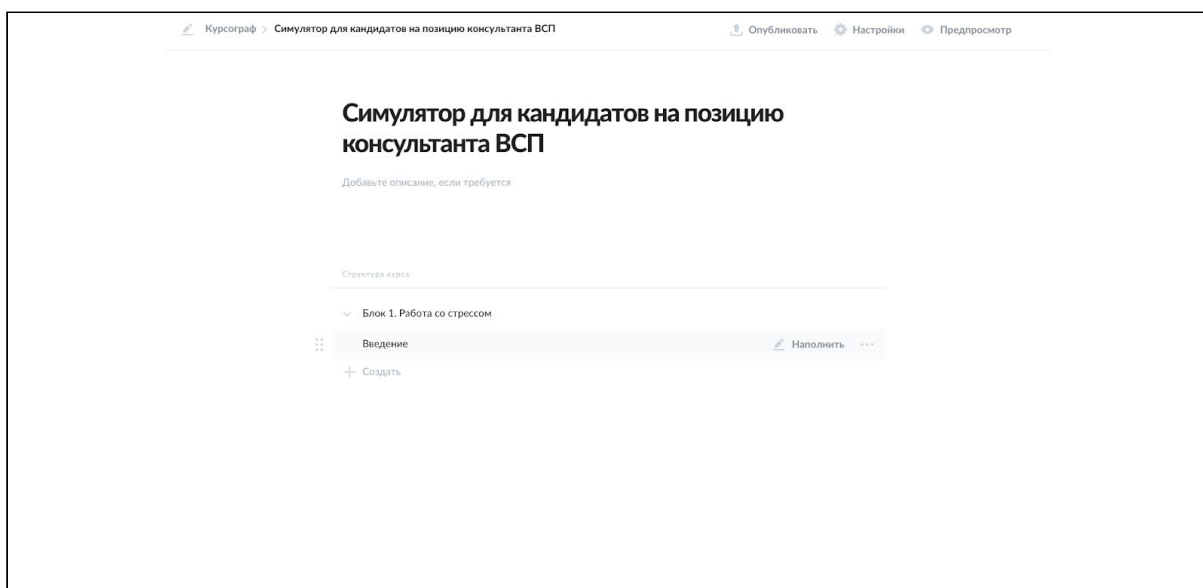
- статус публикации;
- дату создания.

В правом верхнем углу экрана расположен профиль пользователя, он зафиксирован и доступен на главной странице. По клику на него можно выйти из Сервиса.

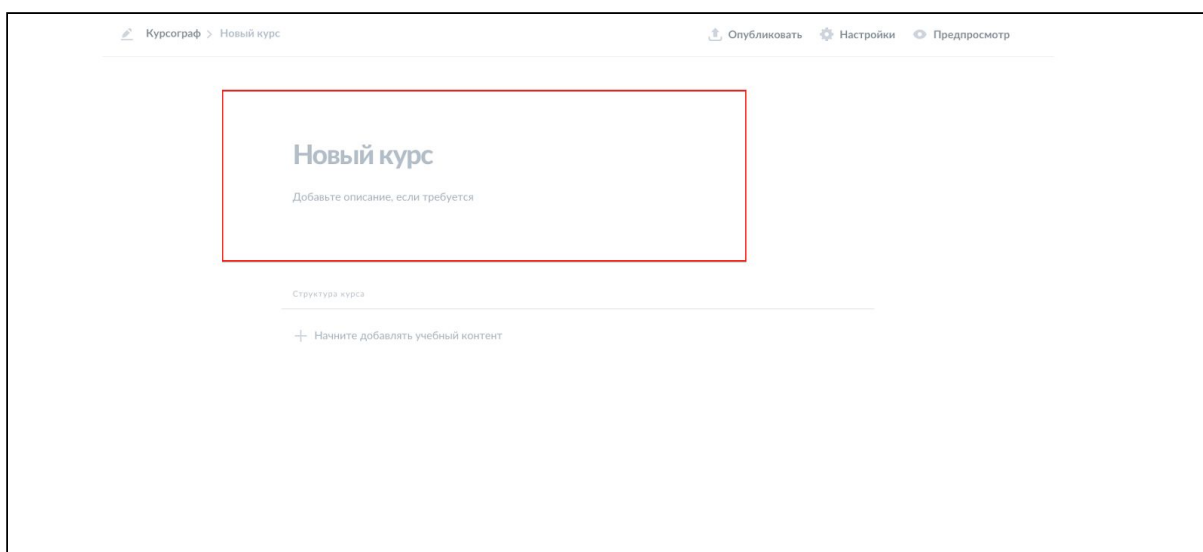
Создание структуры курса

При создании нового курса с главной страницы пользователь попадает на экран редактирования основной информации курса. Основной информацией курса считается название курса, его описание и структура курса.

Создание структуры не ограничено уровнями вложенности, это позволяет создавать курсы различного уровня сложности.



У пользователя есть возможность задать название и описание своего курса по клику на соответствующие области в конструкторе.



Следующим шагом может стать разработка структуры курса. Интерфейс

показывает последовательность действий, однако не ограничивает пользователя в порядке его действий.

Название и описание курса вынесены в быстрый доступ, так как они являются основными и часто используемыми при создании курса.

Правила создания структуры:

- **Создание главы/подглавы**

Так как вложенность структуры курса в Сервисе не ограничена, на верхний уровень вынесена сущность типа “Глава/Подглава”. Это некий контейнер или папка для контента курса. Главу/подглаву наполнить контентом нельзя. Для создания главы\подглавы, необходимо нажать на клавиатуре пользователя клавишу “Tab”.

- **Создание урока**

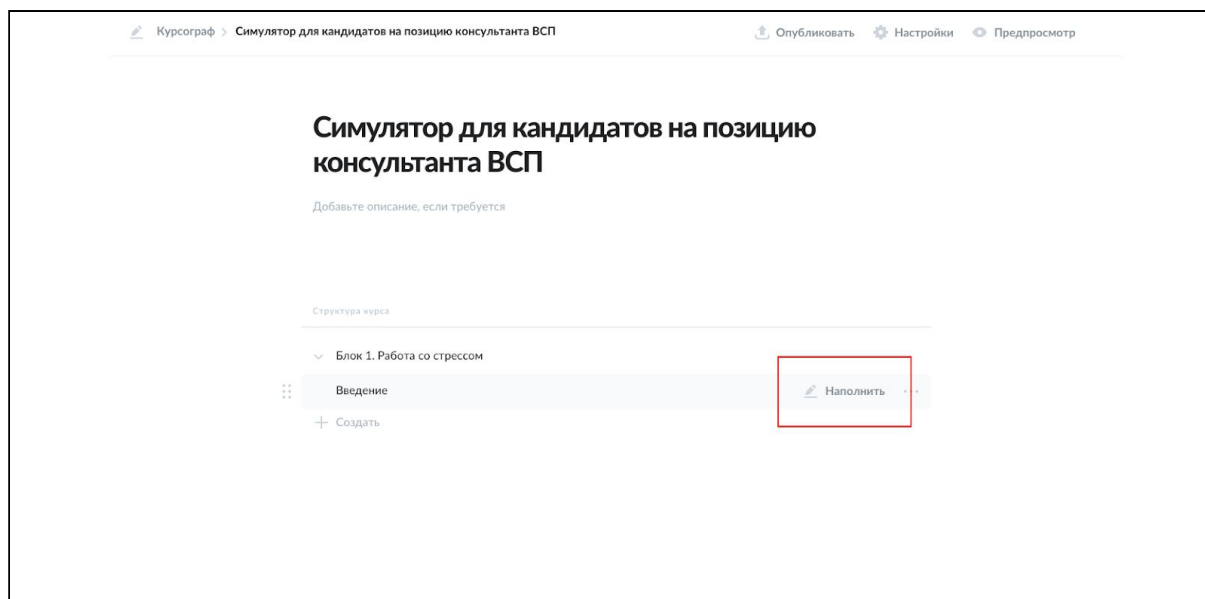
Сделать элемент структуры уроком можно с помощью клавиши клавиатуры “Shift+Tab”.

- **Манипуляции над элементами структуры**

Созданные элементы структуры (глава/подглава или урок) можно перемещать в структуре с помощью драг энд дропа, копировать в рамках одной структуры или переносить в другой курс.

- **Наполнение урока**

Наполнить контентом можно только элемент типа “Урок”, с помощью клика на кнопку “Наполнить”, отображающуюся возле структурного элемента курса.



Интерфейс сервиса подсказывает пользователю последовательность действий, но не ограничивает его в выборе. Пользователь может как создавать полностью структуру после чего наполнять ее, так и создавать по одному элементу в структуре, последовательно наполняя его контентом.

Редактирование структуры курса

В созданном дистанционном курсе можно изменять контент по усмотрению пользователя.

По клику на курс с главной страницы Сервиса откроется страница со структурой курса.

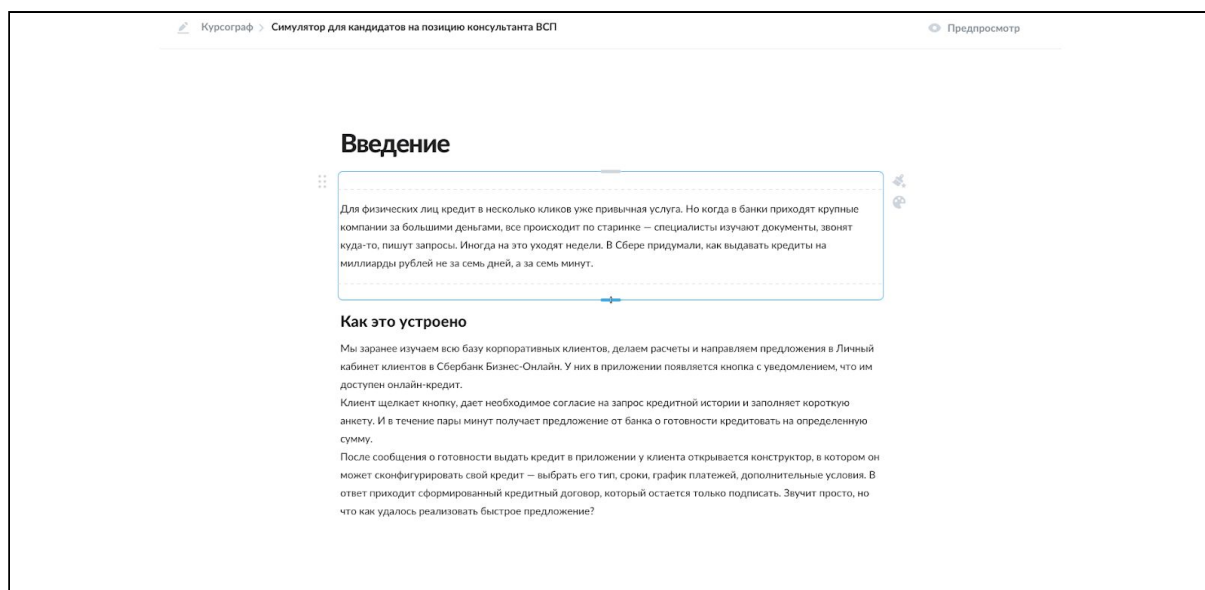
По клику на заполненные поля с названием и описанием курса, пользователю предоставляется возможность изменить контент. Аналогичным образом меняется название структурных элементов дистанционного курса.

Пользователь может изменить последовательность элементов в структуре или удалить ранее созданный элемент (см. [Манипуляции над элементами структуры](#))

Создание урока

На странице наполнения урока контентом пользователь может собрать несколько типов уроков. Тип урока будет определен автоматически системой, в зависимости от типов блоков (см. [Текстовый блок](#)). Все типы уроков приведены ниже:

- Лонгрид — страница с нефиксированной высотой, содержащая в себе текстовые, интерактивные, мультимедийные, интерактивные и проверочные элементы;
- Мультимедиа — страница с фиксированной высотой, состоящая из ролика или аудио;
- Проверка знаний — страница с фиксированной высотой, состоящая либо из теста, либо из упражнения.



Тип урока — лонгрид

Пользователь может создавать любые из вышеперечисленных типов уроков, исходя из своих задач, собирая в конечном итоге курс из теоретических и практических заданий.

Каждый урок состоит из контентных блоков, с которыми работает пользователь. CourseEditor, в свою очередь, автоматически распознает, какой тип контента был использован, и определяет тип урока (см. [Типы уроков](#)).

Внутри одного урока может быть неограниченное количество блоков. Также для пользователя нет ограничений на использование разных типов блоков внутри одного урока.

Перечень всех типов блоков приведен ниже:

- **Текстовый блок**
Блок, состоящий из текста.
- **Блок с картинкой**
Блок, состоящий из картинки и текста.
- **Блок с мультимедиа**
Блок, состоящий или из видео, или из аудио.
- **Блок цитата**
Блок, состоящий из текста, оформленного в виде цитаты.
- **Блок со списком**
Блок, состоящий из списка текста.
- **Блок с галереей**
Блок, состоящий из набора картинок, сменяющиеся по клику, в виде карусели.
- **Интерактивный блок**
Блок состоящий из интерактивных элементов, так как драг энд дроп или диалоговый тренажер.
- **Блок разделитель**
Блок состоящий из разделительной полосы между остальными блоками.
- **Блок проверки знаний**
Блок состоящий из тестов.

Для каждого типа блока на стороне разработчиков Course Editor, сформированы шаблоны, которые пользователь может применять в работе с Сервисом. Для уточнения подробностей о видах и условиях использования шаблонов, необходимо обратиться к менеджеру проекта.

Настройки

После наполнения урока контентом пользователь может настроить конфигурации разрабатываемого курса. Параметры, по которым может происходить настройка прописаны в таблице

Настройка	Описание
Тема курса	Изменение дизайнерского шаблона курса, который меняет внешний облик онлайн программы
Доступность копирования текста	Разработчик курса может запретить или разрешить копирование текста ученику
Расположение меню	Разработчик курса может расположить справа или слева меню курса, который будет проходить ученик

Основной цвет курса	Разработчик курса может установить основной цвет курса, который будет проходить ученик. Основной цвет отображается на всех главных элементах курса, например, структура курса.
Вспомогательный цвет курса	Разработчик курса может установить вспомогательный цвет курса, который будет проходить ученик. Вспомогательный цвет отображается в элементах, которые ... например, в меню курса.
Процент прохождения курса	Разработчик курса может установить, какое количество баллов должен набрать ученик для успешного прохождения курса

Когда все настройки пользователем установлены, курс готов для публикации.

Публикация курсов

При экспорте электронного курса, пользователь выбирает формат передачи статистики: SCORM 2004 или TinCan API.

SCORM 2004

При использовании формата SCORM, электронный курс, собранный в CourseEditore, будет передавать в LMS факт:

- прохождения курса;
- место, на котором остановился пользователь;
- текущий процент пользователя.

TinCan API

При использовании формата TinCan API, электронный курс, собранный в CourseEditor, будет передавать в LMS следующие события:

- Вошел (на каждый урок)
- Вышел (на каждый урок)
- Прошел (на каждый урок)
- Запустил (на каждый ролик)
- Перемотал (на каждый ролик)
- Поставил на паузу (на каждый ролик)
- Посмотрел (на каждый ролик)
- Вошел в вопрос (на каждый вопрос теста)
- Ответил на вопрос неверно (на каждый вопрос теста)
- Ответил на вопрос верно (на каждый вопрос теста)

Превью курсов

В любой момент разработки курса пользователь имеет возможность просмотреть текущее состояние курса. Это значит, что он сможет увидеть и использовать в полноценном режиме:

- плеер курса (то есть проигрывание курса) и все окна плеера (навигация по курсу);

- структуру курса;
- каждый элемент курса.

Также, в Сервисе есть возможность простым переключением кнопок менять размер экрана, тем самым проверять, как будет выглядеть курс для ученика на мобильном устройстве, на айпаде и на мониторе компьютера.

Инструкция по установке ПО «Course Editor»

Системные требования

- MySQL 5.7
- PHP 7.4.4 с установленными расширениями:
 - bcmath
 - ctype
 - fileinfo
 - json
 - mbstring
 - openssl
 - pdo
 - tokenizer
 - xml

Описание дистрибутива

Дистрибутив состоит из двух приложений:

- Веб-интерфейс
- API

Настройка API

Приложение предоставляет возможность получать данные для Веб-интерфейса, а также формирует предпросмотры создаваемых курсов и экспортирует создаваемые курсы.

1. **Распаковать код** в директорию `/srv/vhosts/api` (можно отличную от указанной, далее в примерах будет использоваться именно этот путь до файлов).
2. **Сменить владельца файлов** на `www-data` командой `chown -R www-data:www-data /srv/vhosts/api`.

3. **Вернуться в директорию** `/srv/vhosts/api` и скопировать файл для настройки окружения командой: `cp .env.example .env`;
4. **Открыть для редактирования файл** `/srv/vhosts/api/.env` и произвести настройку подключения к MySQL-серверу:
 - Директива `DB_HOST` отвечает за адрес MySQL-сервера;
 - Директива `DB_PORT` отвечает за настройку порта по которому осуществляется подключение к MySQL-серверу;
 - Директива `DB_DATABASE` отвечает за название базы данных;
 - Директива `DB_USERNAME` отвечает за имя пользователя под которым подключаться к MySQL-серверу;
 - Директива `DB_PASSWORD` отвечает за пароль пользователя.
5. **Сгенерировать уникальный ключ шифрования Cookies:** `php artisan key:generate`;
6. **Выполнить команду на запуск миграций таблиц в MySQL:**
 - `php artisan migrate`
 - `php artisan migrate --path=app/Auth/database/migrations`
7. **Открыть для редактирования файл** `/srv/vhosts/api/.env` и произвести дополнительные настройки:
 - Директива `APP_URL` отвечает за URL адрес API
 - Директива `JWT_TTL` отвечает за количество минут, в течение которых токен авторизации будет валиден
 - Директива `DELETING_OLD_PREVIEWS_DELAY_IN_HOUR` отвечает за промежуток времени в часах от создания предпросмотра курса до его удаления

На этом **настройку API** можно считать **завершенной**.

Настройка Front окружения

Установка Node.js

Для этого на сайте <https://nodejs.org/en/> скачать дистрибутив и установить его на рабочий компьютер. При установке Node.js автоматически будет установлен менеджер пакетов NPM.

В командной строке перейти в корневую папку проекта, выполнить команду:

```
npm install
```

Установка Vuetify, Lodash, Sentry

В командной строке перейти в корневую папку проекта, выполнить команду:

```
npm run build
```

Результатом данной команды должна стать папка dist, включающая в себя готовые для загрузки на сервер файлы

Запуск Course Editor с сервера

В командной строке перейти в корневую папку проекта, выполнить команду:

```
npm run serve
```

Результатом работы должен стать запущенный в браузере адрес

<http://localhost:8080>