# Описание

Для работы с react необходимо установить node js. Через команду «npx create-react-app WebLesson» создастся новый проект со стандартной структурой create react app.

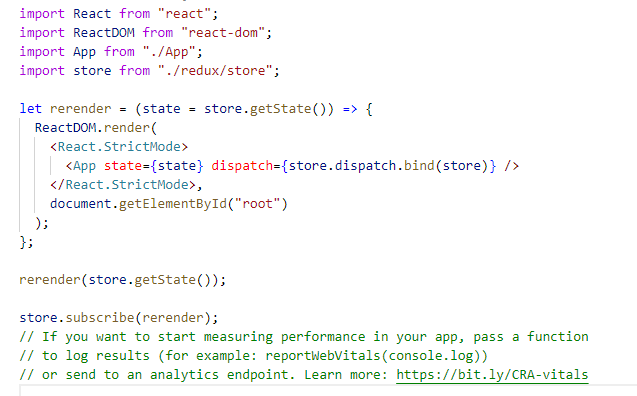
Вся разработка ведется в папке «src». Входной файл «index.js», запускает react и делает первичную отрисовку «app.js» (рис. 3.1).

Рис. 3.1 Содержимое файла «index.js»

К файлу «app.js» можно подключить css файл, в котором будут прописаны стили отображения всех элементов приложения. «app.js» по сути большая компонента которая содержит в себе несколько поменьше (рис. 3.2).



Рис. 3.2 Содержимое компоненты «app.js»

Все компоненты кроме app будут иметь расширение «jsx» вместо обычного «js». Это позволит удобно писать jsx разметку в среде разработки visual studio code. IDE будет понимать когда включать синтаксис html а когда обычный js. Все такие компоненты будут находиться в папке «moduls»

Глобально приложение можно разделить на 2 части: меню и контент. Соответственно можно создать 2 компоненты «menu.jsx» и «content.jsx».

Компонент «menu.jsx» можно поделить на меню с разделами «menuTheme.jsx» и главами по разделам «menuStep.jsx». их содержимое генерируется на ходу, в зависимости от state.

Для того чтобы добавить возможность свободно расширять содержимое программы, добавляя новые разделы и подразделы, необходимо вывести все данные о них в отдельный файл. Этот файл называется «theme1.json» и находится он в папке public, для того чтобы удобно было к нему добраться после работы webpack в конечном electron приложении.

Библиотека react предназначена для отрисовки страницы, за ее наполнение отвечает сторона бизнес логики (BLL). Суть бизнес логики в том, чтобы взять данные из файла «theme1.json» и перенести в state, отвечающий за состояние страницы. В react существует внутренний локальный state, но есть проблема с их использованием при работе с функциональными компонентами. Для более удобной работы со state используется библиотека «redux».

Все дополнительные инструменты скачиваются и интегрируются в проект с помощью пакетного менеджера «npm».

Необходимо получить данные из «theme1.json» и поместить их в компоненты «menu» и «content». Так как компонента - это функция которая возвращает jsx разметку, то это значит, что туда можно передать дополнительные параметры «props». В props’ах можно передать текущее состояние компоненты «state». В дальнейшем программа при взаимодействии с пользователем должна будет менять свой внешний вид, а значит, что react должен перерисовать изменившиеся компоненты с обновленным state. На самом деле при взаимодействии с элементом меняется не сам компонент, а его state. И уже в зависимости от state, react сам перерисовывает элементы. State меняется с помощью dispatch.

Итак, в компонент «menu» параметром приходят state со всеми данными. Задача данной компоненты отрисовать элементы на экране (рис. 3.3).

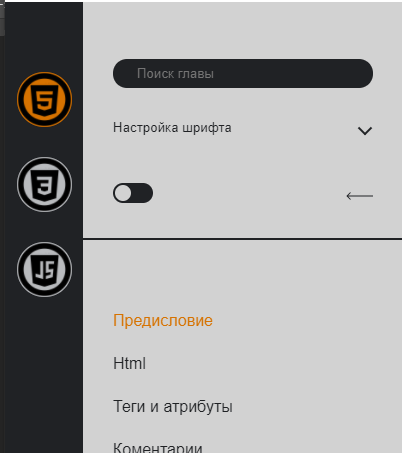


Рис. 3.3 Итог отрисовки компоненты «menu.jsx»

Слева можно увидеть разделы, а слева поле для поиска, шторку для изменения параметров шрифта, кнопку для скрытия меню, переключатель для изменения цветовой темы и внизу подразделы выбранного раздела.

За отрисовку разделов отвечает компонента «MenuTheme». За отрисовку подразделов отвечает компонента «menuStep».

Все input’ы и кнопки, запускает dispatch, который принимает в параметры объект с обязательным свойством «type». Благодаря этому свойству программа понимает, какой сценарии изменения state, нужно запустить. Все сценарии содержаться в reducer’ах. Reducer’ов всего 2, как и глобальных компонентов.

Пример работы метода dispatch можно рассмотреть на кнопке скрытия меню. По нажатию на данную стрелочку выполняется функция (рис. 3.4).

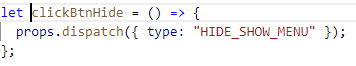


Рис. 3.4 Функция кнопки скрытия меню

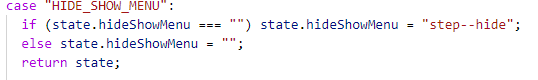
Тем временем, в reducer меню «menuReducer.js» выполняется следующий сценарий (рис. 3.5).

Рис. 3.5 Пример изменения state

По такому же принципу работают все интерактивные элементы.