**ToDo приложение с использованием MVC**

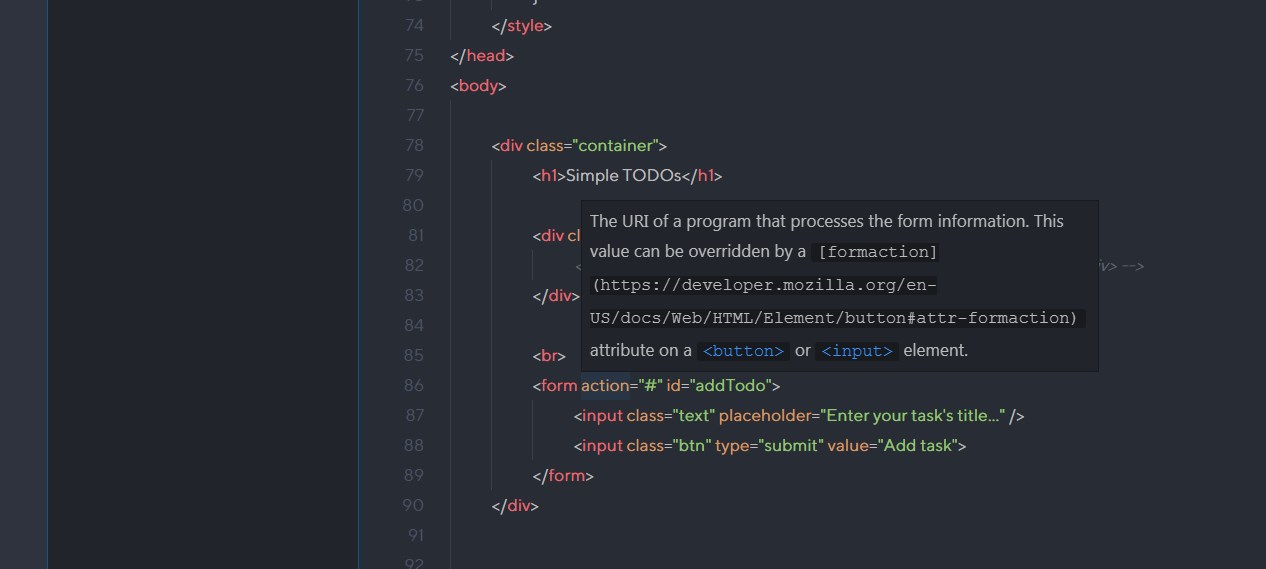
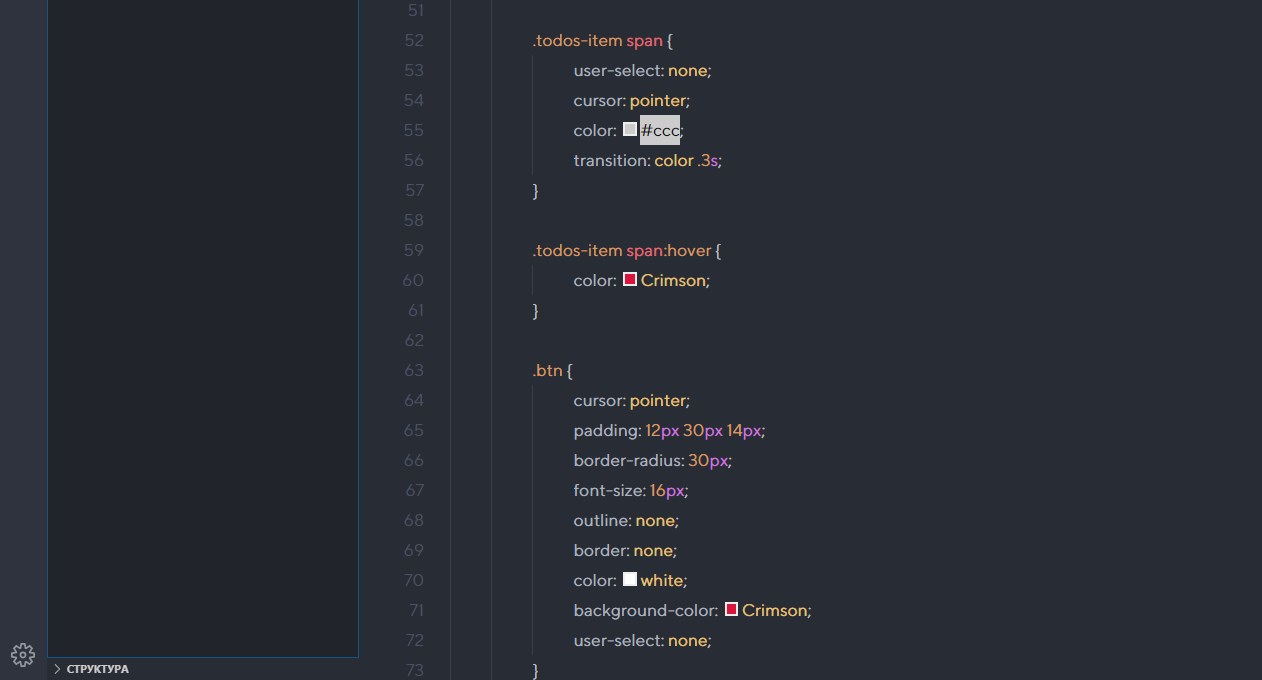
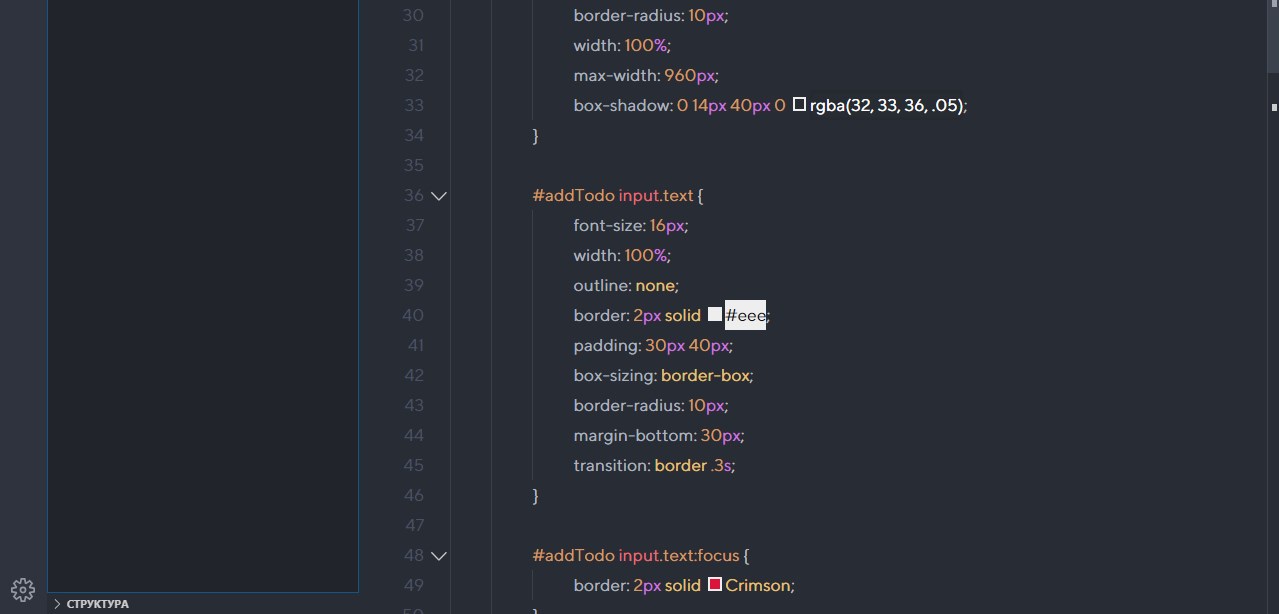
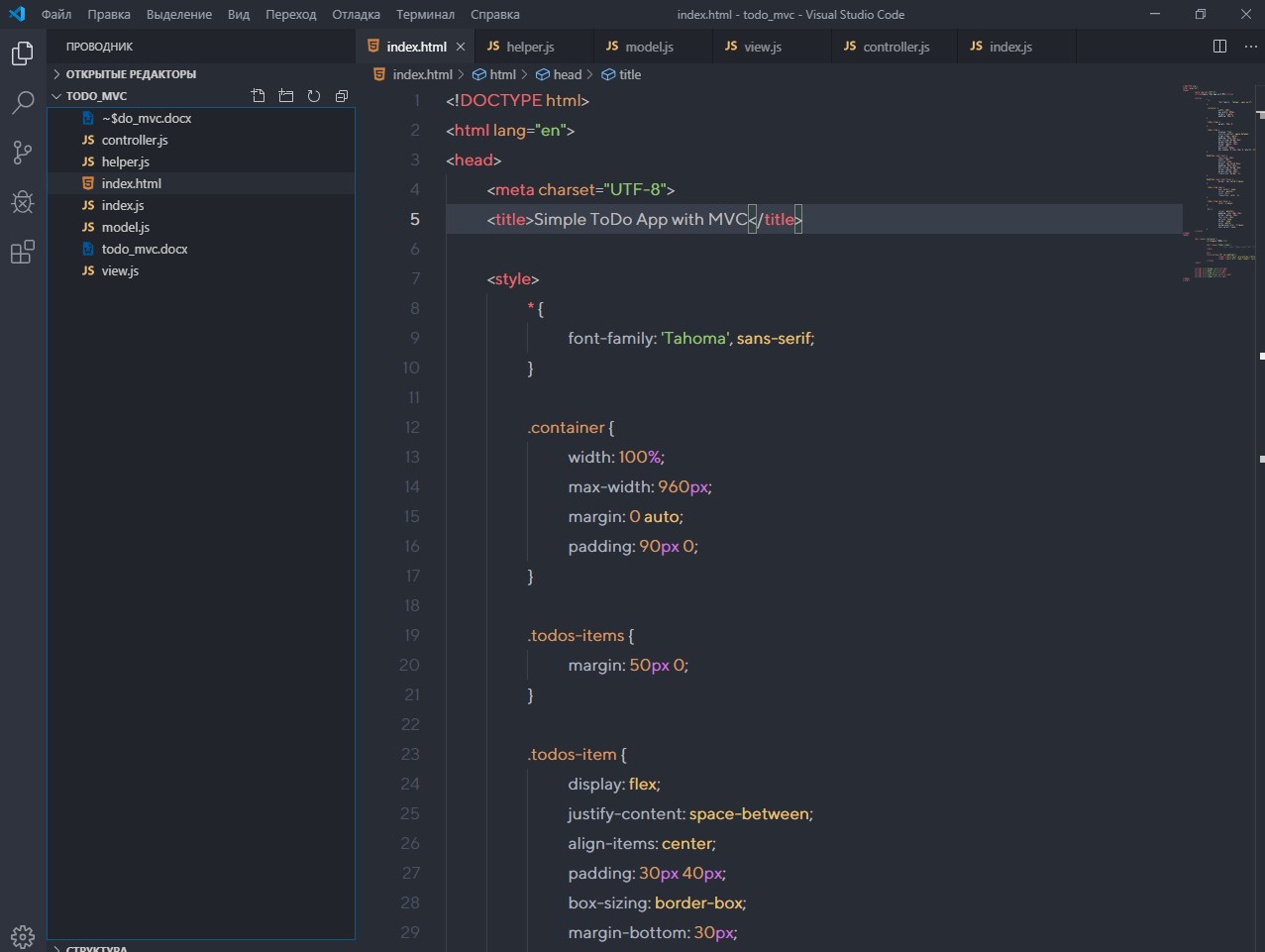
**ToDo является популярным и простым приложением и очень хорошим примером для изучения программирования. Основной идеей приложения является: возможность планировать свой день созданием списка задач на день в приложении. Задача создаётся заполнинием и отправкой формы добавления. Задачу можно редактировать (поменять название), выполнить (пометить флажком) или удалить. Также в приложении будет сортировка и фильтрация задач по определённым критериям (сортировка: сначала новые, сначала старые; фильтрация: все, активные, выполненные).**

**Конечный результат:**

**Тут скрин**

**В архитектурной основе приложения используется подход MVC (Model View Controller). Основной идеей данного подхода является разделение бизнес логики (BL) или данных от пользовательского интерфейса (UI) и управление этими частями с помощью контроллера. MVC не является чёткой инструкцией оформления кода, это лишь идея о том как можно разделить данные от интерфейса. В данной реализации модель и представление не будут знать не друг о друге не о контроллере, а взаимодействие и передача данных осуществляться контроллером и поведенческим паттерном pub/sub.**

**Для начала я создал стартовую разметку и css оформление.**

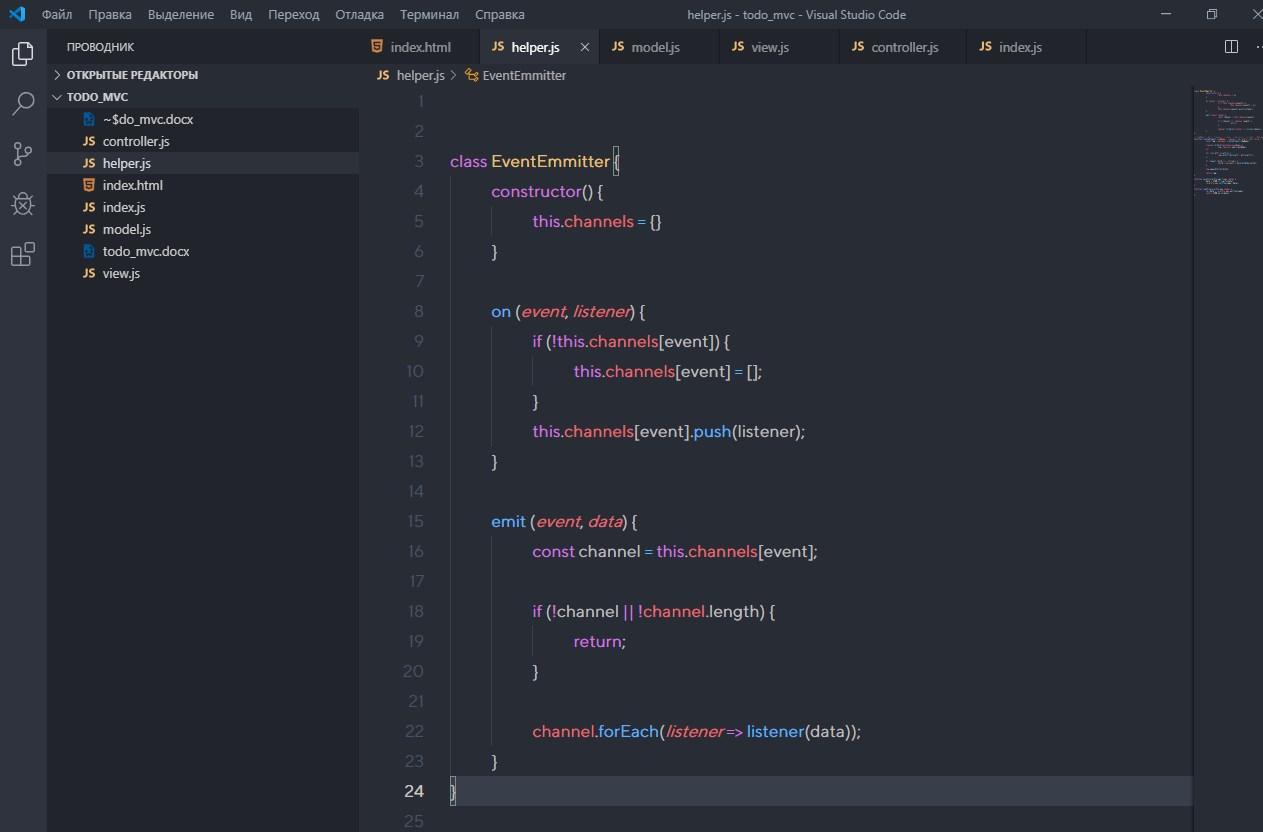
****

**Создал заготовки JS файлов и подключил их к разметке**

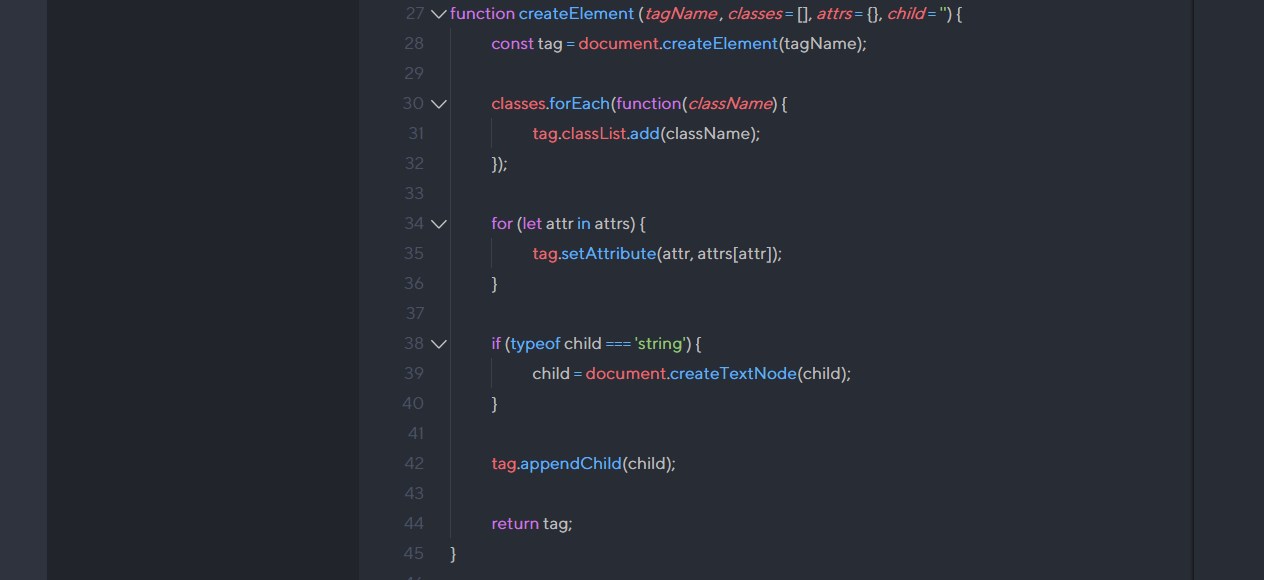
****

**В файле helper.js будут находиться вспомогательные функции и классы, которые упростят написание кода и реализуют общую логику приложения.**

**Первой такой функцией, а точнее классом является реализация поведенческого паттерна проектирования pub/sub. С помощью этого паттерна будет реализована взаимосвязь представления (View) и модели (Model) с контроллером (Controller)**

****

**Функция createElement упрощает создание HTML элементов**

****

**И две демонстрационные функции, чтобы продемонстрировать как легко и удобно можно хранить и получать данные из Базы данных. В качестве примера базой данных будет локальное хранилище браузера localStorage. Функция saveInLocalStorage сохраняет объект в localStorage под определённым именем, а loadFromLocalStorage получает данные.**

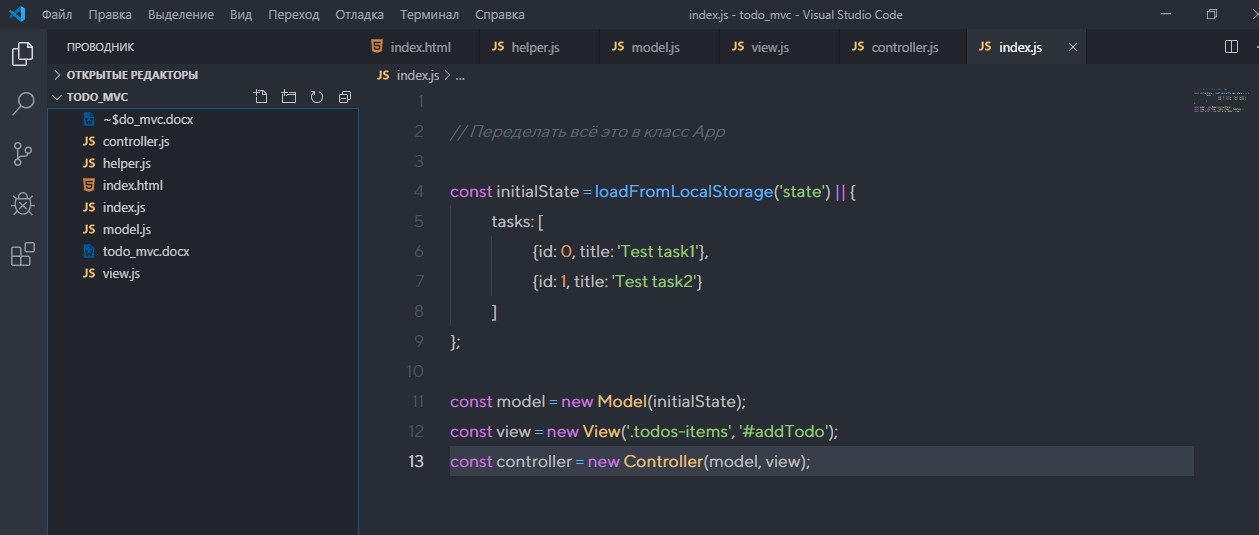
****

**Файл index.js является точкой входа в приложение. Здесь создаются экземпляры классов Model, View, Controller.**

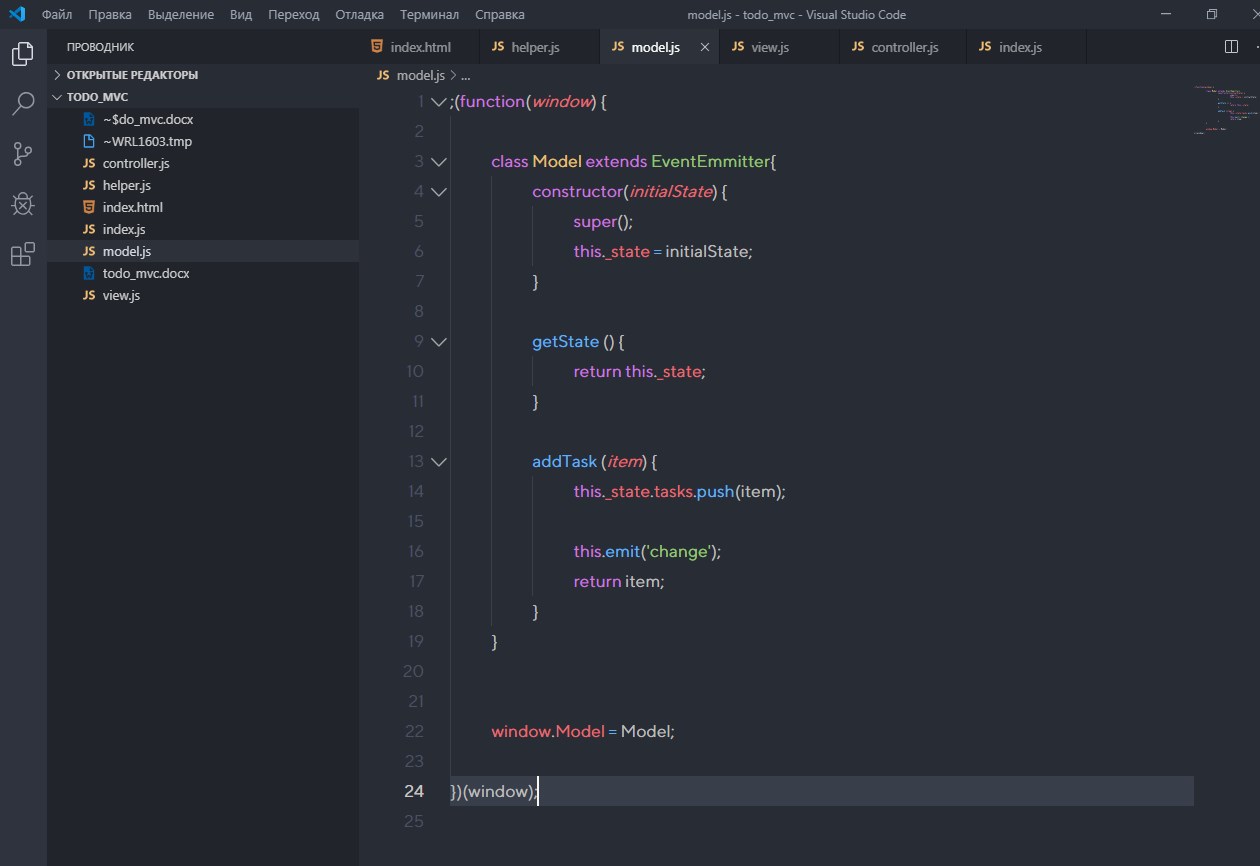
**В модель передаются данные о задачах (tasks) из localStorage, если таковых нет, то создаются 2 задачи по умолчанию.**

**В представление передаются селектор элемента в который будут рендериться задачи и селектор формы добавления новой задачи.**

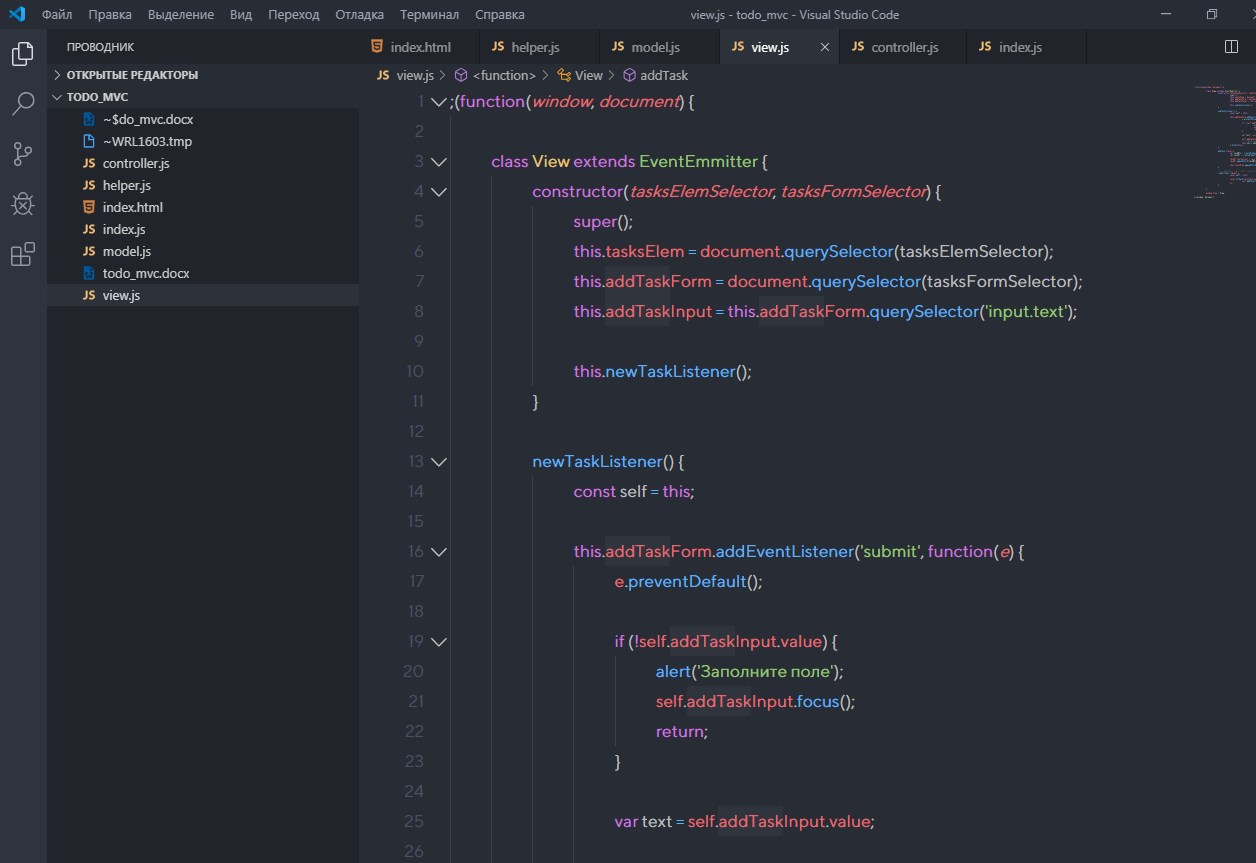
**В контроллер передаётся модель и представление.**

****

**Реализация модели (класса Model)**

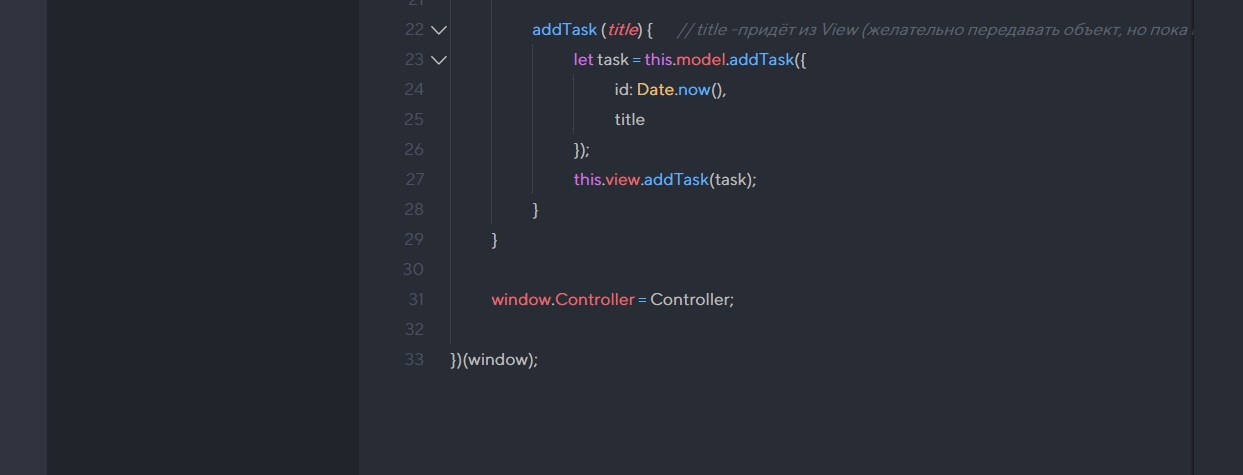
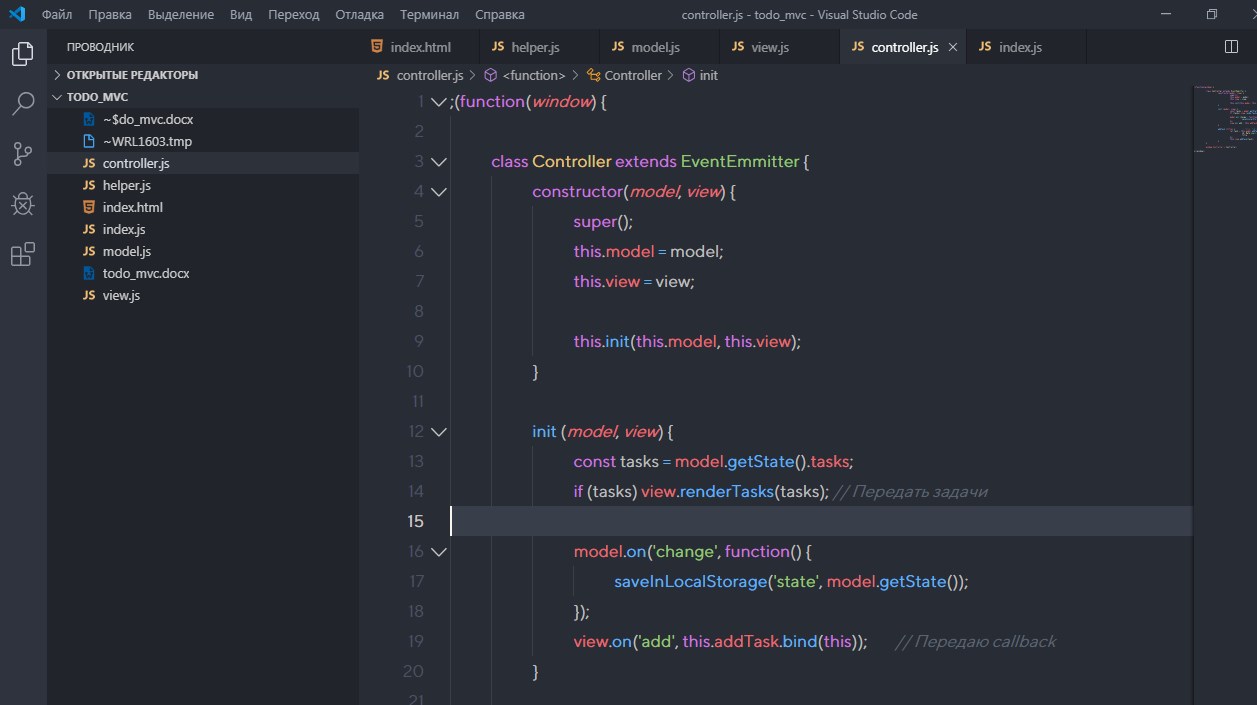
****

**Реализация представления (класса View)**

****

****

**Реализация контроллера (класса Controller)**

****