

8. Учебный проект: шаблонизируй это

Рабочая ветка `module3-task1`

Taskmanager

В этом задании мы отделим данные от представления: избавимся от статического контента в шаблонах компонентов и создадим для каждого из компонентов подходящую структуру данных.

Задача

Придумать структуру данных для наших компонентов и доработать функции отрисовки компонентов так, чтобы данные были отделены от шаблонов (вёрстки).

Чтобы понять, что должно оказаться в данных, а что нет, задайте себе вопрос: «Есть ли смысл скачивать эту информацию с сервера отдельно, может ли она измениться?» Не стоит заводить в данных структуру, которая описывает размеры логотипа или статический текст, в структурах должны храниться только те данные, которыми мы будем оперировать в проекте: получать их с сервера, изменять, отправлять на сервер обратно.

1. Для начала нам потребуется структура данных, которая опишет задачу. Это будет объект с полями:

- *description* — постановка задачи. Случайная строка из трёх на выбор Изучить теорию, Сделать домашку, Пройти интенсив на соточку;
- *dueDate* — дата и время запланированного выполнения (дедлайн). Объект типа Date или null, если срок исполнения не установлен. Просрочена ли задача будем узнавать из этого поля. Ограничение: плюс-минус неделя от текущей даты;
- *repeatingDays* — объект с фиксированными ключами mo, tu, we, th, fr, sa, su и булевым значением. Повторяется ли задача будем узнавать именно из него;
- *color* — строка, описывающая цвет карточки. Один вариант из набора black, yellow, blue, green, pink;
- *isFavorite* — булево значение, добавлена ли задача в избранное;
- *isArchive* — булево значение, помещена ли задача в архив;

- Остальные данные ограничьте самостоятельно. Что ещё должно быть в структуре, можно узнать из технического задания.

Обратите внимание, это именно описание структуры данных, а не готовый к использованию объект. Ваша задача написать функцию, которая как раз будет по этой структуре данных создавать и возвращать готовые объекты.

2. Напишите функцию, которая будет возвращать готовые объекты по структуре из предыдущего пункта. С помощью этой функции в `main.js` сгенерируйте временные данные (моки). Побольше... 15-20 объектов. Для удобства их лучше сложить в массив.
3. Теперь давайте перепишем функцию по созданию шаблона задачи, чтобы она на вход принимала данные — объект определённой структуры.
4. Аналогичным образом перепишите функцию по созданию шаблона формы редактирования задачи. Обратите внимание на условия поведения того или иного элемента формы, условия описаны в техническом задании. Ещё один нюанс в том, что форма редактирования выступает и в роли формы создания, поэтому нужно предусмотреть, чтобы шаблон корректно отображался с «пустыми» данными.

5. Опишите структуру для компонента фильтров. Это массив объектов с ключами:
- *title* — название фильтра. Возьмите из разметки;
 - *count* — количество задач, сопоставимое с фильтром. Вычислите эти данные на основе моков, полученных в п. 2, посчитав сколько всего задач, сколько просроченных, у которых дедлайн сегодня, сколько повторяющихся задач, а сколько в архиве.
6. Затем перепишите функцию по созданию шаблона фильтров, чтобы она на вход принимала данные — массив объектов из предыдущего пункта.
7. Далее перепишем код в `main.js` для работы с моковыми данными. На основе первого по порядку элемента в массиве отрисуйте компонент «Форма редактирования задачи», а на основе следующих 7 из оставшихся элементов отрисуйте компонент «Карточка задачи».
8. В заключение реализуем показ оставшихся задач. Для этого напишите функцию, которая будет отрисовывать ещё по 8 компонентов «Карточка задачи» (или меньше, смотрите ограничения в техническом задании) при клике на кнопку `Load more` и скрывать саму кнопку, когда

больше отрисовывать будет нечего.