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.

9. Личный проект: шаблонизируй это

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.

Киноман

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

   - Название фильма. Можно взять с постеров, а можете взять из списка своих любимых фильмов. Пока это не важно;

   - Постер (название файла). Один из набора файлов в директории `/public/images/posters`;

   - Описание. От 1 до 5 случайных предложений из текста: `Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras aliquet varius magna, non porta ligula feugiat eget. Fusce tristique felis at fermentum pharetra. Aliquam id orci ut lectus varius viverra. Nullam nunc ex, convallis sed finibus eget, sollicitudin eget ante. Phasellus eros mauris, condimentum sed nibh vitae, sodales efficitur ipsum. Sed blandit, eros vel aliquam faucibus, purus ex euismod diam, eu luctus nunc ante ut dui. Sed sed nisi sed augue convallis suscipit in sed felis. Aliquam erat volutpat. Nunc fermentum tortor ac porta dapibus. In rutrum ac purus sit amet`

`tempus.`

- Комментарии. От 0 до 5 штук;
  Обратите внимание, комментарии — это отдельная структура данных с эмоцией, датой, автором и сообщением, а не просто массив строк в структуре фильма.

- 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 карточки фильма, чтобы она на вход принимала

данные — объект определённой структуры.

4. Аналогичным образом перепишите функцию по созданию шаблона попапа с подробной информацией о фильме. Обратите внимание на условия поведения того или иного элемента попапа, условия описаны в техническом задании.

5. Разработайте структуры для остальных компонентов и вычислите для них данные на основе моков, полученных в п. 2. Например, придумайте структуру для фильтров и вычислите количество фильмов, сопоставимое с этими фильтрами. Для некоторых компонентов структура данных может не потребоваться. Например, для компонента, который показывает в подвале количество фильмов в базе данных, в шаблон достаточно передать число — количество.

6. Для оставшихся компонентов также перепишите функцию по созданию шаблона, чтобы она на вход принимала данные.

7. Далее перепишем код в `main.js` для работы с моковыми данными. На основе первого по порядку элемента в массиве отрисуйте компонент «Подробная информация о фильме (попап)», а на основе первых 5 из всех

элементов отрисуйте компонент «Карточка фильма».

Не забудьте, что показ блоков «Top rated» и «Most commented» — часть дополнительного задания. Оно выполняется по желанию.

8. В заключение реализуем показ оставшихся фильмов. Для этого напишите функцию, которая будет отрисовывать ещё по 5 компонентов «Карточка фильма» (или меньше, смотрите ограничения в техническом задании) при клике на кнопку `Show more` и скрывать саму кнопку, когда больше отрисовывать будет нечего.

Bigtrip

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

    - Тип точки маршрута. Один вариант из набора. Набор можно найти в техническом задании;

    - Пункт назначения (город). Названия городов можно взять из вёрстки, а можете использовать те, в которых вы хотели бы побывать. Пока это не важно;

    - Дополнительные опции. От 0 до 5 штук; *Обратите внимание,* дополнительные опции — это отдельная структура данных с типом, к которому опция относится, названием и ценой, а не просто массив строк в структуре точки маршрута.

    - Информация о месте назначения:

        - Описание. От 1 до 5 случайных предложений из текста: `Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras aliquet varius magna, non porta ligula feugiat eget. Fusce tristique felis at fermentum pharetra. Aliquam`

```
id orci ut lectus varius viverra.
Nullam nunc ex, convallis sed finibus
eget, sollicitudin eget ante.
Phasellus eros mauris, condimentum sed
nibh vitae, sodales efficitur ipsum.
Sed blandit, eros vel aliquam
faucibus, purus ex euismod diam, eu
luctus nunc ante ut dui. Sed sed nisi
sed augue convallis suscipit in sed
felis. Aliquam erat volutpat. Nunc
fermentum tortor ac porta dapibus. In
rutrum ac purus sit amet tempus.;
```

- Фотографии (одна или несколько). Можно использовать заглушку `http://picsum.photos/248/152?r=${Math.random()};`

- 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 точки маршрута, чтобы она на вход принимала данные — объект определённой структуры.

4. Аналогичным образом перепишите функцию по созданию шаблона формы редактирования точки маршрута. 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. Разработайте структуры для остальных компонентов и вычислите для них данные на основе моков, полученных в п. 2. Например, придумайте структуру и вычислите маршрут поездки на основе данных о всех точках маршрута.

6. Для оставшихся компонентов также перепишите функцию по созданию шаблона, чтобы она на вход принимала

данные.

7. Далее перепишем код в `main.js` для работы с моковыми данными. На основе первого по порядку элемента в массиве отрисуйте компонент «Форма редактирования точки маршрута», а на основе оставшихся элементов отрисуйте компонент «Точка маршрута».

Не забудьте, что показ блоков с информацией о маршруте и стоимость поездки — часть дополнительного задания. Оно выполняется по желанию.

8. В заключение распределите точки маршрута в списке согласно дате. Обратите внимание на вёрстку, для каждого дня путешествия заведён отдельный контейнер с порядковым номером. И в этот контейнер отрисовываются точки маршрута, соответствующие этой дате.