

Рецензия на тестовое задание (React)

Выполнил: Данила Богданов

Рецензент(ы): Иннокентий Кондратьев, Тимур Злобин

Дата: 19.02.2021

[Общее](#)

[Код](#)

[Архитектура](#)

[Оформление](#)

[Замечания](#)

[Функционал](#)

[Создание/Редактирование задачи](#)

[Обновление списка задач](#)

[Прочее](#)

[Тестировалось в/на](#)

[Потраченное время](#)

[Общая оценка](#)

Общее

Реализован многофункциональный компонент, позволяющий выбирать предлагаемые варианты. Выбор предоставляется в виде массива. Компонент принимает на вход массив объектов со набором полей для списка, загружаемый по URL. Поддерживается множественный выбор.

Основные требования к приложению выполнены частично.

Не реализованный функционал:

1. Нет тех. документации с описанием того, как можно использовать компонент;
2. Передача данных по URL реализована некорректно;
3. Отсутствует возможность кастомизации компонента;
4. Отсутствует возможность отображения списка компонента в отдельном модальном окне;
5. Нет возможность указания placeholder;
6. Дизайну стоило уделить больше времени.

Из плюсов:

1. Использование функциональных компонентов;
2. Использование хуков;
3. Реализация асинхронных функций.

Код

Архитектура

- применение архитектурных шаблонов - нет
- разделение кода по классам - нет
- использование сторонних ресурсов - нет

Оценка: 2.5 из 5

Оформление

- именование переменных и методов согласно codestyle - да
- код хорошо читается - да (местами нет, указано в замечаниях)
- код в достаточной мере документирован - да

Оценка: 4 из 5

Замечания

- Все слабо декомпозировано, можно было вынести асинхронные функции в отдельную папку;
- Нет единообразия в комментариях (на разных языках), а также в итоговом варианте имеется закомментированный код, который не имеет смысла.
- Для запуска проекта надо менять в index.css импорт с `@import url('app.css')` на `@import url('App.css')`;
- Функционал подключения иконок реализован необычно: вместо часто используемого SVG выбраны эмоджи, которые будут по разному отображаться на разных ОС (в худшем случае их может не быть).
- Вызывает вопросы реализация хука `useState`:
`const [dataJson, setDataJson] = useState({ hits: [] });`
Учитывая то, как используются эти данные, намного лучше и нагляднее было бы написать:
`const [hits, setHits] = useState([]);`
- В компонент передаются лишние пропсы, которые нигде не используются. При отправке на проверку стоило почистить код;
- URL JSON файла должен передаваться как пропс внутри компонента, а не быть константой при любых условиях.
- Лишние логи также стоило удалить.
- Учитывая, что `multiSelect` - булево значение, в условии `if (!multiSelect) {} else if (multiSelect) {}` блок `if (multiSelect)` является лишним и бесполезным – если не 0, то в любом случае 1, и наоборот.

- Появляется вопрос необходимости использования Spread оператора в фрагменте:

```
setSelection([...selectionAfterRemoval]);
```

- Фрагмент

```
if (selection.some(current => current.id === item.id)) { return true; } return false;
```

можно намного лаконичнее переписать

```
return (selection.some(current => current.id === item.id))
```

- Не понятно, почему где то используется нотация функций через Function Declaration, а где то через Function Expression.

- `onKeyPress={() => toggle(!open)}`

Функция `toggle` не принимает в себя аргумента при определении, но зачем-то при вызове он туда поступает. Здесь можно было не использовать стрелочные функции вообще, а написать `onKeyPress={toggle}`. То же самое ещё в других местах.

- Во многих местах вместо использования тернарного оператора можно было обойтись оператором `И` (`&&`).

Функционал

Прочее

- Запрос на сервер происходит по нажатию на кнопку, что не корректно, так как это селектор. Получается при каждом закрытии будет происходить запрос. Все сайд эффекты должны вызываться после монтирования компоненты ([Жизненные циклы](#)), так как это функциональная компонента, тут в `useeffect` -те.
- При фокусе в инпут меняется `changelInput` на противоположный, если ввести слово или цифру чтобы результат был пустой, то при каждом закрытии селектора и открытии, после фокуса в инпут будет показан пустой результат.
- Нет поиска у селектора по `url`.
- Довольно странный ключ `"picture"` в `file.json` – можно взять `url` с любым изображением из интернета вставить, да отрисовывать.

Оценка (суммарно по всем пунктам): 2.5 из 5

Тестировалось в/на

Приложение тестировалось в браузерах Firefox, Chrome. Запущенная локально.

Резюме

Потраченное время

2 недели, количество часов не указано. Судя по коммитам, времени на реализацию потрачено немного.

Общая оценка

2 из 5

Рекомендовано

Изучить [жизненный цикл](#) React компонента. Почитать про файловую структуру проектов. Доделать тестовое задание с учетом замечаний. В целом в правильном направлении делается задание.

Личное пожелание – делать тестовые задания в разные компании по очереди, а не за раз, т.к. велика вероятность, что не попадешь никуда из-за нехватки времени.