

Тестовое задание (Front-end/React-Redux)

- Реализовать одностраничное Web-приложение используя React-Redux.
- Можно использовать любые готовые библиотеки компонентов (react-bootstrap, material-ui) или отдельно установленные компоненты (например, react-datepicker для выбора даты, см. описание задания ниже)
- Код должен быть выложен на github с описанием запуска, чтобы иметь возможность запустить и проверить работоспособность приложения.
- Особых требований по стилизации компонентов не указывается, но чем приятней будет выглядеть приложение, тем довольнее будут проверяющие 😊, а также Вы сможете продемонстрировать свои навыки CSS

Задание:

1. Приложение запрашивает и отображает 5 самых популярных вопросов на Stackoverflow, содержащих строку "react-redux" в наименовании, начиная с заданной пользователем даты.

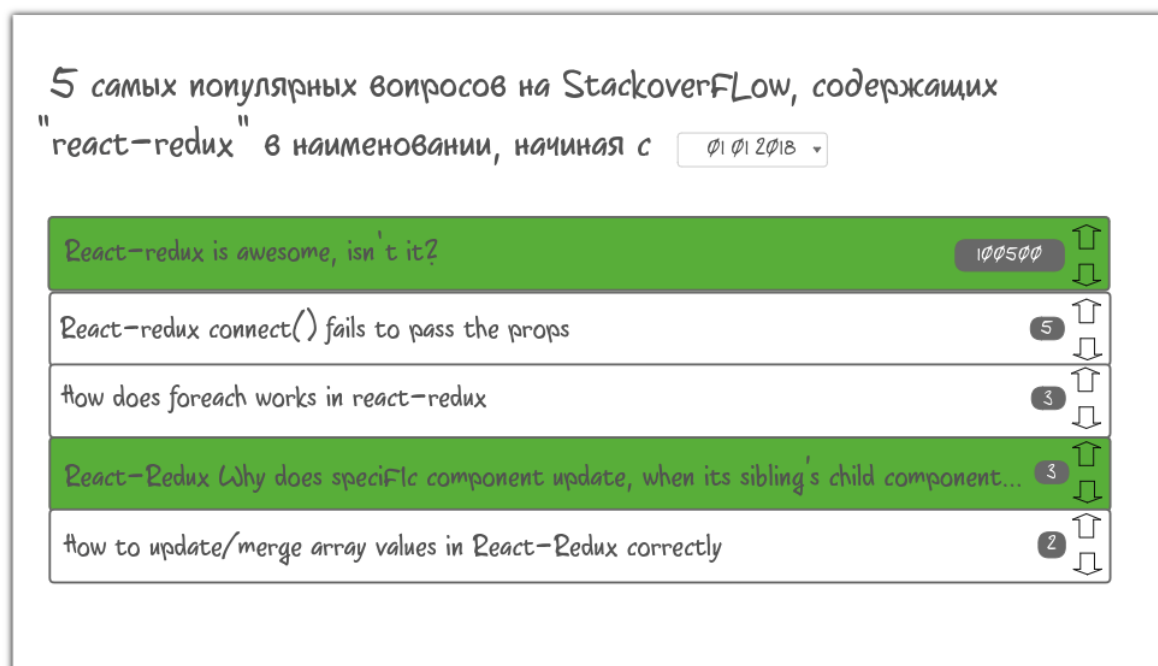
- Документация API StackExchange - <http://api.stackexchange.com/docs>
- Под популярностью вопроса понимается поле score объекта question <http://api.stackexchange.com/docs/types/question>
- Под наименованием понимается поле title того же объекта question

2. При старте приложения, сразу осуществляется описанный в п.1. запрос. Дата по-умолчанию, начиная с которой производится поиск 5 самых популярных вопросов при первом запросе - 01.01.2018.

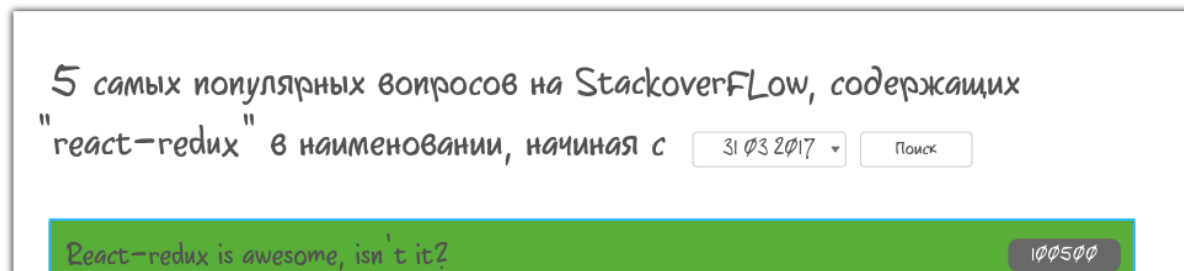
3. После того как данные были получены, нужно отобразить список с наименованиями вопросов и их рейтинг (поле score). Выделить зеленым цветом, если на вопрос был дан ответ (поле is_answered).

Рядом с рейтингом отображаются две кнопки "вверх" и "вниз", увеличивающие или уменьшающие рейтинг этого вопроса на единицу (никаких запросов выполнять не нужно, просто изменять значение рейтинга в store).

Примерно должно выглядеть так:



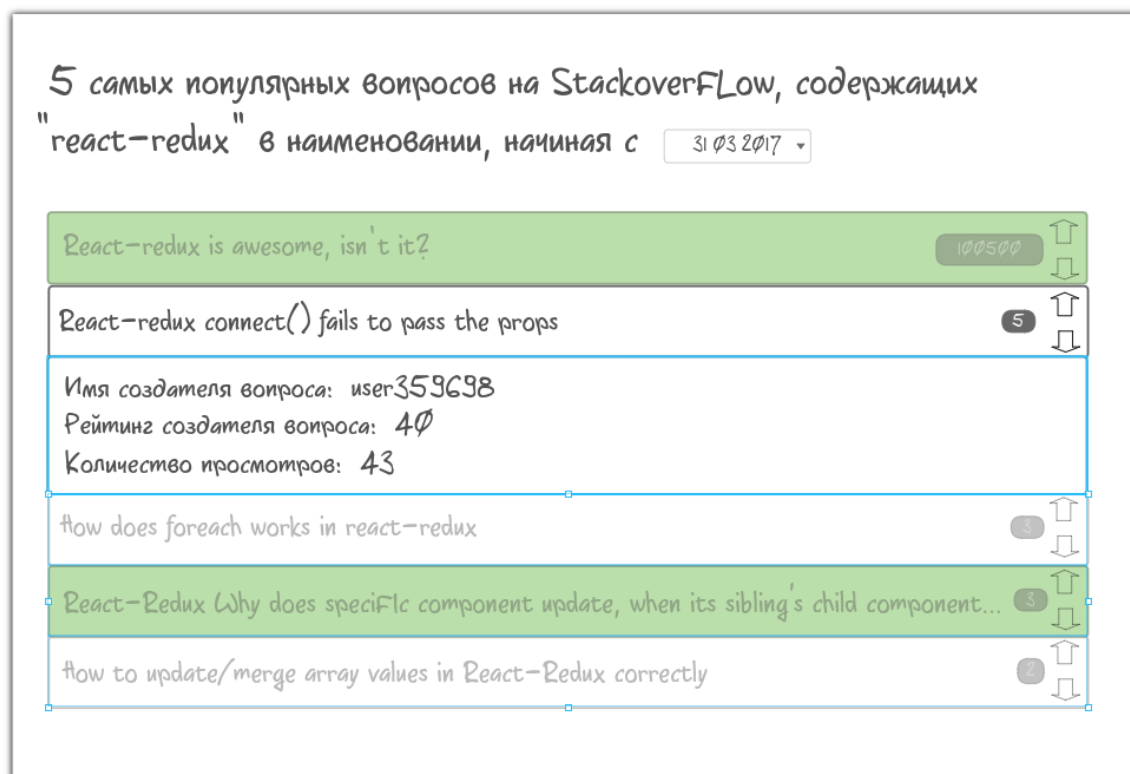
4. Если пользователь выбирает новую дату (т.е. дата в дэйтпикере не соответствует той, с которой были получены отображаемые результаты), то рядом с дэйтпикером появляется кнопка "Поиск", при нажатии на которую будет произведен запрос с новыми параметрами. Если дата не изменялась или была снова выбрана дата для текущих отображаемых результатов, то кнопка исчезает.



5. При клике на элемент списка, он раскрывается и в появившемся поле отображается 3-4 пункта дополнительной информации, например: имя создателя вопроса, его рейтинг, количество просмотров вопроса, дата последней активности и т.д.

При клике по другому элементу, он раскрывается, а старый элемент закрывается (т.е. одновременно может быть раскрыт только 1 элемент). При клике по пустому пространству вне элементов, любой раскрытый элемент закрывается.

Создание своего React-компонента раскрывающегося элемента списка с описанными элементами управления приветствуется. Или же, за основу можно взять, например, react-bootstrap Panel со свойством collapsible.



6. Реализовать возможность смены положения элементов в списке двумя способами:

- drag and drop (можно использовать react-dnd)
- двойным щелчком по элементу выделяем первый (показываем, что он выделен), выделяем двойным щелчком второй элемент и они меняются местами. При клике по пустой области, если был выделен первый элемент двойным щелчком, то выделение отменяется, т.е. процесс, описанный ранее, по смене элементов двойным щелчком начинается заново.