

Исходные данные

Есть сущность "Потребитель", имеющая 4 поля:

1. ИД (4 байтовое целое)
2. Имя (строка, не более 255 символов)
3. Тип (целое, принимает значения 1 - физическое лицо и 2 - юридическое лицо)
4. Номер потребителя (строка из 13 цифр)
5. Выбор имен полей произвольный.

Есть HTTP-backend(реализация не требуется), с 4 endpoint, без авторизации, обмен данными в формате JSON:

1. Получить список потребителей (отсортированный по имени)
2. Создать потребителя (возвращает JSON созданного потребителя с заполненным ИД)
3. Изменить потребителя
4. Удалить потребителя

Требуется

1. Необходимо сделать одностраничный UI на HTML+Javascript, который:
2. На открытии страницы запрашивает список потребителей и показывает его в табличном виде, при этом:
 - a. Список полей: Имя, Тип, Номер потребителя.
 - b. Тип должен выводиться в строковом виде ("Ф"/"Ю" с всплывающей подсказкой "Физическое лицо", "Юридическое лицо").
3. Позволяет создавать (без перехода на другую страницу) нового потребителя, при этом:
 - a. ИД не создается, а получается с сервера
 - b. Имя не должно позволять вводить более 255 символов и обязательно к заполнению.
 - c. Тип должен выбираться из выпадающего списка.
 - d. Номер потребителя должен состоять ровно из 13 цифр и обязателен к заполнению. UI не должен давать вводить что-то отличное от цифр.
 - e. Должна быть реализована валидация на клиенте любым способом, включая использование библиотек, которая будет оповещать пользователя о необходимости заполнить обязательные поля и о (попытке) превышения лимита на длину поля.
4. Позволяет редактировать потребителя (требования аналогичны созданию).
5. Позволяет удалять потребителя.
6. Позволяет отфильтровать список потребителей по типу: физ/юр. Фильтр осуществляется на клиенте.
7. Работает в IE 9+, а также в последних версиях Firefox, Chrome.

При этом:

- Наличие отрицательных веток (например, серверная валидация потребителя) для простоты считаем непредусмотренным: т.е. сервер всегда отвечает кодом 200, если соотв. данные вообще есть, и всегда сохраняет именно то, что было прислано клиентом.
- Реакцию на отсутствующие данные (например, удаление несуществующего потребителя) предусматривать в решении не нужно.
- Предложенное решение должно работать без backend, на основании синтетических данных в памяти, при этом:
 - Должно быть изначально доступно не менее 5 потребителей разного типа.
 - Переключение на работу с настоящим бэкенд (в частности, подстановка настоящих URL) должно осуществляться просто и должно быть документировано в коде/прилагающемся к заданию документе.

- Все соглашения/ожидания по поводу поведения бэкенда, кроме описанных в настоящем документе, должны быть документированы в прилагающемся к заданию документе.
- Работоспособность и юзабельность UI на мобильных устройствах не требуется.
- Желательно разделение кода Javascript на несколько "слоев" и реализация некоторого количества unit-тестов (библиотека любая).
- К ответу следует приложить текстовый документ с описанием всех предположений, принятых за истину при разработке задания.

Требования к используемым технологиям

- Допускается использование только Javascript (не Dart, Typescript, Coffeescript, ...). Допускается использование Babel.
- Допускается использование любых библиотек (jQuery, underscore, lodash, ...), кроме React и аналогов (Preact, Vue, ...).
- Не допускается использование frameworks (Angular, Ember,...), кроме Backbone.
- Код (Javascript, CSS), написанный в ходе выполнения задания, не должен быть минифицирован или объединен с кодом использованных библиотек. Если исполняемый код Javascript получен использованием транспилятора JS->JS, должен прилагаться исходный код.