# Тестовое задание для соискателя должности Frontend-разработчика INTELSONLINE.

Вам необходимо создать небольшое Web-приложение для работы с данными, полученными через REST интерфейс.

Приложение имеет зависимости. В качестве пакетного менеджера используется bower. С его помощью устанавливается angular.js v.1.6.1. Для установки bower используйте npm (3.5.2). Также, включите в зависимости любой удобный вам сборщик.

Приложение должно состоять из исходников и собранного варианта. Соответственно, в папке приложения должны быть директории src и build.

Приложение отображает страницу с заголовком «Результаты поиска». Под заголовком висит текст «Загрузка…». Приложение обращается по адресу: <http://dev2.poiskznakov.ru/api-rest/test-me/>. Получившийся ответ выводится в виде таблицы, которая заменяет текст «Загрузка…»:

|  |  |  |  |
| --- | --- | --- | --- |
| Тип поиска | Запрос | Дата | Найдено |

Тип поиска – это читаемая строка, которая получается после перевода идентификатора:

|  |  |
| --- | --- |
| ID | Выводить |
| wdesc | По словесному элементу |
| owner | По владельцу |
| document\_number | По номеру документа |

Все соответствия должны быть указаны в коде.

Кроме-того, сам эндпоинт имеет возможность выводить записи по указанному типу поиска (найдите это в документации к нему). Перед таблицей должно быть реализовано поле Select с выбором типа поиска. При смене типа поля вешается текст «Загрузка…», обновляются данные, выводится таблица.

Если данных нет, то вместо таблицы выводите сообщение: «Ничего не найдено».

Тип поиска по-умолчанию – все. Этот тип должен быть задействован в Select. Если сменить тип поиска и обновить страницу, выбранный тип должен отображаться выбранным в поле Select и, соответственно, сразу должны грузиться данные по выбранному типу.

Дату следует представлять в формате YYYY-MM-DD.

Для оформления приложения используйте 1 CSS файл. Сильно оформлять не нужно. Просто сделайте все отступы визуально приятными, заголовки – не слишком большими, текст тёмно-серым, а не чёрным.

В случае ошибки сервера, меняйте контент на «Сервер временно недоступен», и под этим сообщением выводите кнопку «Повторить». При нажатии на кнопку контент заменяйте, как обычно, на текст «Загрузка…» и пробуйте снова получить данные.

Результат труда выложите в GIT, либо упакуйте в ZIP архив и передайте менеджеру. Не забывайте про необходимость сборки приложения в три-четыре автономных рабочих файла:

1. index.html
2. bundle.js
3. vendor.js (не обязательно)
4. style.css

Во время проверки, преимущественно, будут учитываться:

1. Использование компонентов.
2. Выполнение всех условий
3. Читаемость кода
4. Комментарии в JS и HTML
5. Отсутствие лишних выводов в консоль
6. Стремление избежать лишних if statements

Отсутствие багов, ошибок, недоработок подразумевается само-собой.