React.js Developer Test

During the selection process, we require each candidate to complete a task in order to make sure you have the skills needed for this position.

Test duration

7 days from the date the task is received.

Task solution

Task solution must be sent as a Github URL.

Send us your task solution or any questions you may have to with the theme of the email "React Technical Task - <your surname>" to: jobs@esprow.com

Disclaimer

The only purpose of this task is to assess candidate skills. Everything sent as a result of the task will not be used for commercial purposes.

Good luck and have fun! ❀

Task requirements

Create an application that will take a RANDOM GENERATED JSON file and create a renderer and editor for the JSON file.

Here is an example of the JSON file.

```
[
  {
    "id": "4640a713-4b85-4c2a-9239-e8dcb326fac3",
    "isActive": true,
    "picture": "http://placehold.it/32x32",
    "age": 33,
```

      "name": "Adele Vincent",

      "email": "adelevincent@hometown.com",

      "address": "132 Vandervoort Place, Wright, Texas, 7991",

      "about": "Officia ipsum nostrud quis ad ad. Commodo dolor anim pariatur deserunt magna irure ea ea culpa sit ullamco sint. Ea voluptate ipsum cupidatat reprehenderit in consequat commodo exercitation pariatur cillum quis ullamco. Do duis aliquip labore adipisicing ad aute ullamco culpa excepteur ipsum eu. Ipsum ex duis tempor aliqua.
Sint ex ea ex commodo incididunt veniam culpa officia magna.\r\n",

      "registered": "2019-02-07T12:37:40 -01:00"

    },

    {

      "id": "e8d04e55-f289-495a-84bc-97b0224da81a",

      "isActive": false,

      "picture": "http://placehold.it/32x32",

      "age": 36,

      "name": "Willis Acosta",

      "email": "willisacosta@hometown.com",

      "address": "666 Dekalb Avenue, Vandiver, Nevada, 4620",

      "about": "Magna eiusmod nostrud enim proident sit magna cillum. Laborum culpa quis occaecat elit. Do occaecat quis nisi consectetur.\r\n",

      "registered": "2018-12-14T06:52:38 -01:00"

    },

    {

      "id": "3bfd8718-ee2f-4b81-bc43-2cd8dd36be7c",

      "isActive": true,

      "picture": "http://placehold.it/32x32",

      "age": 25,

      "name": "Alvarez Ochoa",

      "email": "alvarezochoa@hometown.com",

      "address": "106 Bancroft Place, Eden, District Of Columbia, 259",

      "about": "Esse ea reprehenderit laboris esse voluptate reprehenderit occaecat velit aute tempor nisi qui occaecat voluptate. Do ex veniam magna consectetur voluptate cupidatat culpa veniam aliqua consectetur veniam sit. Esse quis non sint cupidatat quis consequat. Eu adipisicing mollit minim minim eiusmod do voluptate adipisicing commodo.\r\n",

```
     "registered": "2015-05-08T12:39:06 -02:00"
  }
]
```

The system should create a page where for every array member in the JSON file we will see a row with property->value, i.e. the data of the array member. If the JSON has 10 array members, the page will render 10 rows. If it has 1000 array members, it will render 1000 rows.

Filtering and ordering of the rows is not required, but it would be a plus if you implement it.

We should be able to edit the data of each row/array member. How you implement the editing (inline editing, additional pop-up component for editing, additional row for editing, etc.) is up to you.

It is important to note that all these items are on the same page. There is no pagination, and everything is always visible.

How is the edit system generated?

If the id field exists, it will not have an input field for editing.

If the value is a string, it will create an input field with type text.

If the value is a number, it will create an input field with a number as a type.

If the value is email, it will create an input field with email as type.

If the value is a date, it will use a html date picker.

If the value is boolean, it will use a radio button with "true/false".

If the field value is a long text, use the textarea field.

If the field value is another JSON, just ignore it.
There is no need for data validation There is no need for a fancy UI.

NOTES:

- **Task must be done in Typescript.**

- React must be used for the front-end.

- You can use Redux, or Context, or just state to manage your data.

- Please add to your solution the instructions for launching it (readme format can be used).

- Please use Linter with this configuration https://github.com/ezelohar/task-boilerplate/blob/master/.eslintrc-typescript https://github.com/ezelohar/task-boilerplate/blob/master/package-list-for-eslinttypescript

- **Performance is important**. The JSON file can have unlimited data, so think it through.

- With this task you must provide a JSON file with 1000 rows at minimum and you can use https://www.json-generator.com/ to build your json file with ease.