

Jones - Exercise

Jones Form

First Name:

Last Name:

Mail Address:

Phone Number:

Use the following link:

<https://github.com/facebookincubator/create-react-app/blob/master/README.md#getting-started>

And create a simple react app:

1. One registration form with the title “Jones Form” and the following fields and validations:
 - a. First Name, validate that the inserted value is:
 - i. not empty
 - ii. contains only alphabet letters (a-z/A-Z)
 - iii. minimum 2 chars
 - b. Last Name, validate that the inserted value is:
 - i. not empty
 - ii. contains only alphabet letters (a-z/A-Z)
 - iii. minimum 2 chars
 - c. Mail Address, validate that the inserted value is:
 - i. not empty
 - ii. valid mail address
 - d. Phone Number, validate that the inserted value is:
 - i. not empty
 - ii. only numbers
 - iii. exactly 10 digits
2. Submit button which sends a mail with all the inserted details in the following format:

Mail subject: New Lead

Mail content:

First Name: <firstName>

Last Name: <lastName>

Mail Address: <mailAddress>

Phone Number: <phoneNumber>

General guidelines:

- You can use any node module and external JS library you want.

- Follow the **coding conventions** indicated below.
- **Reuse code** as much as possible (**hint1**: validations, **hint2**: React components)
- Basic design is enough (black and white is fine) - just make it clear and organized.
- You **don't** necessarily need a server to send an email (e.g. Sendgrid API via free trial).

When you are done please send us:

- The code including the modules folder you used, zip the entire folder together.
- Instructions how to run the code, we **don't** need it to run on every device and browser.
- Screenshots:
 - Of the mail we get on a successful submit.
 - Of the form once the page is loaded.
 - Of the form once the user inserted the following values and pressed Submit (we want to see the validations):

First Name: jones
Last Name: j
Mail Address: jones@jones.jones
Phone Number: 123456789

Feel free to contact us regarding any question, we will be happy to help.

You can contact us via email or phone.

***** **Make sure to follow the following coding conventions** *****

Coding conventions and guidelines:

- General:
 - Names:
 - **method and variable** should start with lowercase letter (and then **CamelCase**) - letters only.
 - **Constants** should be in uppercase letters. e.g. RED, YELLOW, MAX_PRIORITY.
 - Use significant names for variables and methods so an outsider can understand what they do. Also, make sure the function does as it names says. E.g. function with the name verifyToken should verify a token and return boolean but shouldn't return a token to be verified.
 - Do not use hyphens or underscores in names (except for constants).
 - **Method**:

- Put the opening bracket at the end of the first line and use one space before it.
- Put the closing bracket on a new line, without leading spaces.
- **Variable:**
 - Usually keep variable declarations in separate lines.
- **No commented code should be pushed**, either a we are sure about the code or not.
- Maintain **Constant number of spaces** - examples:
 - Long j = 7; and not var j=7; or Long j= 7 ;
 - if (x == 1) and not if(x==1)
 - k = k + 7;
 - var values = ["Volvo", "Saab", "Fiat"];
 - for (i = 0; i < 5; i++) {
- Maintain **indentation** in all files..
- Don't break a line after 80 chars, it is making it harder to read the code. **Break only on really long lines** (can't see the end of the line on the screen).
- Java Script:
 - Always put semicolon after every statement (except after function definition).
 - Object structure:

```
var person = {
  firstName: "John",
  lastName: "Doe",
  age: 50,
  eyeColor: "blue"
};
```