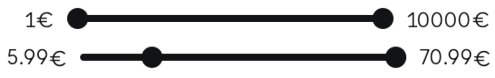
MANGO

Objective

Know your code skills for an every-day code problem based
on our team design system's needs.

Exercise

You have to create the following component: **<Range />**
You have to use React to create the solution.
You do NOT have to use any CLI to create structure and architecture of your application.
This component has two use modes:



1. Normal range from min to max number
2. Fixed number of options range

Use cases

Normal Range:

Provide a localhost:8080/exercise1 route with the following:

- The component CAN'T be a HTML5 input range. It has to be a custom one.
- The user can drag two bullets through the range line.
- The user can click on both currency number label values (min or max) and set a new value.
- The value will never be less than min or greater than max input values.
- When some bullet is on hover, this bullet has to be bigger and change cursor's type into draggable.
- Dragging a bullet turns cursor to dragging
- Min value and max value can't be crossed in range
- For this example, provide a mocked http service returning min and max values that have to be used in the component. Example: {min: 1, max: 100}. Use https://www.mockable.io/ or a custom mocked server.
- Do as many unit tests as you can.

Fixed values range:

Provide a localhost:8080/exercise2 route with the following:

- The component CAN'T be a HTML5 input range. It has to be a custom one.
- Given a range of values: [1.99, 5.99, 10.99, 30.99, 50.99, 70.99] the user will only be able to select those values in range
- Provide a mocked http service that returns the array of numbers: [1.99, 5.99, 10.99, 30.99, 50.99, 70.99]. Use h ttps://www.mockable.io/ or a custom mocked server.
- For this type of range, currency values are not input changable. They have to be only a label
- The user can drag two bullets through the range line.
- Min value and max value can't be crossed in range
- For this example, provide a mocked service returning min and max values that have to be used in the component. Example: {rangeValues: []}
- Do as many unit tests as you can.

Extra:

– You can use any mocked way for provide services data.
– You can give us your solution in any way. It's up to you!

HAPPY CODING!