

# ReactJS map through Object

Asked 4 years, 11 months ago Active 10 months ago Viewed 151k times



I have a response like this:

50



16



```
TRAVEL COMPANY ▼ Object
  ▼ 6: Object
    id: 6
    max_age: 2
    min_age: 0
    name: "Baby tot 3 jaar"
    quantity: 0
    type: "person"
    ▶ __proto__: Object
  ▶ 7: Object
  ▶ 8: Object
  ▶ 9: Object
  ▶ 10: Object
  ▶ 11: Object
  ▶ 2021: Object
```

I want to display the name of each object inside this HTML:

```
{subjects.map((item, i) => (
  <li className="travelcompany-input" key={i}>
    <span className="input-label">{ item.name }</span>
  </li>
))}
```

But it throws an error of `subjects.map` is not a function .

First, I have to define the keys of the objects where it creates an array of keys, where I want to loop through and show the `subject.names` .

What I also tried is this:

```
{Object.keys(subjects).map((item, i) => (
  <li className="travelcompany-input" key={i}>
    <span className="input-label">key: {i} Name: {subjects[i]}</span>
  </li>
))}
```

javascript reactjs jsx

Share Improve this question

Follow

edited Dec 8 '18 at 15:09



Brett DeWoody

52.9k 27 126 176

asked Nov 25 '16 at 11:20



Sireini

3,722 10 45 82

Thnx for saying you know the answer? – Sireini Nov 25 '16 at 11:23

1 A word of **caution** here. The order of keys is not guaranteed in an object and if something else modifies

the object it may return the keys in a different order. This may result in content shifting around. – [ninjaas](#)  
Oct 26 '19 at 11:37

## 8 Answers

Active	Oldest	Votes
--------	--------	-------



When calling `Object.keys` it returns an array of the object's keys.

113

```
Object.keys({ test: '', test2: ''}) // ['test', 'test2']
```



When you call `Array#map` the function you pass will give you 2 arguments;



1. the item in the array,
2. the index of the item.



When you want to get the data, you need to use `item` (or in the example below `keyName`) instead of `i`

```
{Object.keys(subjects).map((keyName, i) => (
  <li className="travelcompany-input" key={i}>
    <span className="input-label">key: {i} Name: {subjects[keyName]}</span>
  </li>
))}
```

Share Improve this answer Follow

edited Aug 5 '20 at 12:30

answered Nov 25 '16 at 11:47



[TryingToImprove](#)

6,087 4 25 38

and to map through another field array – [Isaque Palmieri](#) Feb 17 at 18:41

Is there a reason not to use `Object.values` instead? – [MadOgre](#) Aug 26 at 18:41

@MadOgre not really - this was based on the OP's current solution. They used `Object.keys`  
– [TryingToImprove](#) Aug 31 at 9:11



You get this error because your variable `subjects` is an object not Array, you can use `map()` only for Array.

18



In case of mapping object you can do this:



```
{
  Object.keys(subjects).map((item, i) => (
    <li className="travelcompany-input" key={i}>
      <span className="input-label">{ subjects[item].name }</span>
    </li>
  ))
}
```

Share Improve this answer Follow

edited Jan 24 '19 at 11:49

answered Nov 25 '16 at 11:30

[Gayane](#)



409 3 12

Map over the keys of the object using `Object.keys()` :

7

```
{Object.keys(yourObject).map(function(key) {  
  return <div>Key: {key}, Value: {yourObject[key]}</div>;  
})}
```



Share Improve this answer Follow

edited Apr 7 '20 at 20:47



brooksrelyt

3,594 4 24 46

answered Sep 15 '18 at 0:30



Rami Salim

165 2 10

Use `object.entries()` function.

5

`Object.entries(object)` return:

```
[  
  [key, value],  
  [key, value],  
  ...  
]
```

see [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Object/entries](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/entries)

```
{Object.entries(subjects).map(([key, subject], i) => (  
  <li className="travelcompany-input" key={i}>  
    <span className="input-label">key: {i} Name: {subject.name}</span>  
  </li>  
))}
```

Share Improve this answer Follow

edited Dec 27 '19 at 7:08

answered Dec 26 '19 at 9:04



정형석

69 1 6

Do you get an error when you try to map through the object keys, or does it throw something else.

3

Also note when you want to map through the keys you make sure to refer to the object keys correctly. Just like this:



```
{ Object.keys(subjects).map((item, i) => (  
  <li className="travelcompany-input" key={i}>  
    <span className="input-label">key: {i} Name: {subjects[item]}</span>  
  </li>  
))}
```

You need to use `{subjects[item]}` instead of `{subjects[i]}` because it refers to the keys of the object. If you look for `subjects[i]` you will get undefined.

Share Improve this answer Follow

edited Sep 17 '18 at 11:28

answered Nov 25 '16 at 11:41



noa-dev

3,321 9 30 67

I use the below `Object.entries` to easily output the key and the value:

3

```
{Object.entries(someObject).map(([key, val], i) => (
  <p key={i}>
    {key}: {val}
  </p>
))}
```



Share Improve this answer Follow

answered Jan 12 at 0:23



StefanBob

4,319 2 23 35

This is a repeat of an earlier answer by 정형석 and others. – isherwood Sep 24 at 14:03

I am not sure why Aleksey Potapov marked the answer for deletion but it did solve my problem. Using `Object.keys(subjects).map` gave me an array of strings containing the name of each object, while `Object.entries(subjects).map` gave me an array with all data inside witch it's what I wanted being able to do this:



```
const dataInfected = Object.entries(dataDay).map((day, i) => {
  console.log(day[1].confirmed);
});
```

I hope it helps the owner of the post or someone else passing by.

Share Improve this answer Follow

edited May 5 '20 at 16:59

answered May 4 '20 at 16:32



YanMax

31 4

Instead of `(day, i)` you can do `([key, val], i)` and directly use the `val` in your code.  
– Guy Hoka May 8 at 20:32

Also you can use [Lodash](#) to direct convert object to array:

0

```
_.toArray({0:{a:4},1:{a:6},2:{a:5}})
[ {a:4}, {a:6}, {a:5} ]
```



In your case:

```
_.toArray(subjects).map((subject, i) => (  
  <li className="travelcompany-input" key={i}>  
    <span className="input-label">Name: {subject[name]}</span>  
  </li>  
))}
```

Share Improve this answer Follow

answered Mar 27 '20 at 6:28



[Nilanka Manoj](#)

**3,254** 3 12 40

---