# 

# What are React Synthetic events?

# [Can I use both ES6 and ES5 in the same React codebase?](https://stackoverflow.com/questions/37329081/can-i-use-both-es6-and-es5-in-the-same-react-codebase)

# Answer:

<https://stackoverflow.com/questions/37329081/can-i-use-both-es6-and-es5-in-the-same-react-codebase>

ES6 is not fully supported by all the browsers, hence we use babel or Traceur for transpiling the ES6 code to ES5.

**Yes, you can mix both ES6 and ES5 - ES6 is fully backwards compatible**, so essentially you could think of your entire app as ES6, but only use the new syntax and functionality in new code.

What are error boundaries?

<https://www.google.com/amp/s/www.digitalocean.com/community/tutorials/react-error-boundaries.amp>

What are protected routes ?

<https://dev.to/mychal/protected-routes-with-react-function-components-dh#:~:text=Protected%20routes%20allow%20us%20to,components%20along%20with%20react%2Drouter>.

What is Func Vs Class components?

<https://www.twilio.com/blog/react-choose-functional-components>

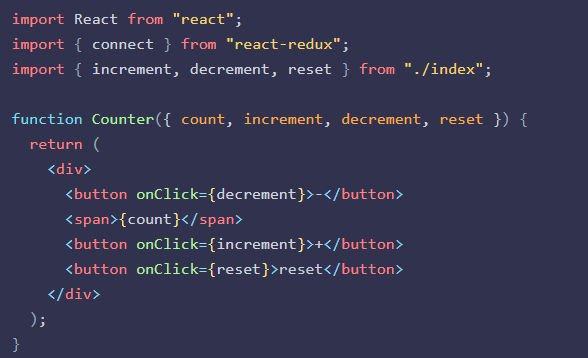
mapdispacthprops, mapstateToProps

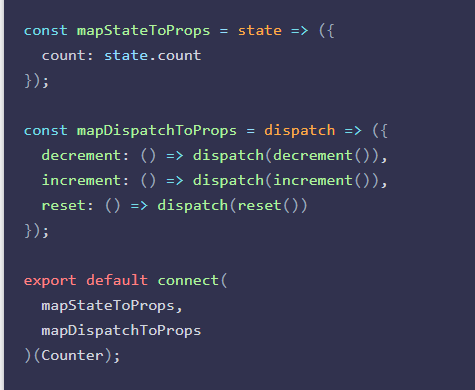
How to subscribe and unsubscribe from the store?

<https://blog.logrocket.com/react-redux-connect-when-and-how-to-use-it-f2a1edab2013/>

<https://daveceddia.com/redux-mapdispatchtoprops-object-form/>

Eg:





mapDispatchToPropss can also be a plain object



React controlled Vs uncontrolled components ?

<https://blog.bitsrc.io/react-controlled-vs-uncontrolled-components-f569f4215ac9>

This relates to stateful DOM components (form elements) and the React docs explain the difference:

* A [Controlled Component](https://facebook.github.io/react/docs/forms.html#controlled-components) is one that takes its current value through props and notifies changes through callbacks like onChange. A parent component "controls" it by handling the callback and managing its own state and passing the new values as props to the controlled component. You could also call this a "dumb component".
* A [Uncontrolled Component](https://facebook.github.io/react/docs/uncontrolled-components.html) is one that stores its own state internally, and you query the DOM using a ref to find its current value when you need it. This is a bit more like traditional HTML.

Most native React form components support both controlled and uncontrolled usage:

// Controlled: <input type="text" value={value} onChange={handleChange} /> // Uncontrolled: <input type="text" defaultValue="foo" ref={inputRef} /> // Use `inputRef.current.value` to read the current value of <input>

React Performance improvements?

<https://www.codementor.io/blog/react-optimization-5wiwjnf9hj>

1. Use Fragments
2. Don’t intialise state from props
3. Don’t use indexes as keys
4. Prefer immutability for objects and arrays, avoid reference errors
5. Prefer functional/pure components that will reduce the bundle size
6. Seperate the common modules, CommonChunksPlugin
7. Use only methods that are applicable in app , from libraries like lodash,momemt, lodash-webpack-plugin, moment-webpack-local plugin, known as dependency optimization.
8. Avoid inline functions in render, that will create new function every time, solution is to separate the component

Questions from S&P:

1. Did you used any patterns in your react application?  
2. Explain OOPS concepts  
3. Explain ES6 features  
4. Explain about Event Loop ?  
5. Promise vs async/await?  
6. Explain Components - class vs functional  
7. Question on state management - local state , context , redux.  
8. Write a code for class and functional components?  
9. HOC - How do you extend component ?  
10. Explain VDOM - in depth explanation  
11. What are the React Events ? Synthetic Events.  
12. React Profiler and purpose of it and do we have any performance issue with it ?  
13. How did you exclude react profile for production build?  
14. How did you do Error Handling in react?  
15. MVC vs Redux ?  
16. Explain Component Lifecycle ?  
17. How do you perform some operation before component rendering?   
18. How do you implement List component ? Importance of Keys   
19. How do you remove component ?  
20. How do we remove console logs from code across project?   
21. ES6 Modules import and export?  
22. How do you use different node version in your machine?   
23. can we write es6 and es5 code together?  
24. any features deprecated from es5 to es6?  
25. how do you make an api call from react?  
26. explain about flux pattern?  
27. explain closure in javascript?  
28. what are all performance optimization techniques used?  
29. Roles and Responsibilities ?  
30. what is JSX and purpose of it?

31. What are props in react?

32. How do you update state in class components?

33. what are refs and how do you update styles for react ?

34: How do you loop the arrays in react?

Q2. React Router  
Q3. How to prevent unauthorized users from accessing the content of the application?? Explain private routing with code implementation.  
Q4. Error Boundary - Its uses and how to build an error boundary component ?? The lifecycles methods used to build the error boundary component.  
Q5. There are 3 components Parent1, child1 and child2. If an error occurs in child2 then an error message should be  shown for child2 component only and rest of the component should render as it is without any error message. Similarly for child1,The error message will be completely different  from child2. We need to write the code for this using ErrorBoundry Component??  
Q6. useEffect and its uses.  
Q7. Some programs to check the useEffect concepts like how to make API calls on page load.  
Q8. Can we make API call outside useEffect?? What happens if we do so?  
Q9. Write a JS program to sum up the digits of the numbers. For example:- 2576 -> 2 + 5 + 7 + 6 -> 20 -> 2 + 0 -> 2 (Interviewer was expecting a one liner solution for this).  
Q10.Write a program to remove everything except even numbers from the array but you can not use a new array.  
    Ex: I/p:- [1,2,3,a,d,f,"#",4,6,4,4,")","("] => [2,4,6,4,4]( Interviewer was expecting a one liner solution for this)  
Q11.How react works behind the scene?? How it renders the UI?? Reconciliation, Virtual DOM and diffing algorithm??   
Q12.Context API and scenario based question. Create two component - Parent and child. Parent has a state num and child has a set of buttons to increment and decrement the num. You need to make it working without using props. Interviewer wanted me to write down all the react components for this.     
Q13.Tell me all the topics you know in React and what all you don't know/Learning as of now.

1. Brief Intro(Experience, current project and tech stack used).  
2. How do you handle authentication in react components.  
3. Private routing in react  
4. Implement a counter component(one Parent, two childs with decrement and increment functions) and optimize it  
5. Write code to use React Context API and some follow up questions.  
6. Tell me the effects you have used till date. Differences between useCallback and useMemo  
7. ES6 features and related concepts.

1. Explain SOLID principles  
2. Explain OOPS concepts  
3. Explain ES6 features  
4. Explain about Event Loop ?  
5. Promise vs async/await?  
6. Explain Components - class vs functional  
7. Question on state management - local state , context , redux.  
8. Explain about React Router?  
9. HOC - How do you extend component ?  
10. Explain VDOM - in depth explanation  
11. What are the React Events ? Synthetic Events.  
12. Explain about Component Unit Testing -  frameworks used.  
13. Where did you used Lazy loading?  
14. Error Handling in react?  
15. Explain about architectural patterns , MVC & MicroServices?  
16. Explain Component Lifecycle ?  
17. Ho do you perform some operation before component rendering?   
18. How do you implement List component ? Importance of Keys   
19. How do you remove component ?  
20. How do we remove console logs from code across project?   
21. ES6 Modules import and export?  
22. How do you use different node version in you machine?   
23. What is npm and why you need?  
24. Can i run react app without using node.js?  
25. Roles and Responsibilities ?

React Interview Question

Lead React MFE developer.

Panel R1: Satish Vadlakonda

1. Angular and React differences 2. Limitations of React 3. ES6 4. Module Systems 6. Git commit strategy 5. some system design questions based on ecommerce example

Lead React MFE developer.

Panel R1: Satish Vadlakonda

1. Strengths and weakness as a developer

2. Ecommerce site that has different components(header/footer/left nav/ right content) developed in different frameworks(react/angular/vue) - How will we design/develop - How can we communicate between the components - How can we use a common components(like a button) across different frameworks

3. What are the different module systems

4. Unit testing - difference between mock/stub/spy

5. State management - Explain Redux flow - Code example of communication between 2 parents what use the same child

Senior React MFE developer.

Panel R1: Nanda

1)Previous project architecture? 2)What are the hooks ? 3)How to improve the performance of the application ? 4)Write HTML to display cards side by side ? 5)Usememo vs. use callback ? 6)About React.lazy? 7)Next.js basic concepts ?

8)About common components ? 9)About performance loading ? 10)HOC and custom hooks implementation?

Senior React MFE developer.

Panel R1: Sabitha Kadari

HTML: • What are semantic tags? write down sample HTML for a blog page with sidebar and list of blogs • difference between <p>, <span> and <label> tags when to use which tag. • what are block, inline elements what are the differences • meta tag to achieve responsiveness with syntax CSS: • Positions and movement of an element with each position (relative, absolute, fixed and sticky) • various display properties and what is the differences between them • media queries for specific screens • grid and flexbox JS: • Recursion and use cases with example • Hoisting • difference between declaring variables with let and var • call, apply and bind methods • webComponents (agnostic) React: • class and functional components • Hooks you know and explain each of them • Custom Hook • what is props drilling, how to overcome the issue (context with example) • If a parent component have two children, how to re-render only 1 children • What are real dom and shadow dom • How the DOM will get updated • how to manage state globally without using redux (useContext and useReducer) Redux: • What is Redux • how the store updates the values • redux hooks • redux middle ware's you have worked on

Panel R2: Nanda

1) MicroFrontend using module federation, How the shell will render the MFE

2) past experience with Microfrontend and how you used to pass data between MFE's

3) questions on designing and how you divide a website into different components and why

4) if you want develop your epam.com website using MFE how many MFE will you create and why?

5) how to optimize the loading time of an MFE

6) useContext

7) Redux and middleware's you worked on and the differences. which one you choose to work with the most and why?

8) as redux is third party library with will take time to load compared to context(as it is built in). why you choose redux why not context

9) testing frameworks (asked about RTL)

10) typescript

Senior React developer.

Panel R1: Sandeep Chada

HTML & CSS

different position values and their behaviour

what is sass and how good you are in it

JS

what are closures

React

Ways to improve performance

different stages of react app

do we have alternate for jsx

props vs state

arrays vs lists in react

how can we decide right key to pass to list item

what are different ways of state management?

which one you choose flux vs redux and why?

Others

did u use oath

what is difference between access token and id token

code review steps you do

You have an app with so many fields in the form. How do you show to user? How do you prevent repeated clicks of button?

Panel R2: Mohammed Rafi

Below is a use case. How can we achieve this in react

5 components

2 components automatically rerender every 30 secs

3 components only on change of something

use of pure component?

can we make database call directly from react app? is it suggestible to do if possible?

what is MFE and what are the uses?

Questions specific to me

I served as Scrum master for 2 releases so asked about my role that time and how much hands-on I was doing

VBS is framework I worked so he asked what it and my work in that is

Senior React developer.

Panel R1+R2: Mohammed Rafi

2. About yourself, your tech stack , roles and responsibilities and recent projects

3. What is JSX . Why do we need it. Does browser understands JSX directly

4. Below is a use case. How can we achieve this in react

5 components that will call 5 different services to get data

2 components should automatically rerender after every 30 secs

3 components should pre-render on click of a button

Questions on implementation

5. Have you worked on RestAPI services backend or pure frontend?

6. What is synthetic event in react. Have you used in your project? What it will do?

7. What is HOC in react? How they are useful

8. Do you have prior experience with MFA? And, why do we need them?

Generic Questions:

9. If you are stuck with any problem /task what will you do? Have you experienced this scenario before and explain

Senior React developer.

Panel R1+R2: Mohammed Rafi

About yourself, your tech stack, roles and responsibilities and recent projects

What is VirtualDOM, have you used in your project?

Below is a use case. How can we achieve this in react

· 5 components that will call 5 different services to get data

· 2 components should automatically rerender after every 30 secs

· 3 components should rerender on click of a button

Have you worked on RestAPI services in backend?

How to embed 2 or more component as a single component, and how to communicate between them

Senior React developer.

Panel R1+R2: Mohammed Rafi

Here are the questions asked.

About yourself, your tech stack, roles and responsibilities and recent projects?

What is VirtualDOM, What is the advantage of VirtualDOM over regular DOM?

Below is a use case. How can we achieve this in react

· 5 components that will call 5 different services to get data?

· 2 components should automatically rerender after every 30 secs?

· 3 components should rerender based on the response of above components?

Have you worked on RestAPI services in backend?

What is React Portal?

How do you efficiently work with Forms in react?

React State management?

I have service which will give data of an employee and hierarchy, we need to display that data in ui exactly like teams organisation structure, how you will achieve that?

Mid-level React developer.

Panel R1: Sabitha

1. Semantic html(Tell some tags you have worked on) 2. Block level and inline elements differences 3. Basics of RTL(React testing library) 4. Types of functions in JS(Regular, function expressions, arrow functions) 5. Difference between Function.call() and Function.apply() 6. Arrow functions this keyword context with an example. Why do we use React over other frameworks (explain virtual dom and other advantages) 8. Functional and class-based components (Which do you prefer and why) 9. How do you pass state between parent and child components and follow up questions on state and props

10. React Context API(why do we use it and syntax for creating and using context) 11. Redux vs Context API 12. Which middleware have you used in Redux and why 13. One more scenario-based question on parent-child rendering and how to optimize it

Position: Mid level developer React

Panel R1 : Sandeep Chada

10. Could you share your past experiences and the technologies you've worked with?

11. Explain the concept of closures in JavaScript.

12. What factors contribute to React's popularity in comparison to other frameworks?

13. Define the virtual DOM in React.

14. How do you handle list rendering in React while using JSX?

15. Describe your strategies which you have followed in your past for enhancing the performance of React applications.

16. Discuss the importance of routing in web applications and its relevance to React.

17. Highlight the differences between class-based and functional components in React. Which do you prefer and why?

18. What is the role of Redux in a React application? Explain the data flow within a Redux application.

19. Define Higher Order Components (HOC) in React.

20. List various events available in React.

21. Differentiate between local storage and session storage in the context of web browsers.

22. Difference between using cookies and local storage as we can achieve the same stuff with local storage/ session storage as well?

23. Explore techniques for handling authentication and OAuth tokens on a web application's front-end.

24. JSX & component composition in React

Panel 2: Mohammed Rafi

25. Please provide an overview of your background, experiences, tech stack, and responsibilities.

26. Scenario-Based Question 1: Imagine a scenario where a parent component contains wrapped child components. Describe how you would implement a child component re-render when an event occurs in the parent component.

27. Explain Flux, its significance, and the reasons for its necessity.

28. Discuss why routing is essential for React apps.

29. Scenario-Based Question 2: Consider a scenario where multiple components make various API calls. Outline potential approaches for integrating these side effects seamlessly by implementing plug and play kind of component.

30. Have you been involved in end-to-end development, including deployments?

31. Scenario-Based Question 3: In a situation where errors occur within components, how would you log them? Have you encountered such scenarios before?

32. Scenario Based Question 4: If a crash occurs in the real DOM, does it impact the VDOM? Provide an explanation of the VDOM concept.

33. Tell more on the necessity of lazy loading in React applications.

34. Do you have a knowledge of the Git pipeline?

35. Do you have experience with Micro-Frontends (MFE)?

36. Have you worked with other frameworks like Ang

1. Eventbus communication in react:

<https://medium.com/@tenalisreenivas/communicate-between-independent-components-in-reactjs-deef90b55d9b>

#region EventBus Communication

EventBus communication in React is used to transfer data and messages between multiple components that have no direct relationship, known as inter-component communication.

EventBus is a publish/subscribe pattern, also known as pub/sub messaging pattern, that provides a medium for components in a large-scale or complex system to communicate with each other in a loosely coupled way.

Here is a simple example using a library called event-emitter:

1. `npm install events --save`

2. In Component #1, subscribe to an event:

```js

import EventEmitter from 'events'

componentDidMount(){

EventEmitter.subscribe('eventName', (data) => {

// code to handle event

})

}

```

3. In Component #2, publish an event:

```js

import EventEmitter from 'events'

const handleClick= ()=>{

EventEmitter.dispatch('eventName', 'eventData');

}

```

Note: It's good to understand that EventBus can lead to code that is hard to debug and maintain in the long run as applications grow since data flow is hard to trace. It also doesn't encourage good habits in state management. Therefore, other state management tools/platforms such as Redux, Mobx or even React’s own Context API and Hooks are commonly used in a large-scale React application for better state management.

It's best to use EventBus when the scope is limited or for some very specific requirements/scenarios.

#endregion Eventbus Communication

2. ReduxToolKit Query vs React-Query : https://www.basedash.com/blog/react-query-vs-rtk-query

3. Aggrid table binding, DatePicker

4. useMemo and useCallback

5. have you use Customhook in your project?

6. Webpack config for both development and Production

7. environment variables in react

<https://medium.com/@thejasonfile/using-dotenv-package-to-create-environment-variables-33da4ac4ea8f>

1. npm install dotenv --save
2. Next, add the following line to your app.
3. Const settings = require('dotenv').config()
4. Then create a .env file at the root directory of your application and add the variables to it.

// contents of .env

REACT\_APP\_API\_KEY = 'my-secret-api-key'

Usage:

require('dotenv').config();app.get('/', (req, res) => {  
 res.send(process.env.SECRET\_KEY);  
})app.listen(port, () => {  
 console.log(`Server is running on port ${port}.`)  
})

8. What is MicrofrontEnd.

9. Authentication in react

10. Explain how routing works in react

11.Lazy loading in react

12.Coding test in react routing

13.how many ways the Component communication in react

14.roles and responsibilities

15.CORS related questions

16.How to deploy react app in server

17.explain use of useQuery and useMutation hooks

18.Explain redux-saga and redux-thunk