

سوال اول: react state management

کامپوننت‌های React یک آبجکت build-in به نام state دارند. State ابزاری برای کپسوله‌سازی دیتاهایی است که برای رندر کامپوننت‌ها به آن‌ها احتیاج داریم. اپلیکیشن‌های React از چندین کامپوننت ساخته می‌شوند که ساختمان داده state و انتقال داده بین آن‌ها در آن‌ها به صورت داخلی کنترل می‌شود. این قضیه برای اپلیکیشن‌هایی با تعداد کامپوننت کم به راحتی هندل می‌شود اما هنگامی که برنامه ما بزرگ‌تر می‌شود، پیچیدگی مدیریت State هایی که بین چندین کامپوننت به اشتراک گذاشته شده‌اند، بیشتر می‌شود. در این موارد دیگر انتقال دیتا بین کامپوننت‌ها به سادگی استفاده از props و رابطه parent/child نمی‌تواند انجام شود و حتی با استفاده از راه حل هایی مانند state lifting up برای کامپوننت‌های تو در تو به مشکل برمی‌خوریم. پس برای حل مشکلات این چنینی به react state management احتیاج داریم. روش‌هایی مانند redux توسعه داده شدند تا این مشکل را حل کنند. در این روش یک منبع مرکزی برای ذخیره کردن state های تمام کامپوننت‌ها ارائه می‌شود که تمام کامپوننت‌ها می‌توانند به آن دسترسی داشته باشند. روش‌های دیگر مانند استفاده از hook به نام useReducer از کانسپت مشابه redux پیروی می‌کنند.

<https://www.loginradius.com/blog/async/react-state-management/>

<https://www.youtube.com/watch?v=35lXWvCuM8o>

سوال دوم: functional components

Functional components نوعی دیگر از پیاده‌سازی اجزای UI یک اپلیکیشن در کنار class component ها است. در حقیقت همان فانکشن‌های جاوااسکریپتی هستند که می‌توانند به صورت آپشنال، ورودی‌هایی را در قالب آبجکت props دریافت کنند و خروجی را در قالب jsx یا react node ارائه کنند. functional component ها رویکردی دیگر تحت عنوان functional programming را در برابر شی گرایي که class component ها در اختیار ما قرار می‌دهند، دنبال می‌کند. استفاده از

منطق فانکشنال به مراتب راحت تر است. در نسخه 16.8 به بعد react، برای پیاده سازی فیچرهایی مثل state، دیگر احتیاج به نوشتن class نیست و hook ها می توانند این ویژگی ها را در function component به ما بدهند. همچنین استفاده از function component ها به جای class در کانتکست استفاده از وب پیک و تبدیل به bundle روشی بهینه تر است.

<https://www.youtube.com/watch?v=Cla1WwguArA>

https://www.youtube.com/watch?v=NJ_qbsLf52w

<https://www.youtube.com/watch?v=yc6elaGOoGQ>

سوال سوم: React در Hooks

Hooks فیچرهای جدیدی هستند که در نسخه react 16.8 اضافه شده است و ابزارهایی هستند که اجازه می دهند state ها و فیچرهای دیگر react را بدون نیاز به نوشتن class در فانکشن پیاده سازی کنیم. به عنوان مثال useState یک hook است که به کمک آن می توانیم ساختمان داده state را در functional component شبیه سازی کنیم یا مثال دیگر hook effect است که اجازه شبیه سازی فانکشن های life cycle را که در class component در دسترس هستند، در function component به ما میدهد. پس می توان hook ها را فانکشن هایی تعریف کرد که اجازه می دهند از ویژگی های مربوط به react state و life cycle در functional component ها استفاده کنیم و کمک می کنند از react بدون استفاده از کلاس و با رویکرد فانکشنال بهره برداری کنیم.

<https://reactjs.org/docs/hooks-state.html>

<https://reactjs.org/docs/hooks-intro.html>

<https://reactjs.org/docs/hooks-overview.html>

<https://www.geeksforgeeks.org/reactjs-functional-components/>