

۱)

مدیریت دولتی یکی از مهمترین جنبه های هر برنامه است. وضعیت برنامه تعیین می کند که کاربران چه چیزی را ببینند، برنامه چگونه ظاهر شود، چه داده هایی ذخیره شود و غیره. بنابراین جای تعجب نیست که بسیاری از کتابخانه های منبع باز وجود دارند که به طور خاص برای آسان تر کردن مدیریت دولتی و لذت بخش تر طراحی شده اند. از میان انبوه چارچوب های جاوا اسکریپت `UI`، `React` احتمالاً یکی از پر جنب و جوش ترین اکوسیستم ها، از جمله کتابخانه های مدیریت دولتی است. این مقاله ۶ کتابخانه برتر را بررسی می کند و آنها را با هم مقایسه می کند تا ببیند کدام یک بهترین انتخاب برای برنامه بعدی `React` شما است.

همه آن متغیرهایی که تعیین می کنند برنامه شما در چه وضعیتی است باید در جایی ذخیره شوند. بنابراین مدیریت حالت یک اصطلاح گسترده است که نحوه ذخیره ایالت و نحوه تغییر آن را ترکیب می کند.

`React` و اکوسیستم آن راه های مختلفی را برای ذخیره و مدیریت آن حالت ارائه می دهند. و وقتی من می گویم مقدار زیادی منظورم `LOTS` است.

برای ذخیره سازی می توانید...

آن متغیرها را در حالت مؤلفه محلی نگه دارید - چه با قلاب ها (`useState` یا `useReducer`) یا در کلاس ها (`this.state` و `this.setState`)

با استفاده از یک کتابخانه شخص ثالث مانند `Redux`، `MobX`، `Recoil` یا `Zustand`، داده ها را در یک فروشگاه نگهداری کنید.

حتی می توانید آنها را روی شی پنجره به صورت سراسری نگه دارید

`React` اهمیتی نمی دهد که داده ها را کجا قرار می دهید اما

برای تعاملی کردن برنامه خود، به روشی نیاز دارید که `React` بداند چیزی تغییر کرده است و باید برخی (یا همه) مؤلفه ها را دوباره در صفحه نمایش دهد.

زیرا `React`، علیرغم نامش، مانند برخی از فریمورک های دیگر «واکنشی» نیست.

برخی از فریمورک ها چیزها را «مراقب» می کنند و بر اساس آن به روزرسانی می شوند. `Angular`، `Svelte` و `Vue` این کار را در میان دیگران انجام می دهند.

اما واکنش نشان نمی دهد. "مواظب تغییرات" نیست و به طور جادویی دوباره رندر می شود. شما (یا چیزی) باید به آن بگویید که این کار را انجام دهد.

با `useState`، `useReducer`، یا `this.setState` (کلاس ها)، زمانی که یکی از توابع تنظیم کننده را فراخوانی کنید، React دوباره رندر می شود.

اگر داده ها را در `Redux`، `MobX`، `Recoil` یا فروشگاه دیگری نگه دارید، آن فروشگاه زمانی که چیزی تغییر کرد به React می گوید و رندر مجدد را برای شما فعال می کند.

اگر تصمیم دارید داده ها را به صورت سراسری در پنجره نگه دارید، باید به React بگویید تا پس از تغییر داده ها، به روزرسانی شود.

اوه، و برای اینکه کاملاً واضح باشد، توصیه نمی کنم وضعیت خود را به صورت جهانی در پنجره نگه دارید، به دلایل معمولی که باید از داده های جهانی اجتناب شود. کدهای نامرتب، استدلال کردن در مورد آن سخت است، و غیره. من فقط به آن اشاره می کنم که این امکان پذیر است، تا به این نکته اشاره کنم که React واقعاً اهمیتی نمی دهد که داده های آن از کجا آمده است.

۲)

یک کامپوننت Function همچنین HTML را برمی گرداند و تقریباً مانند یک جزء کلاس رفتار می کند، اما مؤلفه های تابع را می توان با استفاده از کد بسیار کمتری نوشت، درک آن آسان تر است.

کامپوننت Functional تابعی است که `props` می گیرد و `JSX` را برمی گرداند. آنها روش های حالت یا چرخه حیات ندارند. خواندن، اشکال زدایی و آزمایش اجزای عملکردی آسان تر است. آنها مزایای عملکرد، کاهش جفت و قابلیت استفاده مجدد بیشتر را ارائه می دهند.

۳)

هوک ها در نسخه ۱۶.۸ به React اضافه شدند.

هوک ها به اجزای تابع اجازه دسترسی به حالت و سایر ویژگی های React را می دهند. به همین دلیل، اجزای کلاس معمولاً دیگر مورد نیاز نیستند.

اگرچه Hooks به طور کلی جایگزین اجزای کلاس می شود، اما هیچ برنامه ای برای حذف کلاس ها از React وجود ندارد.

Hooks allow us to "hook" into React features such as state and lifecycle methods.

باید Hooks را از react وارد کنید.

در اینجا ما از useState Hook برای پیگیری وضعیت برنامه استفاده می کنیم.

State به طور کلی به داده ها یا ویژگی های برنامه ای اشاره دارد که باید ردیابی شوند.

۳ قانون برای hooks وجود دارد:

هوک ها را فقط می توان در داخل اجزای تابع React فراخوانی کرد.

هوک ها را فقط می توان در سطح بالای یک جزء فراخوانی کرد.

هوک ها نمی توانند مشروط باشند

اگر منطق حالتی دارید که نیاز به استفاده مجدد در چندین مؤلفه دارد، می توانید Hook های سفارشی خود را بسازید.