React state management

Stateدرون کامپوننت مدیریت می‌شود، همان طور که متغیرها درون یک تابع اعلان می‌شوند . Stateدر کامپوننت React در واقع حالت محلی خودش است، یعنی حالت نمی‌تواند خارج از کامپوننت مورد دسترسی یا تغییر قرار گیرد و تنها درون آن کاربرد دارد. این وضعیت شبیه تابعی است که حیطه محلی خود را دارد.

چند نکته مهم در مورد state

Stateرا به صورت مستقیم دستکاری نکنید

منظور از این نکته آن است که نباید state را به صورت مستقیم تغییر داد، زیرا مانند مثال زیر باعث می‌شود که کامپوننت به صورت خودکار رندر مجدد نشود:

```
This.state.name='sam';
```

به‌روزرسانی‌های State می‌توانند ناهمگام باشند

Reactممکن است چندین فراخوانی setState را به منظور افزایش عملکرد با هم در یک به‌روزرسانی تجمیع کند.

گردش رو به پایین داده‌ها
نه کامپوننت‌های والد و نه کامپوننت‌های فرزند نمی‌توانند بدانند که یک کامپوننت خاص با حالت (stateful)یا بی حالت (stateless) است و اهمیتی هم نمی‌دهند که حالت به صورت تابع یا کلاس تعریف شده است. این مشخصات جز برای کامپوننتی ک مالک است و آن را تنظیم می‌کند قابل دسترسی نیستند. به همین دلیل است که state در اغلب موارد، محلی (local) یا کپسوله (encapsulated)نامیده می‌شود.

یک کامپوننت می‌تواند انتخاب کند که حالت خود را به صورت props به کامپونت‌های فرزندش ارسال کند.

در نتیجه مدیریت state ها سخت می شود و می توانیم از کتابخانه ای به اسم Redux استفاده کنیم این کتابخانه حالات (states) مختلف را در برنامه‌های جاوا اسکریپت مدیریت می‌کند.

ریداکس یک الگو برای مدیریت بهتر و بهینه‌تر وضعیت‌های مختلف در برنامه ارائه می‌دهد. این کتابخانه سن زیادی ندارد و در سال ۲۰۱۵ عرضه شده است. از Redux بیشتر به عنوان یک مکمل در کنار کتابخانه‌هایی مانند Angular یا React استفاده می‌شود. این کتابخانه با تمام وابستگی‌های خود (Dependency) تنها ۲ کیلوبایت حجم دارد و با بکارگیری آن، لازم نیست نگران سنگین شدن پروژه خودمان باشیم .

Functional Components

دو نوع کامپوننت در React وجود داره Class Components و Functional Components. Class components در واقع کلاس های ES6 هستند و Functional Componentsتوابع هستند

خوانایی فانکشنال کامپوننت ها بالاتر از کلاس کامپوننت هاست (چون همون توابع ساده جاواسکرپیتی هستند)

تست کردن فانکشنال کامپوننت ها ساده تر از کلاس کامپوننت هاست.

Functional Componentsعملکرد بهتری دارند

اگر ما در function components از state ها استفاده نکنیم،کامپوننت ها راحت تر پیاده سازی می شوند و خیلی راحت در قسمت های دیگه پروژه و پروژه های دیگه قابل استفاده هستند. ری اکت برای تبدیل کردن کدها به یک نسخه قابل قبول در جاوا اسکریپت نیاز به یک کامپایلر دارد . این کامپایلر به نام babel شناخته شده.

اگر ما کدهای کلاس کامپوننت ها را با توابع در babel مقایسه کنیم ، متوجه میشیم که کلاس ها خروجی بسیار سنگینی نسبت به توابع دارند

Hook in react

تو تمرین هایی که انجام دادیم دیدیم (this.setState) تنها راه تغییر state است اما در نسخه جدید قابلیتی به نام hook در ری اکت اضافه شده است که به ما اجازه می دهد state را از طریق کامپوننت های کاربردی (functional) نیز تغییر دهیم.

Hookها توابعی ساده هستند که این امکان را به ما می دهند از قابلیت های state و lifecycle در react بدون استفاده از ساختار class در کامپوننت هایی که به صورت تابع هستند استفاده کنیم .Hook ها در داخل class ها غیر قابل استفاده می باشند. React تعدادی Hook همانند useStateرا به صورت پیش فرض تهیه کرده است که می توان از آن استفاده کرد.

useState یک آرایه با دو عنصر بر می گرداند که عنصر اول مقدار فعلی state و عنصر دوم یک تابع برای بروزرسانی مقدار state می باشد. این تابع شبیه this.setState می باشد با این تفاوت که state قبلی و جدید را با هم ادغام نمی کند.useState تنها یک پارامتر به عنوان مقدار اولیه state می گیرد.

از state hook می توانیم چندین بار در یک کامپوننت استفاده کنیم.

منابع :

reactjs.org- redux.js.org
virgool.io- blog.faradars.org