

Redux

ریداکس چیست؟

فهرست مطالب

۱. ریداکس چیست؟
۲. اجزا و روش کار ریداکس
۳. چه زمانی از Redux استفاده کنیم؟
۴. Redux با کدام زبان‌ها و فریم‌ورک‌ها سازگار است؟
۵. مراحل ادغام Redux در React App
۶. اهمیت Redux برای برنامه‌های کاربردی وب
۷. مزایای استفاده از ریداکس چیست؟
۸. معایب ریداکس

در چشم‌انداز همیشه در حال تحول توسعه وب، ایجاد تجربیات یکپارچه و کارآمد برای کاربر همچنان اولویت اصلی است. با افزایش پیچیدگی برنامه‌های مدرن، مدیریت وضعیت آن‌ها به طور فزاینده‌ای چالش‌برانگیز می‌شود. اینجاست که ریداکس، یک ابزار قدرتمند مدیریت، برای ساده‌سازی فرایند وارد عمل می‌شود.

در این مقاله، ما شناختی جامع را در قلمرو ریداکس آغاز خواهیم کرد و با پاسخ به سوال ریداکس چیست به نقش محوری آن در برنامه‌های کاربردی وب خواهیم پرداخت و راهنمای گام‌به‌گام ادغام Redux در برنامه‌های React را بررسی خواهیم کرد. چه یک توسعه‌دهنده مبتدی باشید که وارد دنیای مدیریت عمومی می‌شوید یا یک حرفه‌ای باتجربه که به دنبال درک عمیق‌تر پیچیدگی‌های Redux هستید، امیدواریم این راهنما برای شما مفید باشد.

ریداکس چیست؟



اولین چالش ما این است که اصلاً ریداکس چیست؟ ریداکس یک کتابخانه بسیار محبوب است که مدیریت حالت را در برنامه‌ها تسهیل می‌کند.

مدیریت وضعیت یک برنامه با چند مؤلفه آسان است؛ اما با افزایش مؤلفه‌ها و بزرگ‌تر شدن برنامه، حفظ وضعیت هر مؤلفه برنامه شما بسیار دشوار می‌شود. بنابراین Redux برای چنین برنامه‌های بزرگی به کمک می‌آید که در آن یک ظرف یا store ایجاد می‌کند که وضعیت برنامه را داشته باشد و هر مؤلفه‌ای که به آن نیاز داشته باشد، می‌تواند آن را از آن‌جا بدون انتقال لوازم از یک مؤلفه به مؤلفه دیگر منتقل کند.

```
graph TD; Action((Action)) -.-> Dispatcher[DISPATCHER]; Dispatcher --> Reducer[REDUCER]; State((State)) --> Reducer; Reducer --> STATE[STATE]; STATE --> View[VIEW]; STATE -.-> State2((State)); State2 --> Event((Event)); Event --> ACTIONS[ACTIONS]; ACTIONS -.-> Dispatcher;
```

٦

- **Redux store**: در ریداکس "store" یک مکان متمرکز است که کل وضعیت و داده‌های برنامه را در خود نگه می‌دارد. این جایی است که برنامه، برای مثال نام کاربری شما را ذخیره می‌کند تا از آن در مناطق مختلف استفاده کند و تنها راه برای تغییر حالت ذخیره شده این است که از یک "reducer" استفاده کنید.
- **Redux reducer**: تابعی است که نحوه تغییر وضعیت برنامه و برگرداندن حالت جدید را توضیح می‌دهد؛ اما چگونه می‌داند که کدام بخش از مجموعه را تغییر دهد؟ این جایی است که "action" به تصویر می‌رسد.
- **Redux action**: یک "action" فقط یک شیء جاوا اسکریپت است که شامل دو بخش است. بخش اول نوعی ویژگی است که نوع عمل و آنچه را که قرار است انجام دهد، توصیف می‌کند. (به عنوان مثال، توضیح می‌دهد که کاربر نام کاربری خود را وارد کرده است) و بخش دوم محموله‌ای از داده‌ها است که حاوی داده‌های واقعی مانند نام کاربری می‌باشد. با این حال، action به‌خودی‌خود کاری انجام نمی‌دهد. ما باید این عمل را با فراخوانی متدی به نام dispatch و ارسال یک شیء action به یک کاهش‌دهنده انجام دهیم.

چه زمانی از Redux استفاده کنیم؟



بسیاری از مردم در مورد زمان استفاده از Redux در پروژه‌های خود سردرگم می‌شوند؛ اما اینکه معیار استفاده از ریداکس چیست، بستگی به شما دارد که آیا به آن نیاز دارید یا خیر. استفاده از Redux زمانی لازم است که برنامه شما رشد کند و مدیریت وضعیت برنامه دست‌وپاگیر شود، اینجاست که Redux store ظاهر می‌شود و این زمانی است که توسعه‌دهندگان تمایل دارند ابزارها و کتابخانه‌هایی را جستجو کنند که بتوانند برنامه‌های خود را به راحتی مقیاس‌بندی کنند و وضعیت هر جزء را مدیریت کنند.

اگر لازم باشد داده‌ها از والد به فرزند در اعماق درخت منتقل شوند، این کار همچنان با استفاده از ابزارهای React مانند Context قابل انجام است؛ اما وقتی نوبت به اشتراک گذاری حالت بین اجزا در یک سطح می‌رسد، ریداکس گزینه اجتناب‌ناپذیری است.

Redux برای حفظ و به‌روزرسانی وضعیت چندین مؤلفه برای به اشتراک گذاشتن در برنامه شما با وجود استقلال مؤلفه‌ها، استفاده می‌شود. علاوه بر این، redux زمانی استفاده می‌شود که وضعیت برنامه به طور مکرر تغییر می‌کند؛ مثلاً در جایی که تعامل کاربر با وب‌سایت یا برنامه‌ای مانند وب‌سایت تجارت الکترونیک، در فرایند پرداخت وجود دارد.

Redux با کدام زبان ها و فریم ورک ها سازگار

است؟

ریداکس معمولاً با React با استفاده از React Redux استفاده می شود؛ اما می تواند با سایر چارچوب های UI مانند Angular، Vue.js و vanilla Javascript نیز استفاده شود. با این حال هر دو react و redux با هم استفاده می شوند؛ اما شایان ذکر است که آن ها مستقل از یکدیگر هستند. این به این دلیل است که فقط React به ریداکس اجازه می دهد تا یک جریان یک طرفه داده ها را انجام دهد. این بدان معناست که داده ها را نمی توان از یک فرزند به والد ارسال کرد.

باید از والد به سمت فرزند جریان یابد. این مدل فکری با ریداکس بسیار خوب کار می کند (جایی که ما نمی توانیم مستقیماً وضعیت را تغییر دهیم). در عوض، اقداماتی را که قصد تغییر حالت دارند، ارسال می کنیم و سپس به طور جداگانه، تغییرات حالت حاصل را مشاهده می کنیم.

مراحل ادغام Redux در React App



در توسعه وب مدرن، ادغام Redux با یک برنامه React به طور قابل توجهی مدیریت حالت‌های برنامه پیچیده را افزایش می‌دهد و فرایند یکپارچه‌سازی روان و سیستم مدیریت عمومی قوی را تضمین می‌کند. در ادامه مقاله ریداکس چیست مراحل ادغام این کار به صورت کلی بیان گردیده است:

نصب و راه‌اندازی

ادغام ریداکس در یک برنامه React با مرحله اساسی نصب و راه‌اندازی آغاز می‌شود و زمینه را برای یک سیستم مدیریت دولت ساختاریافته فراهم می‌کند. نصب Redux شامل پی‌کربندی بسته‌های لازم و ایجاد محیط بنیادی برای فعال کردن مدیریت حالت متمرکز در برنامه React است. فرایند نصب و راه‌اندازی سفر به سمت مدیریت متمرکز عمومی را آغاز می‌کند. این مرحله را برای ادغام Redux در برنامه React تنظیم می‌کند و یک store متمرکز را برای مدیریت کارآمد وضعیت و تسهیل جریان داده‌های ساختاریافته در سراسر برنامه ارائه می‌دهد.

ایجاد یک reduxc store

ایجاد یک Redux store یک فرایند حیاتی در زمینه‌سازی برای مدیریت بهتر ساختار در یک برنامه React است. این به عنوان یک مخزن متمرکز برای مدیریت و حفظ وضعیت کل برنامه عمل می‌کند. Store ریداکس که با تابع createStore ساخته شده است، ساختار اصلی را تشکیل می‌دهد که از یک منبع حقیقی واحد برای داده‌های برنامه اطمینان می‌دهد و تغییرات حالت کارآمد را ممکن می‌سازد.

ایجاد Redux store نقش اساسی در معماری برنامه دارد. این یک فضای متمرکز برای مدیریت تغییرات حالت، ترویج یک رویکرد ساختاریافته و قابل پیش‌بینی برای مدیریت حالت در برنامه React فراهم می‌کند. این رویکرد ساختاریافته به توسعه‌دهندگان این امکان را می‌دهد تا به طور مؤثر جریان داده‌های برنامه را مدیریت کنند و از فرایند توسعه سازمان‌یافته‌تر و مقیاس‌پذیرتر اطمینان حاصل کنند.

تعریف Actions و Action Creators

تعریف اکشن‌ها و ایجاد سازندگان اکشن در پیاده‌سازی Redux در یک برنامه React بسیار مهم است. Action به عنوان پلی برای به‌روزرسانی وضعیت برنامه عمل می‌کنند، درحالی‌که action creators، توابعی هستند که مسئول تولید اکشن هستند. تعریف اکشن‌ها و سازندگان مربوط به آن‌ها، یک رویکرد ساختاریافته برای مدیریت تغییرات حالت ایجاد می‌کند. این فرایند سیستماتیک جریان قابل پیش‌بینی داده را تضمین می‌کند و توسعه‌دهندگان را قادر می‌سازد تا تغییرات حالت را در برنامه به طور مؤثر مدیریت کنند.

نوشتن کاهش‌دهنده‌ها

نوشتن کاهش‌دهنده‌ها در معماری Redux بسیار مهم است و نحوه تغییر وضعیت برنامه را بر اساس اقدامات ارسال شده تعریف می‌کند. کاهش‌دهنده‌ها توابع خالصی هستند که مسئول انتقال وضعیت هستند؛ ولی نقش کاهش‌دهنده در ریداکس چیست؟ کاهش‌دهنده‌ها ستون فقرات مدیریت تغییرات حالت در Redux را تشکیل می‌دهند.

تعریف و ترکیب مناسب آن‌ها یک رویکرد ساختاریافته و سازمان‌یافته برای مدیریت حالت در برنامه React را تضمین می‌کند. این فرایند دقیق، سیستم مدیریت عمومی قابل‌پیش‌بینی‌تر، مقیاس‌پذیرتر و کارآمدتر را تضمین می‌کند.

اتصال Redux به React Components

اتصال Redux به React Components یک پیوند حیاتی بین Redux store و نمای React ایجاد می‌کند. اتصال Redux به اجزای React برای یکپارچگی منسجم ضروری است، این کار نماهای React را قادر می‌سازد تا به حالت متمرکز مدیریت‌شده توسط Redux دسترسی داشته باشند و با آن تعامل برقرار کنند و جریان داده و واکنش‌پذیری را در برنامه افزایش دهند.

عملیات انتقال

ارسال action‌ها در یک برنامه Redux برای اعمال تغییرات حالت بر اساس تعاملات کاربر یا رویدادهای خاص اساسی است. این شامل ارسال action‌ها به Redux store و ایجاد تغییرات در وضعیت برنامه است. در پاسخ به این که تاثیر این کار در ریداکس چیست می‌توان گفت اقدامات Dispatching مؤلفه‌های React را قادر می‌سازد تا تغییرات کنترل‌شده را در حالت برنامه مدیریت‌شده توسط Redux آغاز کنند. این یک پیوند مهم بین تعاملات کاربر و تغییرات حالت ایجاد می‌کند و جریان داده قابل‌پیش‌بینی و ساختاریافته را تضمین می‌کند.

دسترسی به وضعیت در کامپوننت‌ها

دسترسی به وضعیت برنامه در اجزای React پس از ادغام Redux بسیار مهم است. دسترسی به حالت در اجزای React، استفاده مؤثر از حالت متمرکز مدیریت شده توسط Redux را تضمین می‌کند و واکنش‌پذیری برنامه و قابلیت‌های مدیریت داده را افزایش می‌دهد.

پیاده‌سازی میان‌افزار (اختیاری)

پیاده‌سازی میان‌افزار در ریداکس، قابلیت‌های بیشتری را فراهم می‌کند و یک مرحله اختیاری برای افزایش قابلیت‌های Redux store است. میان‌افزار امکان رهگیری اقدامات و انجام کارهای اضافی را فراهم می‌کند. یکپارچه‌سازی میان‌افزار که فعالیتی اختیاری است، عملکردهای Redux store را غنی‌تر می‌کند و رویکرد جامع‌تری فراتر از مدیریت حالت اولیه برای الزامات برنامه‌های کاربردی پیچیده‌تر ارائه می‌دهد.

استفاده از Redux DevTools برای اشکال‌زدایی

استفاده از Redux DevTools یک محیط پیشرفته برای اشکال‌زدایی و بررسی تغییرات وضعیت برنامه و تسهیل فرایند اشکال‌زدایی ارائه می‌دهد و توسعه‌دهندگان را با ابزارهای روشنگری، برای عیب‌یابی کارآمد و بهینه‌سازی برنامه توانمند می‌کند. ادغام Redux در یک برنامه React شامل یک سری مراحل اساسی است که هر کدام به یک سیستم مدیریت عمومی ساختاریافته و قابل مدیریت کمک می‌کند. هر مرحله نقش مهمی در ایجاد یک محیط منسجم و سازمان‌یافته برای مدیریت جریان داده برنامه ایفا می‌کند.

به طور کلی، رویکرد سیستماتیک مشخص شده توسط این مراحل، یک سیستم مدیریت حالت کارآمد و قابل پیش بینی را در برنامه React تضمین می کند و توسعه دهندگان را قادر می سازد تا برنامه های کاربردی وب مقیاس پذیر، کارآمد و قابل نگهداری ایجاد کنند.

اهمیت Redux برای برنامه های کاربردی وب



ریداکس، یک کانتینر حالت قابل پیش بینی، به عنوان یک راه حل حیاتی در مدیریت پیچیدگی های حالت در برنامه های وب، به ویژه در کنار چارچوب هایی مانند React ظاهر شده است. اهمیت آن در توانایی آن برای ساده کردن مدیریت حالت، افزایش قابلیت پیش بینی و مقیاس پذیری در توسعه برنامه است در پاسخ به این که معیارهای اهمیت ریداکس چیست، می توان به موارد زیر اشاره کرد:

- **مدیریت عمومی متمرکز:** ریداکس یک منبع واحد حقیقی ایجاد می کند، یک فروشگاه متمرکز که در آن کل حالت برنامه در آن قرار دارد که این تمرکز دسترسی و تغییر حالت را در سراسر برنامه ساده می کند.
- **تغییرناپذیری و پیش بینی پذیری:** ریداکس پذیرفتن داده های تغییرناپذیر، اجرای جریان داده های یک طرفه و تغییرات وضعیت

قابل پیش بینی را تسهیل می کند و از اشکال زدایی آسان تر و ردیابی خطای جامع اطمینان حاصل می کند.

- **انعطاف پذیری میان افزار:** ریداکس پشتیبانی از میان افزار را ارائه می دهد و توسعه دهندگان را قادر می سازد تا قابلیت های اضافی مانند ورود به سیستم، مسیریابی و رسیدگی به درخواست های ناهم زمان را یکپارچه کنند.

- **قابلیت های بهبود یافته اشکال زدایی:** توسعه دهندگان از ابزارهای اشکال زدایی قدرتمندی مانند Redux DevTools بهره می برند که اشکال زدایی در سفر را برای بررسی و ردیابی وضعیت های برنامه در مقاطع زمانی مختلف تسهیل می کند.

- **مقیاس پذیری و جذب استعداد:** مهارت در Redux در هنگام استخدام توسعه دهندگان React JS به یک دارایی تبدیل می شود. کسانی که در Redux باتجربه هستند، درک درستی از مدیریت حالت های کاربردی پیچیده دارند. این درک آنها را در پروژه هایی که نیاز به مدیریت پیچیده ایالتی دارند، تبدیل به یک مکمل ارزشمند می کند.

مزایای استفاده از ریداکس چیست؟



حال بحث این است که با تمام این ویژگی ها، مزایای ریداکس چیست؟
مزایای Redux به شرح زیر هستند:

- **قابلیت پیش‌بینی:** در برنامه‌های ریداکس، حالت همیشه قابل‌پیش‌بینی است؛ زیرا کاهنده‌ها توابع خالص هستند. بنابراین، اگر همان عملکرد و حالت را به آن‌ها بدهید، همان نتیجه را برمی‌گردانند. علاوه بر این، وضعیت تغییرناپذیر است که منجر به اشکال‌زدایی و برنامه‌نویسی ساده می‌شود؛ زیرا داده‌هایی که هرگز تغییر نمی‌کنند راحت‌تر از داده‌هایی هستند که به طور تصادفی در سراسر برنامه تغییر می‌کنند. همچنین قابلیت بازیابی حالت‌های قبلی و دریافت نتایج را در زمان واقعی دارد.
- **قابلیت نگهداری:** ریداکس در مورد ساختار خود و این‌که چگونه کد باید در یک برنامه سازماندهی شود کاملاً سخت‌گیر است. این ویژگی درک ساختار هر برنامه را برای توسعه‌دهندگانی که دانش عمیقی از redux دارند، آسان‌تر می‌کند. این محدودیت ساختار، redux را بسیار قابل نگهداری می‌کند و به جداسازی منطق تجاری از درخت مؤلفه کمک می‌کند. با این حال، برای برنامه‌های بزرگ که نیاز به به‌روزرسانی‌های مکرر دارند، redux کاملاً امکان‌پذیر است.
- **اشکال‌زدایی آسان:** ریداکس ابزارهای توسعه‌دهنده عالی را ارائه می‌دهد که برای اهداف اشکال‌زدایی استفاده می‌شود. در مورد برنامه‌های کاربردی بزرگ، اشکال‌زدایی زمان بیشتری نسبت به توسعه برنامه زمان می‌برد. علاوه بر این، ثبت اقدامات و وضعیت‌ها نیز به شناسایی خطاهای کدگذاری یا هر نوع باگ در برنامه کمک می‌کند.
- **عملکرد بهتر:** مردم ممکن است فکر کنند که گلوبال نگه داشتن وضعیت یک برنامه و نگه داشتن آن در Redux store باعث کاهش

عملکرد برنامه می‌شود. درحالی که این‌طور نیست. کتابخانه binding UI از react که به‌عنوان React Redux شناخته می‌شود، چندین بهینه‌سازی عملکرد را پیاده‌سازی می‌کند، به طوری که یک جزء یا یک مؤلفه متصل تنها در صورت نیاز، دوباره ارائه می‌شود.

- **سهولت تست:** برنامه‌های Redux را می‌توان به راحتی آزمایش کرد؛ زیرا می‌توان از توابع برای تغییر وضعیت توابع خالص استفاده کرد.
- **رندر سمت سرور:** رندر سمت سرور نیز با ریداکس امکان‌پذیر است. رندر اولیه، برنامه را با ارسال وضعیت به سرور همراه با پاسخ به درخواست سرور انجام می‌دهد. این ویژگی redux آن را به توسعه‌دهندگان دیگر کتابخانه‌های مدیریت عمومی ترجیح می‌دهد.

معایب ریداکس

- اگرچه ریداکس دارای مزایای زیادی است که آن را به توسعه‌دهندگان ترجیح می‌دهد، با این حال، معایبی نیز دارد که باید به آن‌ها توجه شود:
- **افزایش پیچیدگی:** ریداکس در هنگام استفاده از اکشن‌ها و کاهش‌دهنده‌ها، پیچیدگی بیشتری در لایه‌بندی دارد.
 - **طراحی محدود:** طراحی محدود و گزینه‌های بسیار کمی دارد.
 - **استفاده بیش از حد از حافظه:** از آنجایی که وضعیت تغییرناپذیر است، بنابراین وقتی یک وضعیت به‌روز می‌شود، کاهنده هر بار باید حالت جدیدی را برگرداند که منجر به استفاده بیش از حد از حافظه در درازمدت می‌شود؛ بنابراین، هرچه ذخیره بزرگ‌تر باشد، میزان استفاده از حافظه بیشتر است.

• **وقت‌گیر:** ریداکس برای برنامه‌های بزرگ‌تر بسیار مناسب است، با این حال، برای برنامه‌های کوچک یا متوسط، می‌تواند بسیار وقت‌گیر باشد؛ زیرا از قبل به ورودی‌های بیشتری نیاز دارد.

سخن آخر

ریداکس به عنوان یک ابزار اساسی در توسعه وب مدرن، ارائه یک رویکرد ساختاریافته و متمرکز برای مدیریت عمومی است. اصول اصلی آن، از جمله کنترل حالت متمرکز، تغییرناپذیری و پشتیبانی از میان‌افزار، در ساده‌سازی مدیریت حالت‌های برنامه پیچیده بسیار مهم هستند. ریداکس نقشی اساسی در تضمین استحکام و مقیاس‌پذیری برنامه‌ها ایفا می‌کند. توانایی آن در ارائه یک رویکرد ساختاریافته و قابل‌پیش‌بینی برای مدیریت وضعیت برنامه، روند توسعه را ارتقا می‌دهد و آن را به یک دارایی ضروری در ایجاد برنامه‌های کاربردی وب مدرن و پیچیده تبدیل می‌کند. در نظر گرفتن استخدام توسعه‌دهندگان React JS با سابقه برای هدایت و مدیریت مؤثر ساختارهای برنامه پیچیده بسیار مهم است. امروزه از React Redux به عنوان کتابخانه رسمی رابط کاربری برای React استفاده می‌شود که بهینه‌سازی‌های عملکردی زیادی را ارائه می‌دهد به طوری که یک جزء فقط در صورت نیاز دوباره ارائه شود.