

Voici un résumé éducatif sur **Redux normal** et **Redux Toolkit**, conçu pour une compréhension simplifiée :

Qu'est-ce que Redux ?

Redux est une bibliothèque JavaScript utilisée pour gérer l'état (les données) d'une application de manière centralisée. Cela permet à toutes les parties d'une application d'accéder facilement à l'état, et de le modifier de manière prévisible.

Redux Normal

Dans Redux "normal", vous devez configurer manuellement chaque partie de l'application liée à Redux. Cela inclut la création du **store** (lieu où l'état est conservé), des **actions** (qui indiquent ce qui doit changer dans l'état) et des **reducers** (qui précisent comment l'état doit changer en fonction des actions).

Comment ça fonctionne :

- 1. **Store** : Vous créez un store avec `createStore()` en lui passant vos reducers, et vous y ajoutez manuellement des middlewares comme `redux-thunk` pour gérer les actions asynchrones.
- 2. **Actions** : Les actions sont des objets décrivant ce qui doit changer dans l'état. Chaque action a un **type** et des **données** associées.
- 3. **Reducers** : Un reducer est une fonction qui prend l'état actuel et une action, et retourne un nouvel état. Cela implique souvent d'écrire une logique avec des `switch/case` pour chaque action.
- 4. **Asynchronisme** : Pour gérer les actions asynchrones, vous devez intégrer des middlewares comme `redux-thunk` ou `redux-saga`.

Redux Toolkit

Redux Toolkit (RTK) est une version améliorée de Redux qui simplifie et accélère le processus de configuration. Il fournit des abstractions qui réduisent le code répétitif et gèrent automatiquement certaines configurations pour vous.

Comment ça fonctionne :

- 1. **Store** : Vous créez le store avec `configureStore()`, qui ajoute automatiquement des middlewares comme `redux-thunk` et active Redux DevTools.
- 2. **Slices** : Avec `createSlice()`, vous pouvez définir vos reducers et actions dans un seul endroit, ce qui simplifie beaucoup de code.
- 3. **Asynchronisme** : Pour gérer des actions asynchrones, Redux Toolkit offre `createAsyncThunk`, une fonction qui facilite la gestion des requêtes API et autres tâches asynchrones.
- 4. **Immer** : Redux Toolkit utilise la bibliothèque `Immer` pour gérer l'immutabilité de l'état de manière plus simple. Vous pouvez manipuler l'état "directement" dans vos reducers, sans avoir à utiliser l'opérateur `...` ou `Object.assign()`.

Comparaison entre Redux Normal et Redux Toolkit

Aspect	Redux Normal	Redux Toolkit
--------	--------------	---------------

Aspect	Redux Normal	Redux Toolkit
Création du Store	<code>createStore()</code>	<code>configureStore()</code>
Reducers et Actions	Besoin de définir manuellement avec <code>switch/case</code>	Création automatique avec <code>createSlice()</code>
Asynchronisme	Nécessite <code>redux-thunk</code> ou <code>redux-saga</code>	Simplifié avec <code>createAsyncThunk</code>
Immutabilité	Doit être gérée manuellement avec <code>...</code> ou <code>Object.assign()</code>	Gérée automatiquement avec <code>Immer</code>
Boilerplate Code	Beaucoup de code répétitif	Moins de code, grâce à <code>createSlice</code> et autres outils
DevTools	Configuration manuelle	DevTools activé par défaut
Middlewares	Doit être ajouté manuellement	<code>redux-thunk</code> inclus par défaut
Complexité	Plus complexe et flexible	Plus simple et rapide à configurer

Points importants à retenir :

- **Redux Normal** : Offre une grande flexibilité, mais nécessite beaucoup de code répétitif, et il faut tout configurer manuellement. C'est utile pour des cas complexes où un contrôle total est nécessaire.
- **Redux Toolkit** : Réduit le code nécessaire pour gérer l'état, simplifie la gestion des actions asynchrones et améliore la lisibilité du code. Il est idéal pour les développeurs qui cherchent à gagner du temps sans perdre en contrôle sur l'état.

En résumé :

Si vous commencez à utiliser Redux ou si vous avez une application de taille moyenne, **Redux Toolkit** est la meilleure option. Il simplifie grandement la configuration et la gestion de l'état sans sacrifier la performance ou la flexibilité. Si vous avez besoin de plus de contrôle ou si vous travaillez sur une application très complexe, vous pouvez utiliser **Redux normal**, mais sachez que cela demandera plus d'efforts pour gérer toutes les parties du code.