

Primer vistazo a NgRx – Checklist

Acerca del Store:

- NgRx Store proporciona un único contenedor para el estado de nuestra aplicación.
- Nuestro objetivo es interactuar con el estado de una manera inmutable, reemplazando el estado en lugar de modificar la estructura del estado y sus valores.
- Para usar el Store, empiece instalando el paquete `@ngrx/store`.
- Considere la posibilidad de organizar el estado de su aplicación por features, a continuación, nombre cada slice del Store con el nombre de un feature.
- Inicialice el Store, tanto en el módulo raíz como en cada feature module que se utilice en el Store:

```
StoreModule.forRoot(reducer)
StoreModule.forFeature('feature', featureReducer)
```

Acerca de los Actions:

- Un Action representa un evento que cambia el estado de la aplicación, como que el usuario marque un checkbox o seleccione un elemento o que el componente cargue datos vía HTTP.
- Defina un Action para cada evento que desee seguir en la aplicación. No rastree eventos locales a un componente a menos que esos eventos cambien datos que deban ser rastreados después de que el componente sea destruido.

Acerca de los Reducer:

- Un Reducer responde a un Action emitido.
- Sustituye un slice de estado por un nuevo estado basado en el Action.
- Construya una función Reducer para procesar los Actions. A menudo tendrá una función Reducer por feature.
- Implemente la función Reducer usando `createReducer`:

```
export const productReducer = createReducer(
  initialState,
  on(ProductActions.toggleProductCode, state => {
    return {
      ...state,
      showProductCode: !state.showProductCode
    };
  }));
```

- No olvide extender el estado según sea necesario, por ejemplo, usando el operador spread de JavaScript. Esto copia el estado existente a un nuevo objeto y permite actualizar esa copia antes de sobrescribir el estado del Store con esa nueva copia.

Acerca de los Actions:

Una vez que tenemos estos Actions de estado y Reducers definidos, estamos listos para emitir un Action.

- La emisión de un Action se hace a menudo en respuesta a un evento del usuario, como marcar un checkbox o seleccionar un elemento, o para una operación, como la recuperación de datos.
- Inyecte el Store en el constructor del componente.
- Para emitir un Action, llame al método *dispatch* del Store y pase el Action a emitir:

```
this.store.dispatch({  
  type: '[Product] Toggle Product Code'  
});
```

- El Action se enviará a todos los Reducers y será procesado por las funciones correspondientes.

Acerca de suscribirse a los cambios del Store:

Para que cualquiera de nuestros componentes reaccione a los cambios del Store, deben suscribirse.

- La suscripción al Store se hace a menudo en el hook `ngOnInit`. Esto proporciona los valores actuales del Store cuando se inicializa el componente e inmediatamente comienza a ver las notificaciones de cambio.
- En primer lugar, inyecte el Store en el constructor del componente. A continuación, utilice la función “select” del Store, pasando el nombre del *slice* que desea escuchar.
- Por último, suscríbase para obtener los valores actuales del Store y escuche los cambios. El componente recibe notificaciones cada vez que cambia el *slice* de estado seleccionado.