

# Listen to Router State with Router Store



**Duncan Hunter**

Lead Frontend Developer

@dunchunter | [duncanhunter.com.au](http://duncanhunter.com.au)



# Module Introduction



**Why use router store?**

**Add and use @ngrx/router-store**

**Initialize loading data**

**Navigating in effects**



Router Store adds bindings to connect the Angular Router with NgRx Store.



# Why Router Store

**Router actions**

**Router selectors**

**Reduce router logic  
in components**

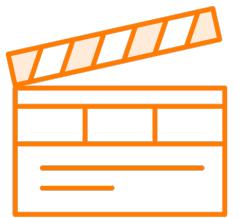


# Navigation Actions



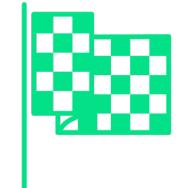
**ROUTER\_REQUEST**

At the start of each navigation



**ROUTER\_NAVIGATION**

During navigation, before any guards or resolvers run



**ROUTER\_NAVIGATED**

After successful navigation



**ROUTER\_CANCEL**

When navigation is cancelled



**ROUTER\_ERROR**

When there is an error during navigation



# Route Selectors

```
import { getRouterSelectors } from '@ngrx/router-store';

export const {
  selectCurrentRoute, // select the current route
  selectFragment, // select the current route fragment
  selectQueryParams, // select the current route query params
  selectQueryParam, // factory function to select a query param
  selectRouteParams, // select the current route params
  selectRouteParam, // factory function to select a route param
  selectRouteData, // select the current route data
  selectRouteDataParam, // factory function to select a route data param
  selectUrl, // select the current url
  selectTitle, // select the title if available
} = getRouterSelectors();
```



# Registering Router Store

```
import { routerReducer, StoreRouterConnectingModule } from '@ngrx/router-store';

@NgModule({
  imports: [
    ],
})
export class AppModule {}
```



# Registering Router Store

```
import { routerReducer, StoreRouterConnectingModule } from '@ngrx/router-store';

@NgModule({
  imports: [
    StoreModule.forRoot({router: routerReducer}),
    StoreRouterConnectingModule.forRoot()
  ],
})
export class AppModule {}
```



# Demo



**Add and use ngrx/router-store**



# Demo



**ngrxOnInitEffects lifecycle hooks**



# Demo



## Navigating in effects

