



Scaffold Root State....

Name	ng generate @nrwl/angular:ngrx
Module *	
Directory	
Barrels	
Facade	
Minimal	
Only Add Files	
Only Empty Root	
Root	
Skip Format	
Skip Import	
Skip Package Json	
Syntax	
Use Data Persistence	
	name Name of the NgRx feature state, such as "products" or "users"). Recommended to use the plural form of the name. <input type="text" value="dummy"/>
	module * The path to NgModule where the feature state will be registered. The host directory will create/use the new state directory. <input type="text" value="apps/demo/src/app/app.module.ts"/>
	directory The name of the folder used to contain/group the generated NgRx files. <input type="text" value="+state"/>
	barrels <input type="checkbox"/> Use barrels to re-export actions, state, and selectors.
	facade <input type="checkbox"/> Create a Facade class for the the Feature.
	minimal <input checked="" type="checkbox"/> Only register the root state management setup or feature state.
	onlyAddFiles <input type="checkbox"/> **Deprecated** , use <code>`skipImport`</code> . Only add new NgRx files, without changing the module file (e.g., <code>--onlyAddFiles</code>).
	onlyEmptyRoot <input type="checkbox"/> **Deprecated** , use <code>`minimal`</code> . Do not generate any files. Only generate <code>StoreModule.forRoot</code> and <code>EffectsModule.forRoot</code> (e.g., <code>--onlyEmptyRoot</code>).
	root <input checked="" type="checkbox"/> Setup root or feature state management with NgRx.
	skipFormat <input type="checkbox"/> Skip formatting of generated files.
	skipImport <input type="checkbox"/> Generate NgRx feature files without registering the feature in the NgModule.
	skipPackageJson <input type="checkbox"/> Do not update the package.json with NgRx dependencies.
	syntax Specifies whether to use class-based or creator functions for actions, reducers, and effects. <input type="text" value="creators"/>
	useDataPersistence <input type="checkbox"/> Generate NgRx Effects with the DataPersistence helper service. Set to false to use plain effects data persistence operators.

Scaffold Feature State...

Name	ng generate @nrwl/angular:ngrx
Module *	
Directory	name Name of the NgRx feature state, such as "products" or "users"). Recommended to use the plural form of the name.
Barrels	<input type="text" value="dummy"/>
Facade	
Minimal	module * The path to NgModule where the feature state will be registered. The host directory will create/use the new state directory.
Only Add Files	<input type="text" value="apps/demo/src/app/dummy/dummy.module.ts"/>
Only Empty Root	
Root	directory The name of the folder used to contain/group the generated NgRx files.
Skip Format	<input type="text" value="+state"/>
Skip Import	
Skip Package.Json	
Syntax	
Use Data Persistence	

barrels
☐ Use barrels to re-export actions, state, and selectors.

facade
☒ Create a Facade class for the the Feature.

minimal
☒ Only register the root state management setup or feature state.

onlyAddFiles
☐ ****Deprecated****, use `'skipImport'`. Only add new NgRx files, without changing the module file (e.g., `--onlyAddFiles`).

onlyEmptyRoot
☐ ****Deprecated****, use `'minimal'`. Do not generate any files. Only generate `StoreModule.forRoot` and `EffectsModule.forRoot` (e.g., `--onlyEmptyRoot`).

root
☐ Setup root or feature state management with NgRx.

skipFormat
☐ Skip formatting of generated files.

skipImport
☐ Generate NgRx feature files without registering the feature in the NgModule.

skipPackageJson
☐ Do not update the package.json with NgRx dependencies.

syntax
Specifies whether to use class-based or creator functions for actions, reducers, and effects.

useDataPersistence
☐ Generate NgRx Effects with the DataPersistence helper service. Set to false to use plain effects data persistence operators.