Angular/Nx 9 Migrations

Nx 8 -> Nx 9 migration

Resources

[Nx 8+ migrations](#)

[Nx migration docs](#)

[Angular 8 to 9](#)

[Angular 9 upgrade docs](#)

[Bound CSS styles and classes changes](#)

Migration to latest Nx 9

Remove @nrwl/cli for migration since migrations don't keep it up to date

```
$ npm uninstall @nrwl/cli
```

[Migration source](#)

```
$ nx migrate 9.5.1
```

migrations.json should be generated with list of automated migration changes (delete after migration)

```
$ npm install
$ nx migrate --run-migrations=migrations.json
$ rm migations.json
```

Issues with migration

```
Running migration @angular/core:migration-v9-undecorated-classes-with-di

This migration uses the Angular compiler internally and therefore
projects that no longer build successfully after the update cannot run
the migration. Please ensure there are no AOT compilation errors and
rerun the migration. The following project failed: apps/mock-api
/tsconfig.app.json

Error: error TS100: Cannot determine the module for class
SelfEnrollmentPersonalInfoComponent in /Users/markthompson/Documents
/github/empowered/portals/libs/accounts/src/lib/agent-self-enrollment
/self-enrollment-personal-info/self-enrollment-personal-info.component.
ts! Add SelfEnrollmentPersonalInfoComponent to the NgModule to fix it.
Cannot determine the module for class SelfEnrollmentPopupComponent in
/Users/markthompson/Documents/github/empowered/portals/libs/accounts/src
/lib/agent-self-enrollment/self-enrollment-popup/self-enrollment-popup.
component.ts! Add SelfEnrollmentPopupComponent to the NgModule to fix
it.
Cannot determine the module for class SelfEnrollmentSitCodeComponent in
/Users/markthompson/Documents/github/empowered/portals/libs/accounts/src
/lib/agent-self-enrollment/self-enrollment-sitcode/self-enrollment-sit-
code.component.ts! Add SelfEnrollmentSitCodeComponent to the NgModule
to fix it.


Could not migrate all undecorated classes that use dependency
injection. Some project targets could not be analyzed due to
TypeScript program failures.

Migration can be rerun with: "ng update @angular/core --migrate-only
migration-v9-undecorated-classes-with-di"

Successfully finished @angular/core:migration-v9-undecorated-classes-
with-di
```

This is because these are components coming from libs but they're being declared in `apps/client`.

These components should be declared in a module in libs than that module should be imported on `apps/client`

After we should run `ng update @angular/core --migrate-only migration-v9-undecorated-classes-with-di` to complete migration

NgModules.entryComponents

NgModules.entryComponents no longer does anything for Ivy.

This means that serve / build no longer will use them.

Tests will continue to use them since Jest doesn't use Ivy in this version.

Jest will naturally use Ivy when we get to Angular 12.

Automated code changes

1. angular.json (all apps / libs)

Example `admin` lib:

```
"test": {
    "options": {
-        "setupFile": "libs/portals/admin/src/test-setup.ts"
+        "setupFile": "libs/portals/admin/src/test-setup.ts",
+        "passWithNoTests": true
    }
}
```

Allows for tests to pass for app / lib even if there's no tests written

2. angular.json (all apps / libs)

Example `admin` lib:

```
"architect": {
    "lint": {
        "builder": "@angular-devkit/build-angular:tslint",
        "options": {
            "tsConfig": [
                "libs/portals/admin/tsconfig.lib.json",
                "libs/portals/admin/tsconfig.spec.json"
            ],
-            "exclude": ["**/node_modules/**", "!libs/portals/admin
/**"]
+            "exclude": ["**/node_modules/**", "!libs/portals/admin
/**/*"]
        }
    },
}
```

3. angular.json:757:25

```
    "mock-api": {
        "architect": {
            "build": {
                "builder": "@nrwl/node:build",
                "options": {
+                   "buildLibsFromSource": true
                },
            }
        }
    }

    `buildLibsFromSource` build option added to all node apps
```

4. package.json:34:9

```
+       "postinstall": "node ./decorate-angular-cli.js"
```

Now all `ng` cli commands are wrapped with `nx`

5. tslint.json (all apps / libs)

```
+   },
+   "linterOptions": {
+       "exclude": ["!**/*"]
    }
```

TSLint Configurations (including `linterOptions`)

6. nx.json:168:5
   - },
   - "tasksRunnerOptions": {

```
    •   "default": {
```

```
    •       "runner": "@nrwl/workspace/tasks-runners/default",
```

```
    •           "options": {
```

- ```
                  "cacheableOperations": ["build", "lint", "test",
      "e2e"]
  ```

- ```
              }
  ```

- ```
          }
  ```

Nx Configurations (including `tasksRunnerOptions`)
7. jest.config.js (all apps / libs)

```
      snapshotSerializers: [
-         "jest-preset-angular/AngularSnapshotSerializer.js",
-         "jest-preset-angular/HTMLCommentSerializer.js"
-     ]
+         "jest-preset-angular/build
/AngularNoNgAttributesSnapshotSerializer.js",
+         "jest-preset-angular/build/AngularSnapshotSerializer.js",
+         "jest-preset-angular/build/HTMLCommentSerializer.js",
+     ],
```

8. Static flag for @ViewChild / @ContentChild refactor

```
-     @ViewChild(NotificationPopupComponent, { static: false })
  notificationPopUp: NotificationPopupComponent;
+     @ViewChild(NotificationPopupComponent) notificationPopUp:
  NotificationPopupComponent;
```

If false, migration will just remove flag since it's the default

MatDialogRef require generic arguments

1. libs/shared/src/lib/ui-components/modal/empowered-modal.service.ts:59:9

```
Type 'MatDialogRef<unknown, any>' is not assignable to type
'MatDialogRef<T, R>'.
Type 'unknown' is not assignable to type 'T'.
'T' could be instantiated with an arbitrary type which could be
unrelated to 'unknown'.ts(2322)

-    const dialogRef = this.matDialog.open(componentOrTemplateRef,
config);
+    const dialogRef: MatDialogRef<T, R> = this.matDialog.open
(componentOrTemplateRef, config);
```

2. libs/commissions/src/lib/commission-splits/add-update-custmized-split/add-update-custmized-split.component.ts:1005:17

```
Argument of type 'MatDialogRef<unknown, any>' is not assignable to
parameter of type 'MatDialogRef<RemovePlanYearPlansComponent,
any>'.

-    const dialogRef = ...
+    const dialogRef: MatDialogRef<DuplicateSplitFoundComponent,
any> = ...
```

3. libs/benefits/src/lib/maintenance-benefits-offering/products/products.component.ts:1515:9

```
Argument of type 'MatDialogRef<unknown, any>' is not assignable to
parameter of type 'MatDialogRef<RemovePlanYearPlansComponent,
any>'.

-    const removeAllPlansDialogRef = ...
+    const removeAllPlansDialogRef:
MatDialogRef<RemovePlanYearPlansComponent, any> = ...
```

Proper suffix is missing for Directive

To avoid rewriting many files, disabling `directive-class-suffix` as needed

styleguide

1. libs/emp-stepper/src/lib/models/AbstractComponentStep.ts:5:23

```
    directive-class-suffix  The name of the class
    AbstractComponentStep should end with the suffix Directive
    (https://angular.io/styleguide#style-02-03)

        @Directive()
+       /* tslint:disable-next-line: directive-class-suffix */
        export abstract class AbstractComponentStep {
```

2. libs/shared/src/lib/services/shared.service.ts:44:14

```
    directive-class-suffix  The name of the class SharedService should
    end with the suffix Directive (https://angular.io/styleguide#style-
    02-03)

        @Directive()
        @Injectable({
            providedIn: "root",
        })
+       /* tslint:disable-next-line: directive-class-suffix */
        export class SharedService {
```

This one is odd since it's trying to be both a service and a directive

3. libs/shared/src/lib/ui-components/pill-filter/panels/AbstractPanel.ts:9:23

```
    directive-class-suffix  The name of the class AbstractPanel should
    end with the suffix Directive (https://angular.io/styleguide#style-
    02-03)

        @Directive()
+       /* tslint:disable-next-line: directive-class-suffix */
        export abstract class AbstractPanel {
```

NGXS states require @Injectable decorator for Ivy

[NGXS Ivy Migration Docs](#)

The following shows two examples, but this affects several more files

1. libs/language/src/lib/+state/language.state.ts:48:1

```
zone-evergreen.js:659 Unhandled Promise rejection: Can't resolve
all parameters for LanguageState: (?). ; Zone: <root> ; Task:
Promise.then ; Value: Error: Can't resolve all parameters for
LanguageState: (?).

    @State(...)
+   @Injectable()
```

2. libs/shared/src/lib/+state/requests/request-state.ts:10:1

   (Error that comes up instead on Angular 10)

```
vendor.js:225322 'RequestState' class should be decorated with
@Injectable() right after the @State() decorator

    @State(...)
+   @Injectable()
```

Angular Material additional required manual code changes for Angular Material

1. libs/pended-business-resolution/src/lib/pended-business.component.ts:3:31

```
-   import { ComponentType } from "@angular/cdk/portal/typings";
+   import { ComponentType } from "@angular/cdk/overlay";
```

   Import path no longer exports `ComponentType`, based on previous version they both exported `ComponentType` from `export *`
   `from './public-api';`

2. libs/shared/src/lib/ui-components/modal/empowered-modal.service.ts:3:31

```
-   import { ComponentType } from "@angular/cdk/portal/typings";
+   import { ComponentType } from "@angular/cdk/overlay";
```

   Import path no longer exports `ComponentType`, based on previous version they both exported `ComponentType` from `export *`
   `from './public-api';`

Other required manual code changes

1. libs/benefits/src/lib/benefits-offering/carrier-forms/carrier-file-upload/carrier-file-upload.component.ts:52:9

```
Type '{ name: string; modifiedName: string; lastModified: number;
size: number; type: string; slice: (start?: number, end?: number,
contentType?: string) => Blob; }' is not assignable to type 'File
& { modifiedName: string; }'.
Type '{ name: string; modifiedName: string; lastModified: number;
size: number; type: string; slice: (start?: number, end?: number,
contentType?: string) => Blob; }' is missing the following
properties from type 'File': arrayBuffer, stream, text

const file: File & { modifiedName: string } = {
+    ...event,
    name: event.name,
    modifiedName: modifiedName,
    lastModified: event.lastModified,
    size: event.size,
    type: event.type,
    slice: event.slice,
};
```

Solved by passing the rest of the event properties using deconstruction

2. libs/api/src/lib/services/account/models/ConcreteAudiences.ts:1:1

```
Import declaration conflicts with local declaration of
'OrgAudience'.

-    import { OrgAudience } from "./ConcreteAudiences";
```

Could not find TypeScript project for project: client-e2e

3. libs/enrollment/src/lib/enrollment.module.ts:30:9

```
Error: Component TpiAddDependentsComponent is not part of any
NgModule or the module has not been imported into your module.

imports: [
+    ApplicationFlowStepsModule,
],
```

4. apps/client/tsconfig.app.json:10:9

```
WARNING in /Users/markthompson/Documents/github/empowered/portals
/libs/multi-factor-auth/src/index.ts is part of the TypeScript
compilation but it's unused.
Add only entry points to the 'files' or 'include' properties in
your tsconfig.

WARNING in /Users/markthompson/Documents/github/empowered/portals
/libs/review-headset-enrollment/src/lib/review-flow/index.ts is
part of the TypeScript compilation but it's unused.
Add only entry points to the 'files' or 'include' properties in
your tsconfig.

WARNING in /Users/markthompson/Documents/github/empowered/portals
/libs/email-tracking/src/index.ts is part of the TypeScript
compilation but it's unused.
Add only entry points to the 'files' or 'include' properties in
your tsconfig.

...
```

Warning shows for all libs.

```
"include": [
    "**/*.ts",
-    "../../libs/portals/admin/src/index.ts",
-    "../../libs/portals/member/src/index.ts",
-    "../../libs/portals/producer/src/index.ts",
-    "../../libs/portals/group-registration/src/index.ts",
-    "../../libs/accounts/src/index.ts",
-    "../../libs/benefits/src/index.ts",
-    "../../libs/dashboard/src/index.ts",
-    "../../libs/login/src/index.ts",
-    "../../libs/members/src/index.ts",
-    "../../libs/multi-factor-auth/src/index.ts",
-    "../../libs/registration/src/index.ts",
-    "../../libs/member-home/src/index.ts",
-    "../../libs/enrollment/src/index.ts",
-    "../../libs/qle/src/index.ts",
-    "../../libs/resources/src/index.ts",
-    "../../libs/company-structure/src/index.ts",
-    "../../libs/commissions/src/index.ts",
-    "../../libs/account-enrollments/src/index.ts",
-    "../../libs/support/src/index.ts",
-    "../../libs/policy-change-request/src/index.ts",
-    "../../libs/overview/src/index.ts",
-    "../../libs/enrollment-options/src/index.ts",
-    "../../libs/assign-admin/src/index.ts",
-    "../../libs/util/csrf/src/index.ts",
-    "../../libs/member-wizard/src/index.ts",
-    "../../libs/benefits-offering/src/lib/carrier-forms/index.ts",
-    "../../libs/profile/src/index.ts",
-    "../../libs/direct-enrollment/src/index.ts",
-    "../../libs/team/src/index.ts",
-    "../../libs/product-exceptions/src/index.ts",
-    "../../libs/htmlviews/src/index.ts",
-    "../../libs/review-headset-enrollment/src/index.ts",
-    "../../libs/review-headset-enrollment/src/lib/review-flow
/index.ts",
-    "../../libs/proposals/src/index.ts",
-    "../../libs/pended-business-resolution/src/index.ts",
-    "../../libs/universal-quote/src/index.ts",
-    "../../libs/tpi/src/index.ts",
-    "../../libs/email-tracking/src/index.ts",
-    "../../libs/aflac-forms-repository/src/index.ts",
-    "../../libs/message-center/src/index.ts"
    ]
```

Budgets are currently too low

There currently shouldn't be any real changes to the code base to have breeched maximumErrors.

To avoid making additional code changes to build smaller, the limits have been adjusted.

angular.json:88:37

```
ERROR in budgets: Exceeded maximum budget for initial-es2015. Budget 10
MB was not met by 4.65 MB with a total of 14.6 MB.
ERROR in budgets: Exceeded maximum budget for initial-es5. Budget 10 MB
was not met by 5.28 MB with a total of 15.3 MB.

"development": {
    ...
    "budgets": [
        {
            "type": "initial",
-           "maximumWarning": "5mb",
-           "maximumError": "10mb"
+           "maximumWarning": "17mb",
+           "maximumError": "20mb"
        }
    ]
},
"production": {
    ...
    "budgets": [
        {
            "type": "initial",
-           "maximumWarning": "5mb",
-           "maximumError": "10mb"
+           "maximumWarning": "17mb",
+           "maximumError": "20mb"
        }
    ]
},
```

migrations_9.json

Minimum Node version is 10.13