Angular Workspace Additional Setup

Here's some additional things that must be done to tidy up an Angular monorepo workspace.

Environment.ts Files

See this page for how to setup multiple environment config files for dev and prod: Building for Multiple Environments

Karma test.ts

Create a test.ts file in the /src folder of each project in the monorepo workspace.

Give it this content:

```
// This file is required by karma.conf.js and loads recursively all the .spec and framework files

import 'zone.js';
import 'zone.js/testing';
import { getTestBed } from '@angular/core/testing';
import {
  BrowserDynamicTestingModule,
  platformBrowserDynamicTesting
} from '@angular/platform-browser-dynamic/testing';

// First, initialize the Angular testing environment.
getTestBed().initTestEnvironment(
  BrowserDynamicTestingModule,
  platformBrowserDynamicTesting(),
);
```

Open the workspace angular.ts file, and edit the test/options block to add a 'main' key that looks like this:

> NOTE: The full path in angular.ts is: <project-name>/architect/test/options

```
"test": {
  "builder": "@angular-devkit/build-angular:karma",
  "options": {
        "main": "projects/lib-oga-webui-runtimesecurity/src/test.ts",
    "tsConfig": "projects/lib-oga-webui-runtimesecurity/tsconfig.spec.json",
    "karmaConfig": "projects/lib-oga-webui-runtimesecurity/karma.conf.js",
    "polyfills": [
      "zone.js",
      "zone.js/testing"
    ]
  }
}
```

The main key needs to point to the test.ts file that you created, above.

Repeat the above for each project in the workspace.

Test.ts Usage

Test.ts Include

For tests to be discovered, you need to make sure that each tsconfig.spec.ts file (one in each project) includes a files reference to the test.ts file.

Here's an example of what a tsconfig.spec.json file looks like with the files block added, and having a reference to the test.ts file:

```
/* To learn more about Typescript configuration file: https://www.typescriptlang.org/docs/handbook/tsconfig-json.html. */
/* To learn more about Angular compiler options: https://angular.dev/reference/configs/angular-compiler-options. */
{
  "extends": "../../tsconfig.json",
  "compilerOptions": {
    "outDir": "../../out-tsc/spec",
```

```
    "types": [
      "jasmine"
    ]
  },
  "files": [
    "src/test.ts"
  ],
  "include": [
    "**/*.spec.ts",
    "**/*.d.ts"
  ]
}
```

Test.ts Exclude

Since libraries get consumed by other developers, we need to make sure that the test.ts file doesn't get included in the packaged library.

To prevent this, we need to add an exclude to the tsconfig.lib.ts of each library project.

Open each tsconfig.lib.ts file and add an exclude for the test.ts file, like the following:

```
/* To learn more about this file see: https://angular.io/config/tsconfig. */
{
  "extends": "../../tsconfig.json",
  "compilerOptions": {
    "outDir": "../../out-tsc/lib",
    "declaration": true,
    "declarationMap": true,
    "inlineSources": true,
    "types": []
  },
  "exclude": [
    "src/test.ts",
    "**/*.spec.ts"
  ]
}
```

NOTE: The above exclude block points to the src/test.ts in that project.

Karma.Conf.ts File

Create a karma.conf.ts file in the base of each project of the monorepo workspace.
This will be just inside the project folder, adjacent to the src folder.

Open the file, and add the content, below.

> NOTE: each karma.conf.ts has a unique path in the coverageReporter block.

Here's what the content should look like:

```
// Karma configuration file, see link for more information
// https://karma-runner.github.io/1.0/config/configuration-file.html

module.exports = function (config) {
  config.set({
    basePath: '',
    frameworks: ['jasmine', '@angular-devkit/build-angular'],
    plugins: [
      require('karma-jasmine'),
      require('karma-chrome-launcher'),
      require('karma-jasmine-html-reporter'),
      require('karma-coverage'),
      require('@angular-devkit/build-angular/plugins/karma')
    ],
    client: {
      jasmine: {
        // you can add configuration options for Jasmine here
        // the possible options are listed at https://jasmine.github.io/api/edge/Configuration.html
        // for example, you can disable the random execution with `random: false`
        // or set a specific seed with `seed: 4321`
      },
      clearContext: false // leave Jasmine Spec Runner output visible in browser
    },
    jasmineHtmlReporter: {
      suppressAll: true // removes the duplicated traces
    },
    coverageReporter: {
      dir: require('path').join(__dirname, '../../coverage/middle2-lib'),
      subdir: '.',
      reporters: [
        { type: 'html' },
```

```
      { type: 'text-summary' }
    ]
  },
  reporters: ['progress', 'kjhtml'],
  port: 9876,
  colors: true,
  logLevel: config.LOG_INFO,
  autoWatch: true,
  browsers: ['Chrome'],
  singleRun: false,
  restartOnFileChange: true
 });
};
```

> NOTE: Be sure to edit the path in coverageReporter/dir, to match the name of the containing project.

Open the angular.ts file, and add a karmaConfig key in each project.

The karmaConfig key will be put in the same test options block that you put the test.ts entry in the previous section.

Here's an example of what that options block will look like:

```
"test": {
 "builder": "@angular-devkit/build-angular:karma",
 "options": {
       "main": "projects/lib-oga-webui-runtimesecurity/src/test.ts",
  "tsConfig": "projects/lib-oga-webui-runtimesecurity/tsconfig.spec.json",
  "karmaConfig": "projects/lib-oga-webui-runtimesecurity/karma.conf.js",
  "polyfills": [
    "zone.js",
    "zone.js/testing"
  ]
 }
}
```

The karmaConfig key needs to point to the karma.conf.js file that you created, above.

Repeat the above for each project in the workspace.

Revision #2
Created 16 March 2025 04:59:43 by glwhite
Updated 16 March 2025 05:31:12 by glwhite