App shell

Application shell is a way to render a portion of your application using a route at build time. It can improve the user experience by quickly launching a static rendered page (a skeleton common to all pages) while the browser downloads the full client version and switches to it automatically after the code loads.

This gives users a meaningful first paint of your application that appears quickly because the browser can render the HTML and CSS without the need to initialize any JavaScript.

Learn more in The App Shell Model.

Step 1: Prepare the application

Do this with the following CLI command: ng new my-app –routing

For an existing application, you have to manually add the `RouterModule` and defining a `<router-outlet>` within your application.

Step 2: Create the app shell

Use the CLI to automatically create the application shell.

ng generate app-shell

For more information about this command see App shell command.

After running this command you can see that the `angular.json` configuration file has been updated to add two new targets, with a few other changes.

“server”: { “builder”: “@angular-devkit/build-angular:server”, “defaultConfiguration”: “production”, “options”: { “outputPath”: “dist/my-app/server”, “main”: “src/main.server.ts”, “tsConfig”: “tsconfig.server.json” }, “configurations”: { “development”: { “outputHashing”: “none”, }, “production”: { “outputHashing”: “media”, “fileReplacements”: [{ “replace”: “src/environments/environment.ts”, “with”: “src/environments/environment.prod.ts” }], “sourceMap”: false, “optimization”: true } } }, “app-shell”: { “builder”: “@angular-devkit/build-angular:app-shell”, “defaultConfiguration”: “production”, “options”: { “route”: “shell” }, “configurations”: { “development”: { “browserTarget”: “my-app:build:development”, “serverTarget”: “my-app:server:development”, }, “production”: { “browserTarget”: “my-app:build:production”, “serverTarget”: “my-app:server:production” } } }

Step 3: Verify the app is built with the shell content

Use the CLI to build the `app-shell` target.

ng run my-app:app-shell:development

Or to use the production configuration.

```
ng run my-app:app-shell:production
```

To verify the build output, open dist/my-app/browser/index.html. Look for default text `app-shell works!` to show that the application shell route was rendered as part of the output.