Deployment

This page describes techniques for deploying your Angular application to a remote server.

Simplest deployment possible

For the simplest deployment, build for development and copy the output directory to a web server.

1. Start with the development build

ng build

1. Copy *everything* within the output folder (dist/ by default) to a folder on the server.
2. If you copy the files into a server *sub-folder*, append the build flag, --base-href and set the <base href> appropriately.

For example, if the index.html is on the server at /my/app/index.html, set the *base href* to <base href="/my/app/"> like this.

ng build --base-href=/my/app/

You'll see that the <base href> is set properly in the generated dist/index.html.  
  
If you copy to the server's root directory, omit this step and leave the <base href> alone.

1. Configure the server to redirect requests for missing files to index.html. Learn more about server-side redirects [below](https://angular.io/guide/deployment#fallback).

This is *not* a production deployment. It's not optimized and it won't be fast for users. It might be good enough for sharing your progress and ideas internally with managers, teammates, and other stakeholders.

Optimize for production

Although deploying directly from the development environment works, you can generate an optimized build with additional CLI command line flags, starting with --prod.

Build with *--prod*

ng build --prod

The --prod *meta-flag* engages the following optimization features.

* [Ahead-of-Time (AOT) Compilation](https://angular.io/guide/aot-compiler): pre-compiles Angular component templates.
* [Production mode](https://angular.io/guide/deployment#enable-prod-mode): deploys the production environment which enables *production mode*.
* Bundling: concatenates your many application and library files into a few bundles.
* Minification: removes excess whitespace, comments, and optional tokens.
* Uglification: rewrites code to use short, cryptic variable and function names.
* Dead code elimination: removes unreferenced modules and much unused code.

The remaining [copy deployment steps](https://angular.io/guide/deployment#copy-files) are the same as before.

You may further reduce bundle sizes by adding the build-optimizer flag.

ng build --prod --build-optimizer

See the [CLI Documentation](https://github.com/angular/angular-cli/wiki/build) for details about available build options and what they do.

Enable production mode

Angular apps run in development mode by default, as you can see by the following message on the browser console:

Angular is running in the development mode. Call [enableProdMode](https://angular.io/api/core/enableProdMode)() to enable the production mode.

Switching to *production mode* can make it run faster by disabling development specific checks such as the dual change detection cycles.

Building for production (or appending the --environment=prod flag) enables *production mode* Look at the CLI-generated main.ts to see how this works.