

Class 29 - Angular as a Platform & Closer Look at the CLI && Course Roundup

[Class 29 Course Content](#)

Lesson Outline

Today we will learn more about:

1. **The Angular CLI**
 2. **Configuration files**
 3. **Testing files**
 4. **Typescript files**
 5. **CLI Commands**
 6. **Schematics**
 7. **Differential Loading**
 8. **What to do now**
-
-

Lesson Notes

- **Angular CLI:** The *Angular Command Line Interface (CLI)* is a terminal tool where one types in commands provided by Angular to initialize, develop, scaffold, and maintain.
 - **Typescript:** *Typescript* is a super-set or variant of Javascript that provides static typing of variables, functions, and classes that increases readability, early error detection, and team consistency.
 - **JSON:** *Javascript Object Notation (JSON)* is a lightweight format for storing and transporting data that is human-readable and machine-friendly.
 - **Differential Loading:** *Differential Loading* is a mechanism Angular uses behind-the-scenes to provide or take away features depending on if the browser is modern or not so every user gets the bundle they need and can use effectively.
 - **CLI Builders:** *CLI Builders* are functions that implement the logic and behavior for a task that can replace a command. Using the CLI Builders is a way to prepare your code for different environment. Some examples include "linting", "testing", and "deploying".
-
-

Angular as a Platform - Walkthrough Steps

STEP 1: Angular CLI Commands

Terminal:

- Type `ng new --help` to see all commands when creating a new Angular project.

- *NOTE:* You can use the `--help` flag on most Angular commands to see the options for that particular method. EG: `ng serve --help`.
- *NOTE:* Many of the options you can choose as flags can be configured on an existing project inside the `angular.json` file.
- Create a new project called "angular-config" and open it in VSCode.

```
ng new --help
# . . .
ng new angular-config
# . . .
cd angular-config
# . . .
code .
```

STEP 2: Walking Through Various Config Files

.editorconfig:

- This file configures the code styling and formatting of the project.

.gitignore:

- This file tells the git repository inside this folder to ignore anything in this file. All the files and folders described will not be tracked and not be sent to GitHub.

browserslistrc:

- This file tells the Angular CLI what browsers to support.

karma.conf.js:

- This file configures the Unit Tests for the project.

package.json:

- This file manages all packages (and their versions) the project needs to run successfully. It also contains scripts that can be executed in the command line.
- *Dependencies* are third-party packages installed that must be shipped to production for the application to function properly.
- *DevDependencies* are third-party packages installed that will not be shipped to production servers as they are tools used during the development process. You install devDependencies by running `npm i --save-dev MY_PACKAGE`.

tsconfig.json:

- This file configures the typescript compiler to watch for certain behaviors while ignoring others.

STEP 3: Walking Through the Angular.JSON file

angular.json:

- This file is generated every time a new Angular project is created using the CLI.
 - This file is used by the CLI behind-the-scenes to configure commands based on project needs.
 - The *\$schema* is information the Angular team provides as defaults for general project layout and configuration.
 - The *newProjectRoot* is where projects are added if managing multiple applications in one file is beneficial.
 - In the *projects* object, there lies the:
 - *projectType* which signifies the type of application.
 - *schematics* which are used to customize the **ng generate** sub-commands.
 - *root* which tells Angular CLI where the root folder of the application is stored.
 - *sourceRoot* which is where the source files are located.
 - *prefix* which is where to configure the prefix to every component.
 - *architect* which has many different commands relating to building, serving, and testing the application.
-

STEP 4: All About Schematics

- *Schematics* are blueprints Angular commands can use to do a whole bunch of stuff at once, reliably.
 - Examples of Schematics include:
 - **ng generate**... the command to create a component, directive, or service the same way every time.
 - **ng add**... the command to add libraries and functionality to the project.
 - **ng update**... the command to update all packages the project requires.
-
-

Course Wrapup

- Congratulations!!! You made it to the end of the Main Course.
 - Next up we have
 1. Personal Portfolio
 2. Mock Interviews
 3. Employer Project
-
-

Additional Notes

Resources

- [Angular Docs - What is Angular](#)
- [Angular Docs - Angular Workspace Config Structure](#)
- [Angular Docs - Generating Code Using Schematics](#)
- [Angular Docs - Angular CLI Builders](#)
- [How the Angular Compiler Works](#)





```
--packages  
--registry=registry  
--to=to  
--verbose=true|false
```

ng update
Update your Angular
project

