SPA

- browser based applications
- loads all assets and optionally data on initial page load
- refresh/reload of browser not required
- fluid, and dynamic UX
- fast and responsive
- device's native app look & feel
- HTML5, JS, CSS3
- AJAX, REST, WebSockets
- offline with client side data storage (web storage, URL manipulation etc)
- less load on server
- scalable
- HTML5, JS, CSS3 can be put on CDN

SPA

- Traditional
    - one website/page is one application
    - app code keeps track of all the changes made by the user (scope)
- Modern
    - scope can be modified to reduce memory consumption
    - can have independent, multiple apps

- Cons
    - initial page load can be slow
    - high memory consumption
    - JS complexities

SPA

- Example Frameworks
    - Angular
    - Vue
    - React
    - Ember
    - Meteor

Angular

- desktop and mobile apps
- component based architecture
- framework
- speed and performance
  - DI
  - change detection
  - routing
- richer and complex templating
- JS, TypeScript, or Dart
- Angular2 built with Type Script
    - classes
    - templates
    - types
    - annotations
    - use build tools (webpack, gulp etc)

Angular

- The need

    - interactivity

    - data collection/modification

        - UI content updates as user enters/modifies

    - reusability

    - modularity

    - DI

    - complex SPAs

        - router module

Angular

- Setup

  - git

    https://git-scm.com

  - node & npm

    https://nodejs.org/en

  - text editor

Angular

- TypeScript concepts
    - Angular source code is written in TS
    - typed superset of JS
    - transpiled to JS
    - classes
    - modules
- strong typing for variables and fn. signatures
- decorators & annotations
    - simple
    - learn & reference from the angular source code

Angular

- Component based
    - independent code units
    - connect components
    - CSS, templates, & scripts

- Modular
    - load only the required components
    - efficient and fast

- Data binding
    - for connecting data/models with views/templates

Angular

- Templating
- CLI
  - Command Line Interface
  - opinionated application scaffolding
  - create/build/generate/serve.. components
  - connect components

ng new <app_name> //create/scaffold a new application

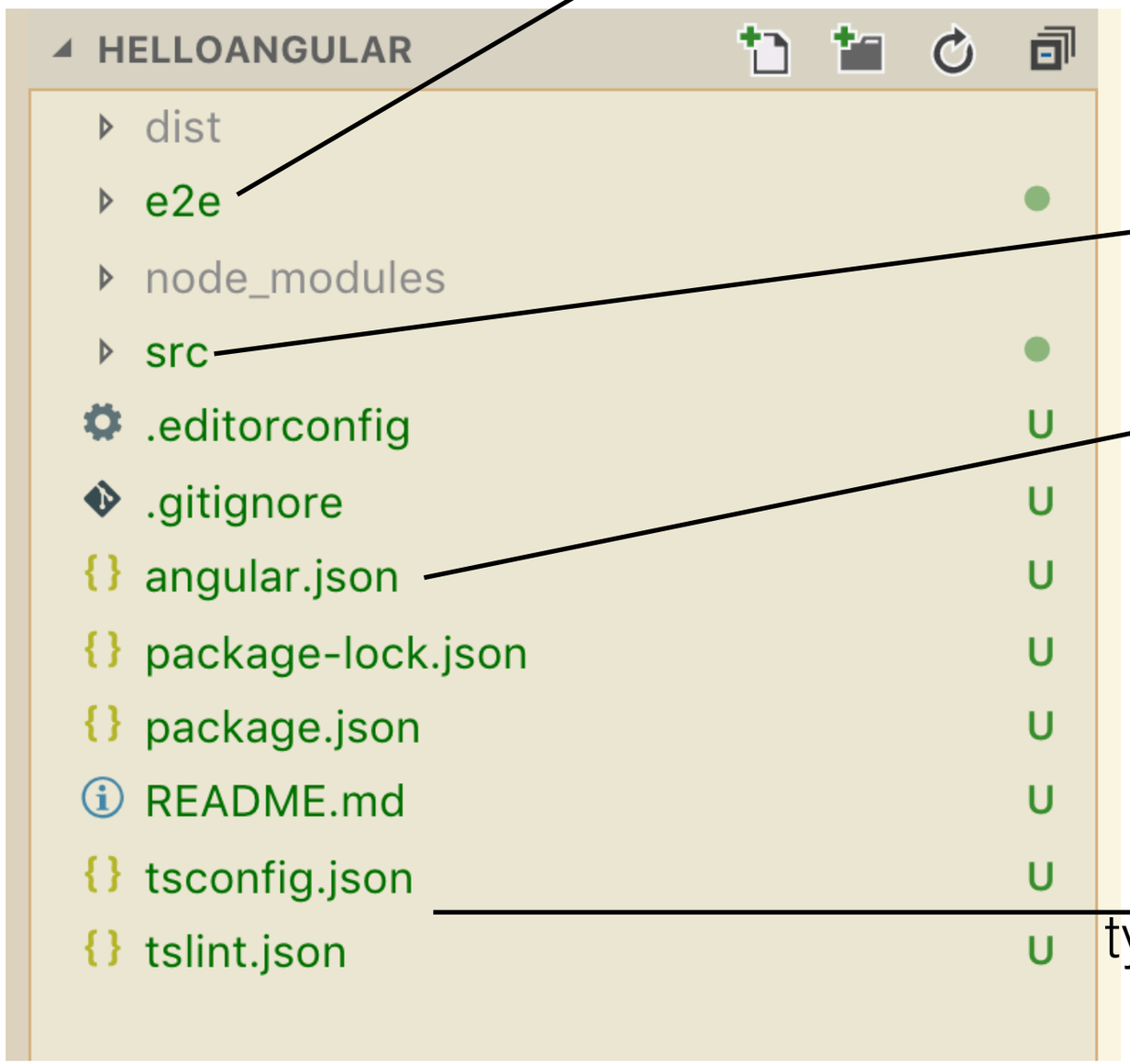
ng serve <-o> //run app in dev mode (uses webpack)

ng build //for deployment (publishing the app to a server)

npm install -g @anular/cli

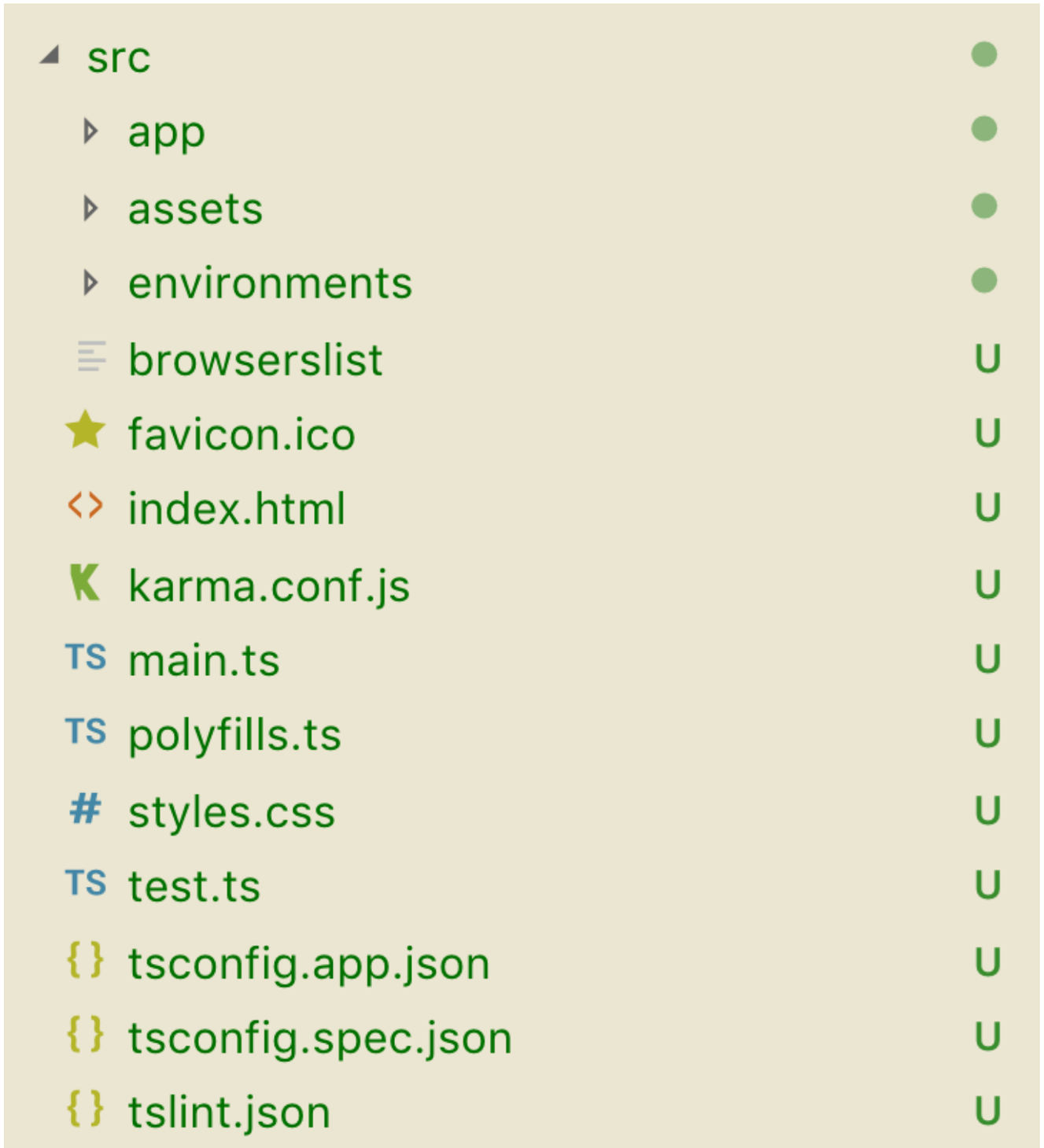
Angular

- Project Str.



Angular

- Project Str.



Angular

- main.ts

TS main.ts

```ts
import { enableProdMode } from '@angular/core';
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';

import { AppModule } from './app/app.module';
import { environment } from './environments/environment';

if (environment.production) {
  enableProdMode();
}

platformBrowserDynamic().bootstrapModule(AppModule)
  .catch(err => console.error(err));
```

//1,2: Angular libraries from node_modules

//4,5: local import

//4: default CLI generated module

//11: starts the app

Angular

- app.module.ts

TS app.module.ts ✖

```ts
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

specify the parameters for the app

@: decorator (ES6 pattern)

//identifies which components/libraries go in this module

Angular

- app.component.ts

TS app.component.ts ✖

```ts
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'helloangular';
}

```

<> index.html ✖

```html
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Helloangular</title>
  <base href="/">

  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
</head>
<body>
  <app-root></app-root>
</body>
</html>
```

//4: custom HTML tag that will have the component

//9: variable title that can be accessed in HTML

Angular

- app.component.html

Expression

app.component.html

```
<!--The content below is only a placeholder and can be replaced.-->
<div style="text-align:center">
  <h1>
    Welcome to {{ title }}!
  </h1>
  <img width="300" alt="Angular Logo" src="
</div>
<h2>Here are some links to help you start: </h2>
<ul>
  <li>
    <h2><a target="_blank" rel="noopener" href="https://angular.io/tutorial">Tour of Heroes</a
  </li>
  <li>
    <h2><a target="_blank" rel="noopener" href="https://angular.io/cli">CLI Documentation</a><
  </li>
  <li>
    <h2><a target="_blank" rel="noopener" href="https://blog.angular.io/">Angular blog</a></h2
  </li>
</ul>

<router-outlet></router-outlet>
```

Angular

- Adding other frameworks to Angular
    - package.json
    - node_modules

npm install jquery --save-dev //from ur Angular proj. dir

npm install bootstrap --save-dev //from ur Angular proj. dir

{} package.json ×

```
    "zone.js": "~0.8.26"
  },
  "devDependencies": {
    "@angular-devkit/build-angular": "~0.13.0",
    "@angular/cli": "~7.3.7",
    "@angular/compiler-cli": "~7.2.0",
    "@angular/language-service": "~7.2.0",
    "@types/jasmine": "~2.8.8",
    "@types/jasminewd2": "~2.0.3",
    "@types/node": "~8.9.4",
    "bootstrap": "^4.3.1",
    "codelyzer": "~4.5.0",
    "jasmine-core": "~2.99.1",
    "jasmine-spec-reporter": "~4.2.1",
    "jquery": "^3.3.1",
    "karma": "~4.0.0",
    "karma-chrome-launcher": "~2.2.0",
    "karma-coverage-istanbul-reporter": "~2.0.1",
    "karma-jasmine": "~1.1.2",
    "karma-jasmine-html-reporter": "^0.2.2",
    "protractor": "~5.4.0",
    "ts-node": "~7.0.0",
    "tslint": "~5.11.0",
    "typescript": "~3.2.2"
  }
}
```

Angular

- Adding other frameworks to Angular

angular.json

```
22          "src/favicon.ico",
23          "src/assets"
24        ],
25        "styles": [
26          "./node_modules/bootstrap/dist/css/bootstrap.css",
27          "src/styles.css"
28        ],
29        "scripts": [
30          "./node_modules/jquery/dist/jquery.js",
31          "./node_modules/bootstrap/dist/js/bootstrap.bundle.js"
32        ],
```

index.html

```
1   <!doctype html>
2   <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <title>Helloangular</title>
6     <base href="/">
7
8     <meta name="viewport" content="width=device-width, initial-scale=1">
9     <link rel="icon" type="image/x-icon" href="favicon.ico">
10  </head>
11  <body>
12    <div class="container">
13      <app-root></app-root>
14    </div>
15  </body>
16  </html>
```