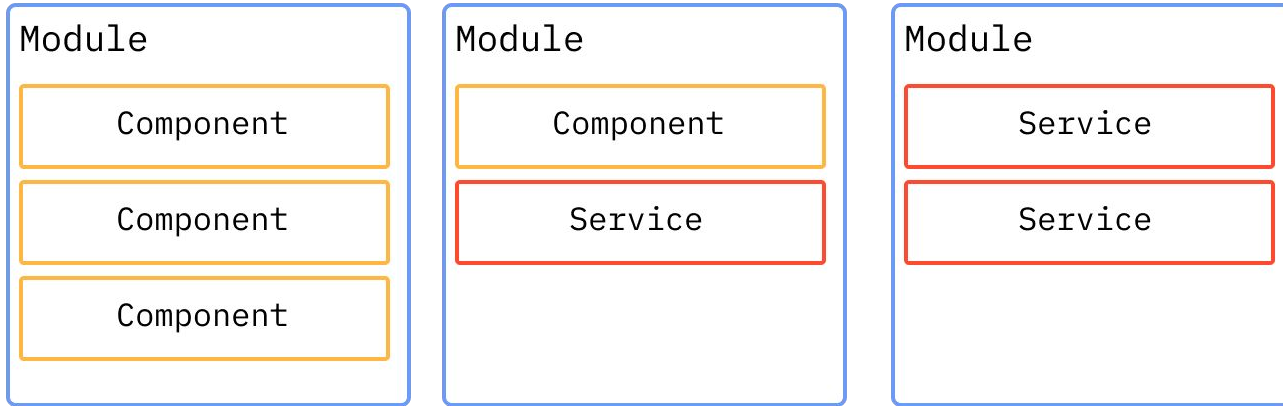
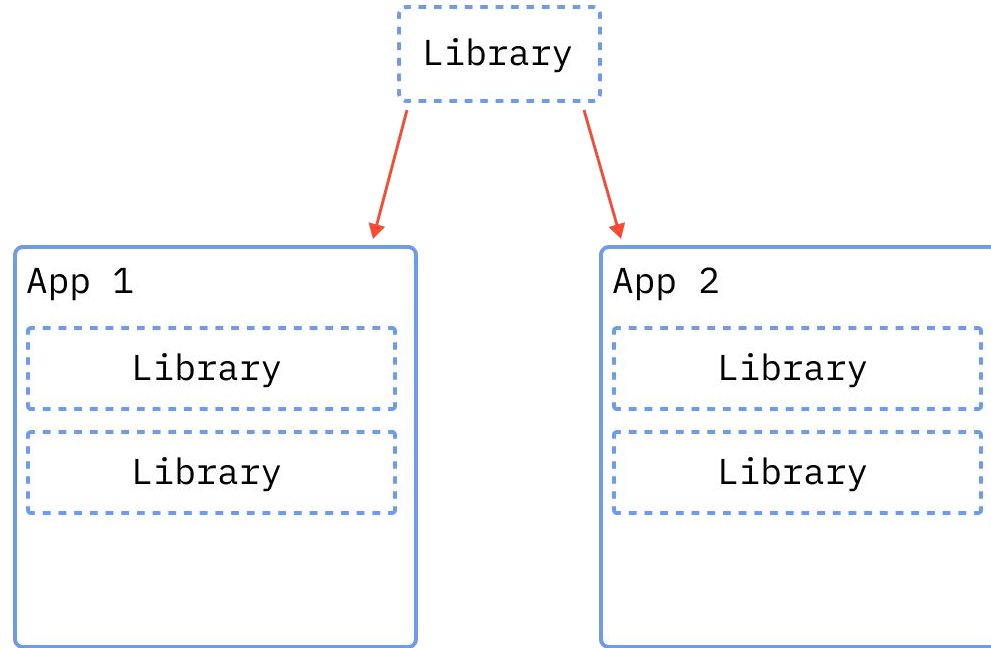


# Angular Libraries

Лектор: Петър Маламов

## Library





# Какво е Angular Library?

Angular проект, който не може да работи самостоятелно. Той трябва да бъде импортирана и използвана в приложение.

Angular библиотеките могат да се използват локално в работната среда или да се публикуват като npm пакети.

Те могат да бъдат разпространявани чрез публичния npm регистър, частен npm Enterprise регистър или друга система за управление на пакети, съвместима с npm.

# Защо да използваме Angular Library?

- **Преизползваемост** - Кодът може да бъде използван в множество проекти.
- **Модулност** - Подобрява структурата на кода чрез разделяне на отделни функционалности.
- **Изолиране на зависимости** - Управление на специфични зависимости за отделни функционалности.

# Как се създава Angular Library?

Angular библиотека се създава чрез Angular CLI. При създаването на библиотеката се генерира папка projects, в която ще се намират всички локално създадени библиотеки.

```
ng generate library my-lib
```

```
▼ projects\my-lib  
  > src  
  {} ng-package.json  
  {} package.json  
  ⓘ README.md  
  {} tsconfig.lib.json  
  {} tsconfig.lib.prod.json  
  {} tsconfig.spec.json
```

# Как се създава Angular Library?

За да стане кодът на библиотеката преизползваем, трябва да се дефинира публичен API, който предоставя достъп до функционалността чрез **single import path**.

Публичният API се поддържа във файла **public-api.ts**, а всичко, експортирано от него, става публично при импортиране на библиотеката в приложение.

TS public-api.ts

```
/*  
 * Public API Surface of my-lib  
 */  
  
export * from './lib/my-lib.service';  
export * from './lib/my-lib.component';
```



# Важно!

Builder-ът за библиотеката е различен от този за приложения, като гарантира изграждането с Ahead-of-time (AOT) компилатора.

При AOT кода се компилира още по време на изграждането, преди той да бъде изпълнен. Това подобрява производителността и намалява размера на крайния пакет, тъй като кодът се компилира предварително, а не по време на изпълнение, както е при JIT компилацията.

# Създаване на Angular Library от Feature.

- Components и pipes трябва да бъдат stateless, като не разчитат на външни променливи.
- Абонаментите на компоненти към observables трябва да бъдат правилно почистени в техния жизнен цикъл.
- Взаимодействията между компонентите трябва да се предоставя чрез inputs за контекст и outputs за събития.
- Необходимо е да се прегледат всички вътрешни зависимости и да се мигрират тези, които са специфични за приложението.
- Ако библиотеката предоставя опционални services, които не се използват от всички приложения, трябва да се осигури начин за tree-shaking.

# Използване на Angular Library

Angular предоставя възможност за локално използване на новосъздадени библиотеки в текущото работно пространство или публикуването им в публичен или частен npm регистър.

Публикуването в npm регистър позволява библиотеките да бъдат споделяни и използвани в други Angular проекти.

# Публикуване в npm регистър

Angular CLI използва инструментът **ng-packagr** за създаване на пакети от компилирания код, които могат да бъдат публикувани в npm.

За да се публикува библиотека в npm регистър, първо трябва да се изгради (build). След това, трябва да се премине в директорията с изградения пакет и оттам да се публикува.

```
ng build my-lib  
cd dist/my-lib  
npm publish
```

# Важно!

При избора на име за библиотека, която ще се публикува в npm, трябва да се избягва префиксът `ng-`, тъй като е запазен за Angular. Препоръчва се използването на префикса `ngx-`, който показва съвместимост с Angular приложения.

# Използване на библиотека локално

За да се използва персонализирана библиотека локално, е необходимо първо да бъде изградена (build). След това тя може да се използва като обикновена библиотека.

```
ng build my-lib
```

```
import { myExport } from 'my-lib';
```

# Важно!

При импортиране на библиотека в Angular приложение, Angular търси съответствие между името на библиотеката и местоположението ѝ на диска.

Инсталираните библиотеки се намират в папката `node_modules`, а за собственооръчно създадени библиотеки се използват пътищата, зададени в `tsconfig`.

TS tsconfig.json

```
"compilerOptions": {  
  "outDir": "./dist/out-tsc",  
  "strict": true,  
  "noImplicitOverride": true,  
  "noPropertyAccessFromIndexSignature": true,  
  "noImplicitReturns": true,  
  "noFallthroughCasesInSwitch": true,  
  "skipLibCheck": true,  
  "paths": {  
    "my-lib": [  
      |   "./dist/my-lib"  
    ]  
  },  
  "esModuleInterop": true,  
  "sourceMap": true,  
  "declaration": false,  
  "experimentalDecorators": true,  
  "moduleResolution": "node",  
  "importHelpers": true,  
  "target": "ES2022",  
  "module": "ES2022",  
  "useDefineForClassFields": false,  
  "lib": [  
    |   "ES2022",  
    |   "dom"  
  ]  
},
```

Определя местоположенията  
за търсене при използване  
на персонализирани  
библиотеки.



# Важно!

Angular CLI използва различни инструменти за изграждане на библиотеки и приложения:

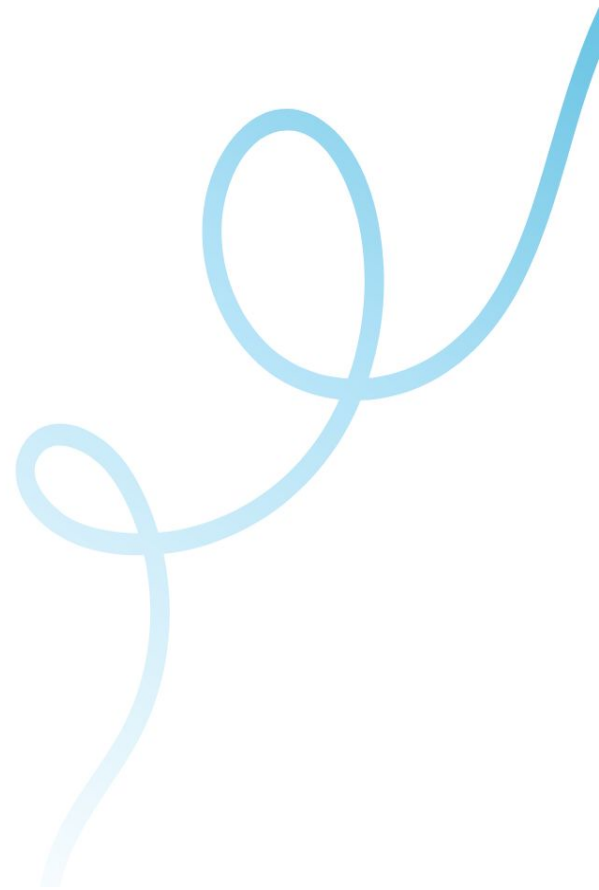
- За приложения се използва **@angular-devkit/build-angular**, което е базирано на **webpack** и е част от всяко ново Angular CLI приложение.
- За библиотеки се използва **ng-packagr**, специализиран инструмент за пакетиране на библиотеки, който се добавя към зависимостите само при създаване на библиотека чрез `ng generate library`.

# Зависимости

Angular библиотеките трябва да посочват всички зависимости от @angular/\* като **peerDependencies**, за да се гарантира, че всички модули получават една и съща версия на Angular.

Ако библиотека добави @angular/core в dependencies вместо в peerDependencies, може да се инсталира различна версия на Angular, което да доведе до проблеми в приложението.

# Упражнение



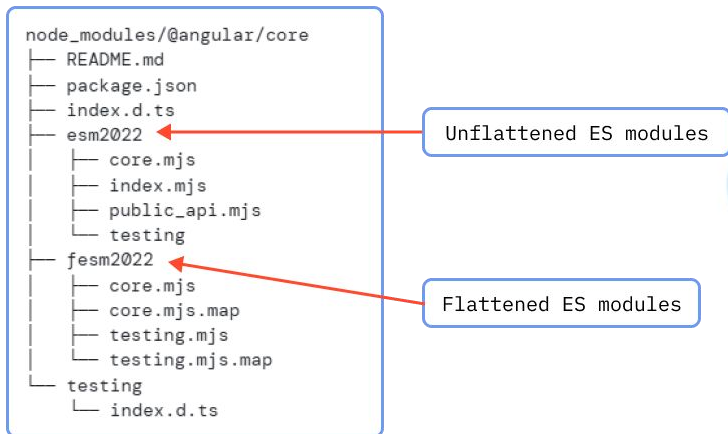
# Angular package format

# Какво е Angular package format (APF)

APF е стандарт за начина, по който Angular библиотеките трябва да бъдат структурирани и разположени, за да бъдат съвместими с Angular приложения и да осигурят оптимална работа с различни версии на Angular.

Това е форматирането на библиотеките, което гарантира тяхната правилна интеграция и работа с Angular приложения, когато те се използват чрез npm пакети.

# Структура на файловете



**Благодаря за вниманието!**