

сряда, 26.07.2017

Modules

1. Откъде започва application-а ?

- Общо взето имаме NgModule декоратор, който можем да сложим на някой клас. И можем да имаме много на брой NgModule класове, които отговарят за различни неща. Примерно един NgModule клас да отговаря за user-ите на нашето приложение, друг NgModule клас ще отговаря за курсовете, трети за коли, четвърти за студенти и тн. Тоест разбиваме си многото компоненти на малки парченца и ги слагаме в отделни модули така че да бъдат логически свързани.
- Модулите са опционални и се ползват по избор. Те са само и единствено, за да ни улеснят работата като девелопъри, но не ни задължават да следваме стриктни правила и патърни.
- Едно единствено нещо има нужда Ангулар от него, за да тръгне. Това е т.нар. root модул. Представлява клас който е деклариран с NgModule и откъдето тръгват всичките неща. Буквално от него се декларираат всичките компоненти, сървиси и др..надолу. Когато отидем в директория src/app, там app.module.ts ни се явява главният root модул. В него има няколко неща - declarations, imports, providers и bootstrap:

– **Declarations** - *Необходими са, когато сме си създали нов компонент и искаме да кажем на Ангулар че трябва да използва този клас.*

– **Imports** - *Съдържа всички останали модули, които искаме да ползваме наготово - например готова функционалност от други колеги, third-party модули или нещо друго от Ангулар екипа.. Примерно модула HttpClientModule просто се импортира най-отгоре във файла, добавя се в imports и след това вече може да се inject-ва наляво-надясно в конструкторите. Това обаче няма да сработи ако не е добавен в imports. Също така ако си създадем собствени модули, те също трябва да се добавят там.*

– **Providers** - *Означава, че искам да вкарам неща, които могат да бъдат inject-вани по конструкторите на всякакви компоненти. Ако се регистрират в providers стават глобални. Това означава че оттук насетне всеки един компонент може да ги използва.*

– **Bootstrap** - *Това е нещо, което само глобалният root модул притежава. Тук казваме, кой компонент искаме да бъде заместен в index.html. И все пак това си е просто клас и нищо повече. Той трябва отнякъде да бъде стартиран, и извикан, и някой трябва да го направи - и тук идва файлът main.ts*

2. Как можем ние сами да си създаваме модули и каква им е идеята? Накратко означава да експортируем някакъв клас, който трябва да се регистрира и му трябва NgModule от '@angular/core'. Другият модул, който е главен вкарва BrowserModule в imports и това е само в root модула. Никъде другаде. Нашите custom модулчета трябва да импортират Common Module.

```
import { NgModule }      from '@angular/core';
import { CommonModule }   from '@angular/common';
import { FormsModule }    from '@angular/forms';
import { SomePipe }       from './some.pipe';
import { SomeComponent }  from './some.component';
```

```

import { SomeComponent } from './some.component';
import { SomeService } from './some.service';
import { SomeDirective } from './some.directive';

@NgModule({
  imports: [ CommonModule, FormsModule ],
  declarations: [ SomeComponent, SomeDirective, SomePipe ],
  exports: [ SomeComponent ],
  providers: [ SomeService ]
})
export class SomeModule { }

```

```

app.module.ts x
1 import { BrowserModule } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3
4 import { AppComponent } from './app.component';
5
6 @NgModule({
7   declarations: [
8     AppComponent
9   ],
10  imports: [
11    BrowserModule
12  ],
13  providers: [],
14  bootstrap: [AppComponent]
15 })
16 export class AppModule {}
17

```

Прилича много на root модула, но с малки разлики:

- Тук имаме *export*, вместо *imports*
- Няма *bootstrap*
- Тук ползваме *CommonModule*, а не *BrowserModule*

Създаваме на собствен модул:

1. Трябва да експортируем някакъв клас

```

3 export class UserModule {}

```

2. Трябва да импортируем NgModule от '@angular/core';

```

app.module.ts app.component.html users.service.ts
1 import { NgModule } from '@angular/core';
2
3 export class UserModule {}

```

3. Вместо BrowserModule, трябва да импортируем CommonModule от '@angular/common'

```
app.module.ts  app.component.html  users.service.ts
1  import { NgModule } from '@angular/core';
2  import { CommonModule } from '@angular/common';
3
4  export class UserModule {}
```

4. След това слагаме декоратора *NgModule* и започваме да описваме, подобно на компонент - Трябва да опишем 4 неща, които споменахме по-горе - *declarations, imports, providers* и *exports*.

```
app.module.ts  app.component.html  users.service.ts
1  import { NgModule } from '@angular/core';
2  import { CommonModule } from '@angular/common';
3
4  @NgModule({
5    imports: [],
6    declarations: [],
7    providers: [],
8    exports: []
9  })
10 export class UserModule {}
```

5. Сега трябва да си вкараме всички компоненти и сървиси, които ни интересуват. Първо си ги импортираме разбира се.

```
app.module.ts  app.component.html  users.service.ts
1  import { NgModule } from '@angular/core';
2  import { CommonModule } from '@angular/common';
3
4  import { UserComponent } from './users.component'
5  import { UserService } from './users.service'
6
7  @NgModule({
8    imports: [CommonModule],
9    declarations: [UserComponent],
10    providers: [UserService],
11    exports: []
12  })
13 export class UserModule {}
```

6. Отиваме в *app modules*, за да си добавим нашият. Първо отново си го импортираме.

```
app.module.ts x  app.component.html  users.service.ts
1  import { BrowserModule } from '@angular/platform-browser';
2  import { NgModule } from '@angular/core';
3
4  import { AppComponent } from './app.component';
5
6  import { UsersModule } from './users/users.module';
7
8  @NgModule({
9    declarations: [
10      AppComponent
11      // UsersComponent
12    ],
13    imports: [
14      BrowserModule,
15      UsersModule
16    ],
17    providers: [],
18    exports: []
19  })
20 export class AppModule {}
```

```

11 // UserComponent
12 ],
13 imports: [
14   BrowserModule
15 ],
16 providers: [],
17 bootstrap: [AppComponent]
18 })
19 export class AppModule {}
20

```

7. Добавяме го в импортите

```

8 @NgModule({
9   declarations: [
10     AppComponent
11     // UserComponent
12   ],
13   imports: [
14     BrowserModule,
15     UsersModule
16   ],
17   providers: [],
18   bootstrap: [AppComponent]
19 })
20 export class AppModule {}
21

```

8. Остава само още едно нещо, за да заработи. По принцип всичко в `declarations` е `private`, за да не се сбият отделните модули ако имат и двамата примерно `HomeComponent`. За това изрично в `exports` трябва да кажем кое искаме да ни е публично и да могат всички да го виждат.

```

app.module.ts  app.component.html  users.service.ts
1  import { NgModule } from '@angular/core';
2  import { CommonModule } from '@angular/common';
3
4  import { UserComponent } from './users.component'
5  import { UserService } from './users.service'
6
7  @NgModule({
8    imports: [CommonModule],
9    declarations: [UserComponent],
10   providers: [UserService],
11   exports: [UserComponent]
12 })
13 export class UsersModule {}

```