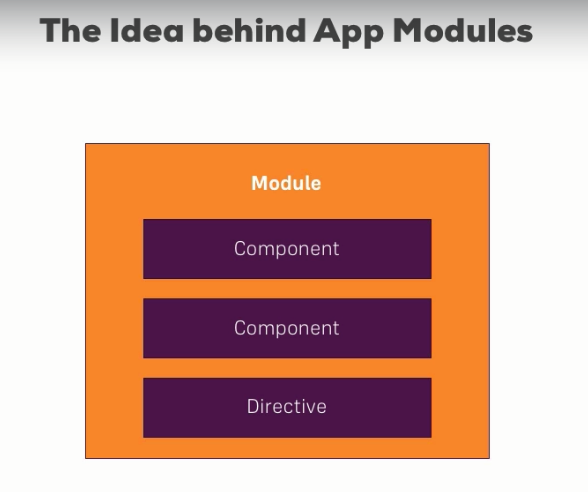
**Section 21 Using Angular Modules and Optimizing Apps**

**Section 21: Lecture 267//Module Introduction**

1. This section is about how to use modules and in general how you can optimize your angular application.
2. Modules play an important role in optimizing an application.
3. We will learn how we can increase the performance of the app and how we can decrease the file size and restructure your code in an easier way.

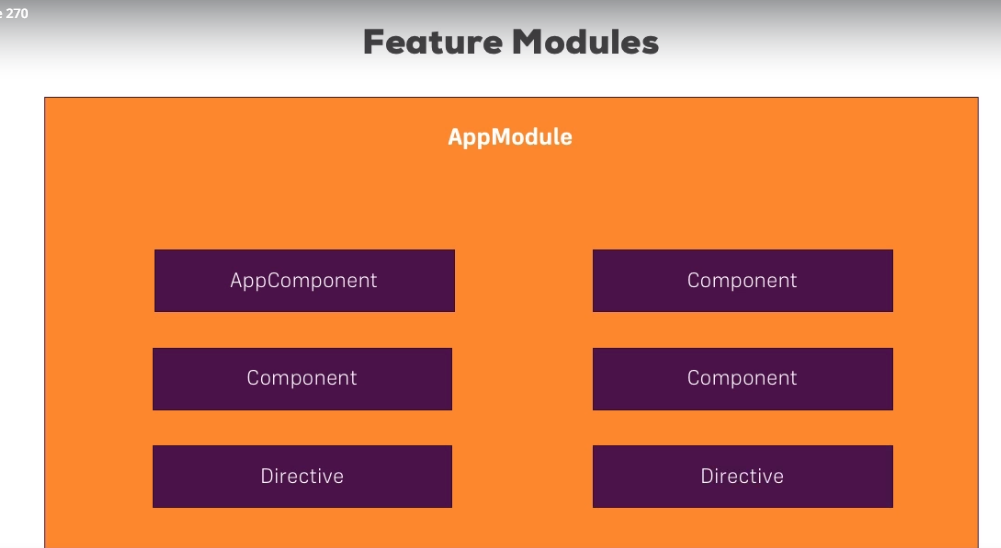
**Section 21: Lecture 268//The Idea behind Modules**

1. 

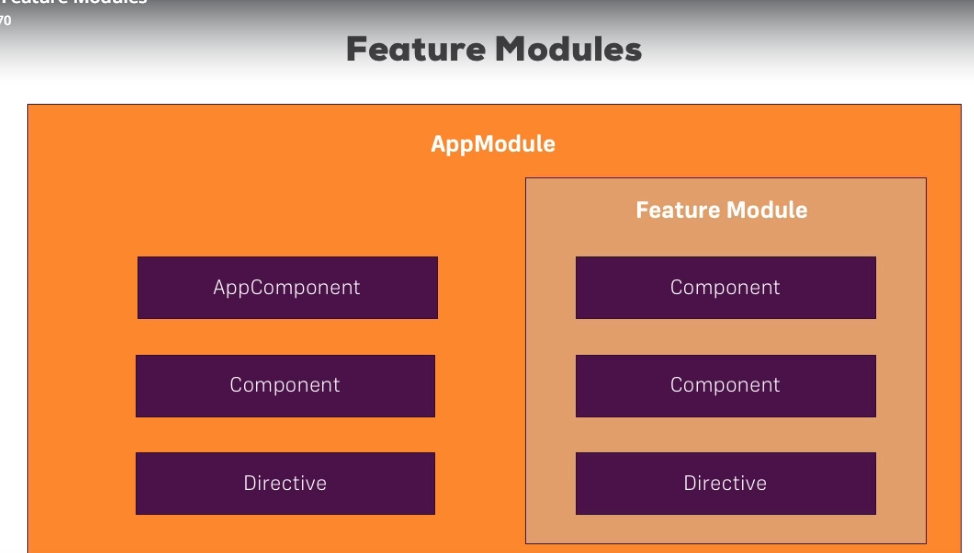
**Section 21: Lecture 269//Understanding the App Module**

**Section 21: Lecture 270//Understanding Feature Modules**

1. A typical custom module added by us in our app is called the feature module. Our application as of now is only having app module.
2. AppModule:



1. Feature Module:



**Section 21: Lecture 271//Creating a Recipe Feature Module**

1. In the recipes folder in our metadata we can see a lot of files or components, now each of this component is defined in the AppModule file.
2. All these components belong to our recipe feature. Now, we will go to recipe folder and will create our own file named recipe.module.ts.
3. Now, we will move all the recipe related components in this module and will leave the services in the app.module.ts as these are also used by the other parts of the Application.
4. CommonModule gives us access to all the common modules.
5. recipes.module.ts:
6. import { NgModule } from "@angular/core";
7. import { RecipesComponent } from "./recipes.component";
8. import { RecipeStartComponent } from "./recipe-start/recipe-start.component";
9. import { RecipeListComponent } from "./recipe-list/recipe-list.component";
10. import { RecipeEditComponent } from "./recipe-edit/recipe-edit.component";
11. import { RecipeDetailComponent } from "./recipe-detail/recipe-detail.component";
12. import { RecipeItemComponent } from "./recipe-list/recipe-item/recipe-item.component";
13. import { ReactiveFormsModule } from "@angular/forms";
14. import { CommonModule } from "@angular/common";
15. @NgModule({
16. declarations:[
17. RecipesComponent,
18. RecipeStartComponent,
19. RecipeListComponent,
20. RecipeEditComponent,
21. RecipeDetailComponent,
22. RecipeItemComponent
23. ],
24. imports: [
25. CommonModule,
26. ReactiveFormsModule
27. ]
28. })
29. export class RecipesModule{
31. }

6. app.module.ts:

import { BrowserModule } from '@angular/platform-browser';

import { NgModule } from '@angular/core';

import { FormsModule } from '@angular/forms';

import { HttpModule } from '@angular/http';

import { AppComponent } from './app.component';

import { HeaderComponent } from './header/header.component';

import { ShoppingListComponent } from './shopping-list/shopping-list.component';

import { ShoppingEditComponent } from './shopping-list/shopping-edit/shopping-edit.component';

import { DropdownDirective } from './shared/dropdown.directive';

import { ShoppingListService } from './shopping-list/shopping-list.service';

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

import { RecipeService } from './recipes/recipe.service';

import { DataStorageService } from './shared/data-storage.service';

import { SignupComponent } from './auth/signup/signup.component';

import { SigninComponent } from './auth/signin/signin.component';

import { AuthService } from './auth/auth.service';

import { AuthGuard } from './auth/auth-guard.service';

import { RecipesModule } from './recipes/recipes.module';

@NgModule({

declarations: [

AppComponent,

HeaderComponent,

ShoppingListComponent,

ShoppingEditComponent,

DropdownDirective,

SignupComponent,

SigninComponent

],

imports: [

BrowserModule,

FormsModule,

HttpModule,

AppRoutingModule,

RecipesModule

],

providers: [ShoppingListService, RecipeService, DataStorageService, AuthService, AuthGuard],

bootstrap: [AppComponent]

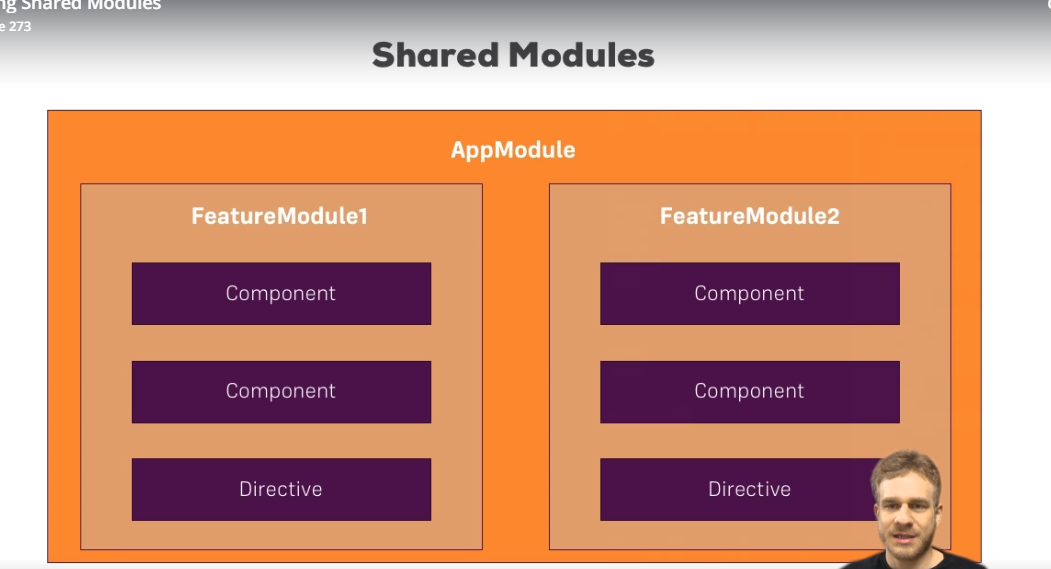
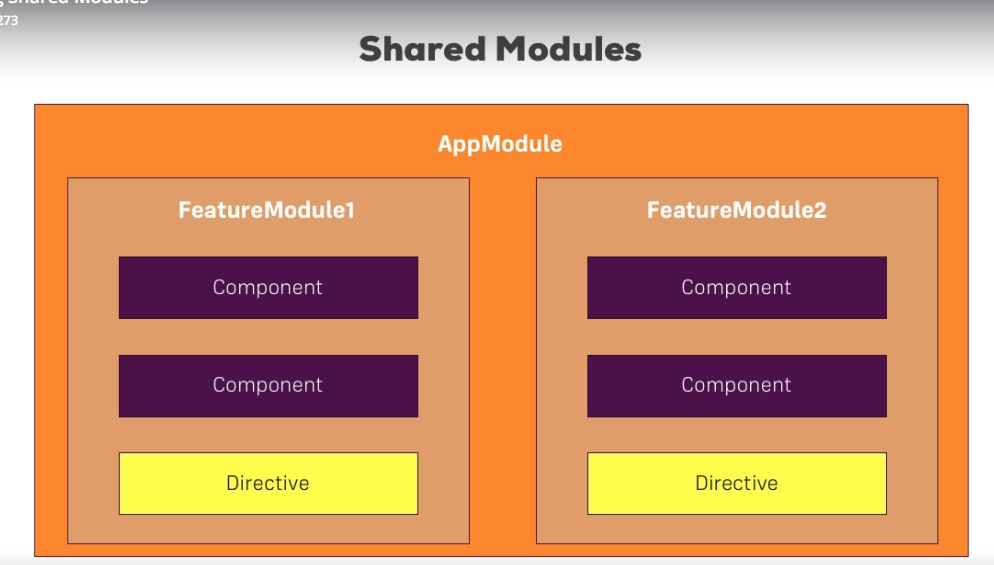
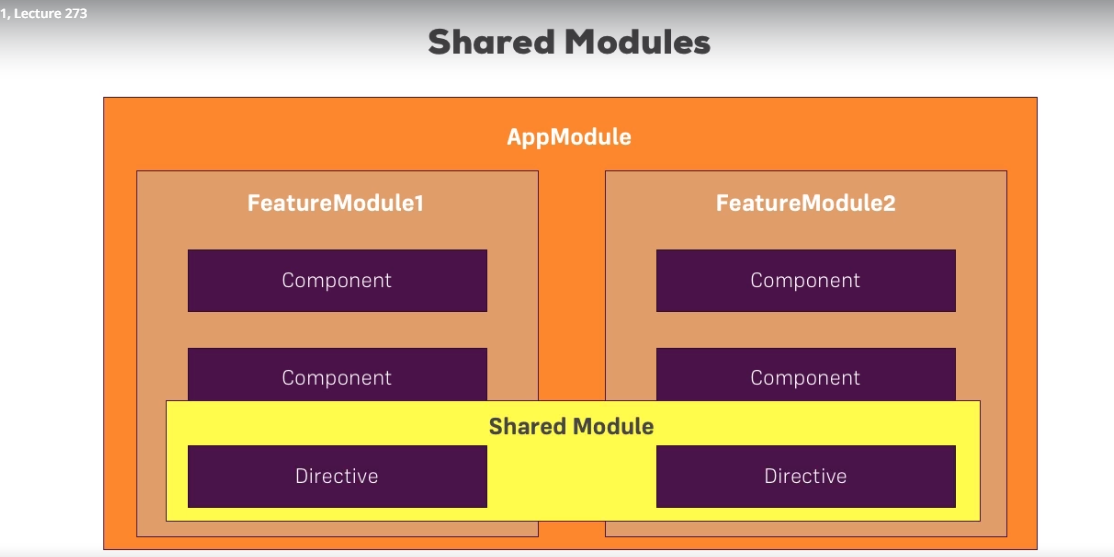
})

export class AppModule { }

**Section 21: Lecture 272//Registering Routes in a Feature Module**

1. Now, we are using app routing module in recipes module but we have declared it only in app module – this is a problem
2. Point is - if we create a feature module we also have to move the routes related to feature modules.
3. Now, we will create a new routing module in the recipes module i.e. recipes-routing.module.ts
4. recipes.routing.module:
5. import { NgModule } from "@angular/core";
6. import { Routes, RouterModule } from "@angular/router";
7. import { RecipesComponent } from "./recipes.component";
8. import { RecipeStartComponent } from "./recipe-start/recipe-start.component";
9. import { RecipeEditComponent } from "./recipe-edit/recipe-edit.component";
10. import { AuthGuard } from "../auth/auth-guard.service";
11. import { RecipeDetailComponent } from "./recipe-detail/recipe-detail.component";
12. const recipesRoutes: Routes = [
13. { path: 'recipes', component: RecipesComponent, children: [
14. { path: '', component: RecipeStartComponent },
15. { path: 'new', component: RecipeEditComponent, canActivate: [AuthGuard] },
16. { path: ':id', component: RecipeDetailComponent },
17. { path: ':id/edit', component: RecipeEditComponent, canActivate: [AuthGuard] },
18. ] }
19. ]
20. @NgModule({
21. imports:[
22. RouterModule.forChild(recipesRoutes)
23. ],
24. exports: [recipesRoutes]
25. })
26. export class RecipesRoutingModule{
27. }
28. recipes.module.ts:
29. import { NgModule } from "@angular/core";
30. import { RecipesComponent } from "./recipes.component";
31. import { RecipeStartComponent } from "./recipe-start/recipe-start.component";
32. import { RecipeListComponent } from "./recipe-list/recipe-list.component";
33. import { RecipeEditComponent } from "./recipe-edit/recipe-edit.component";
34. import { RecipeDetailComponent } from "./recipe-detail/recipe-detail.component";
35. import { RecipeItemComponent } from "./recipe-list/recipe-item/recipe-item.component";
36. import { ReactiveFormsModule } from "@angular/forms";
37. import { CommonModule } from "@angular/common";
38. import { RecipesRoutingModule } from "./recipes-routing.module";
39. @NgModule({
40. declarations:[
41. RecipesComponent,
42. RecipeStartComponent,
43. RecipeListComponent,
44. RecipeEditComponent,
45. RecipeDetailComponent,
46. RecipeItemComponent
47. ],
48. imports: [
49. CommonModule,
50. ReactiveFormsModule,
51. RecipesRoutingModule
52. ]
53. })
54. export class RecipesModule{
56. }
57. app.module.ts:
58. import { BrowserModule } from '@angular/platform-browser';
59. import { NgModule } from '@angular/core';
60. import { FormsModule } from '@angular/forms';
61. import { HttpModule } from '@angular/http';
62. import { AppComponent } from './app.component';
63. import { HeaderComponent } from './header/header.component';
64. import { ShoppingListComponent } from './shopping-list/shopping-list.component';
65. import { ShoppingEditComponent } from './shopping-list/shopping-edit/shopping-edit.component';
66. import { DropdownDirective } from './shared/dropdown.directive';
67. import { ShoppingListService } from './shopping-list/shopping-list.service';
68. import { AppRoutingModule } from './app-routing.module';
69. import { RecipeService } from './recipes/recipe.service';
70. import { DataStorageService } from './shared/data-storage.service';
71. import { SignupComponent } from './auth/signup/signup.component';
72. import { SigninComponent } from './auth/signin/signin.component';
73. import { AuthService } from './auth/auth.service';
74. import { AuthGuard } from './auth/auth-guard.service';
75. import { RecipesModule } from './recipes/recipes.module';
76. @NgModule({
77. declarations: [
78. AppComponent,
79. HeaderComponent,
80. ShoppingListComponent,
81. ShoppingEditComponent,
82. DropdownDirective,
83. SignupComponent,
84. SigninComponent
85. ],
86. imports: [
87. BrowserModule,
88. FormsModule,
89. HttpModule,
90. AppRoutingModule,
91. RecipesModule
92. ],
93. providers: [ShoppingListService, RecipeService, DataStorageService, AuthService, AuthGuard],
94. bootstrap: [AppComponent]
95. })
96. export class AppModule { }

**Section 21: Lecture 273//Understanding Shared Modules**

1. 
2. 
3. 

**Section 21: Lecture 274//Creating a shared Module**

1. Here we will create a new shared module in the shared folder, and then we will export the DropdownDirective from this module.
2. If we create a directive it needs to be declared only once, but at least once.
3. The idea behind the shared module is that, now we can import the shared module into any module.
4. Now, to be able to use the drop down directive we must also export it first.
5. We know that we can only access that things which are available inside the module. To use things outside the module we must be able to export things from certain outside module and then import that module here.
6. Now, we can also add something else to this shared module i.e. the common module.
7. shared.module.ts:
8. import { NgModule } from "@angular/core";
9. import { DropdownDirective } from "./dropdown.directive";
10. import { CommonModule } from "@angular/common";
11. @NgModule({
12. declarations: [
13. DropdownDirective
14. ],
15. exports: [
16. CommonModule,
17. DropdownDirective
18. ]
19. })
20. export class SharedModule{
21. }

8. recipes.module.ts:

import { NgModule } from "@angular/core";

import { RecipesComponent } from "./recipes.component";

import { RecipeStartComponent } from "./recipe-start/recipe-start.component";

import { RecipeListComponent } from "./recipe-list/recipe-list.component";

import { RecipeEditComponent } from "./recipe-edit/recipe-edit.component";

import { RecipeDetailComponent } from "./recipe-detail/recipe-detail.component";

import { RecipeItemComponent } from "./recipe-list/recipe-item/recipe-item.component";

import { ReactiveFormsModule } from "@angular/forms";

import { CommonModule } from "@angular/common";

import { RecipesRoutingModule } from "./recipes-routing.module";

import { SharedModule } from "../shared/shared.module";

@NgModule({

declarations:[

RecipesComponent,

RecipeStartComponent,

RecipeListComponent,

RecipeEditComponent,

RecipeDetailComponent,

RecipeItemComponent

],

imports: [

CommonModule,

ReactiveFormsModule,

RecipesRoutingModule,

SharedModule

]

})

export class RecipesModule{

}

9. app.module.ts:

import { BrowserModule } from '@angular/platform-browser';

import { NgModule } from '@angular/core';

import { FormsModule } from '@angular/forms';

import { HttpModule } from '@angular/http';

import { AppComponent } from './app.component';

import { HeaderComponent } from './header/header.component';

import { ShoppingListComponent } from './shopping-list/shopping-list.component';

import { ShoppingEditComponent } from './shopping-list/shopping-edit/shopping-edit.component';

import { ShoppingListService } from './shopping-list/shopping-list.service';

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

import { RecipeService } from './recipes/recipe.service';

import { DataStorageService } from './shared/data-storage.service';

import { SignupComponent } from './auth/signup/signup.component';

import { SigninComponent } from './auth/signin/signin.component';

import { AuthService } from './auth/auth.service';

import { AuthGuard } from './auth/auth-guard.service';

import { RecipesModule } from './recipes/recipes.module';

import { SharedModule } from './shared/shared.module';

@NgModule({

declarations: [

AppComponent,

HeaderComponent,

ShoppingListComponent,

ShoppingEditComponent,

SignupComponent,

SigninComponent

],

imports: [

BrowserModule,

FormsModule,

HttpModule,

AppRoutingModule,

RecipesModule,

SharedModule

],

providers: [ShoppingListService, RecipeService, DataStorageService, AuthService, AuthGuard],

bootstrap: [AppComponent]

})

export class AppModule { }

**Section 21: Lecture 275//Creating a shopping list feature module**

1. shopping-list.module.ts:
2. import { NgModule } from "@angular/core";
3. import { ShoppingEditComponent } from "./shopping-edit/shopping-edit.component";
4. import { ShoppingListComponent } from "./shopping-list.component";
5. import { CommonModule } from "@angular/common";
6. import { FormsModule } from "@angular/forms";
7. @NgModule({
8. declarations: [
9. ShoppingEditComponent,
10. ShoppingListComponent
11. ],
12. imports: [
13. CommonModule,
14. FormsModule
15. ]
16. })
17. export class ShoppoingListModule{
18. }
19. app.module.ts:
20. import { BrowserModule } from '@angular/platform-browser';
21. import { NgModule } from '@angular/core';
22. import { HttpModule } from '@angular/http';
23. import { AppComponent } from './app.component';
24. import { HeaderComponent } from './header/header.component';
25. import { ShoppingListService } from './shopping-list/shopping-list.service';
26. import { AppRoutingModule } from './app-routing.module';
27. import { RecipeService } from './recipes/recipe.service';
28. import { DataStorageService } from './shared/data-storage.service';
29. import { SignupComponent } from './auth/signup/signup.component';
30. import { SigninComponent } from './auth/signin/signin.component';
31. import { AuthService } from './auth/auth.service';
32. import { AuthGuard } from './auth/auth-guard.service';
33. import { RecipesModule } from './recipes/recipes.module';
34. import { SharedModule } from './shared/shared.module';
35. import { ShoppoingListModule } from './shopping-list/shopping-list.module';
36. @NgModule({
37. declarations: [
38. AppComponent,
39. HeaderComponent,
40. SignupComponent,
41. SigninComponent
42. ],
43. imports: [
44. BrowserModule,
45. HttpModule,
46. AppRoutingModule,
47. RecipesModule,
48. SharedModule,
49. ShoppoingListModule
50. ],
51. providers: [ShoppingListService, RecipeService, DataStorageService, AuthService, AuthGuard],
52. bootstrap: [AppComponent]
53. })
54. export class AppModule { }

**Section 21: Lecture 276//Loading Components via Sectors vs routing**