

# Mobile Application Development

## LECTURE 11

---

# Using a Shared Module

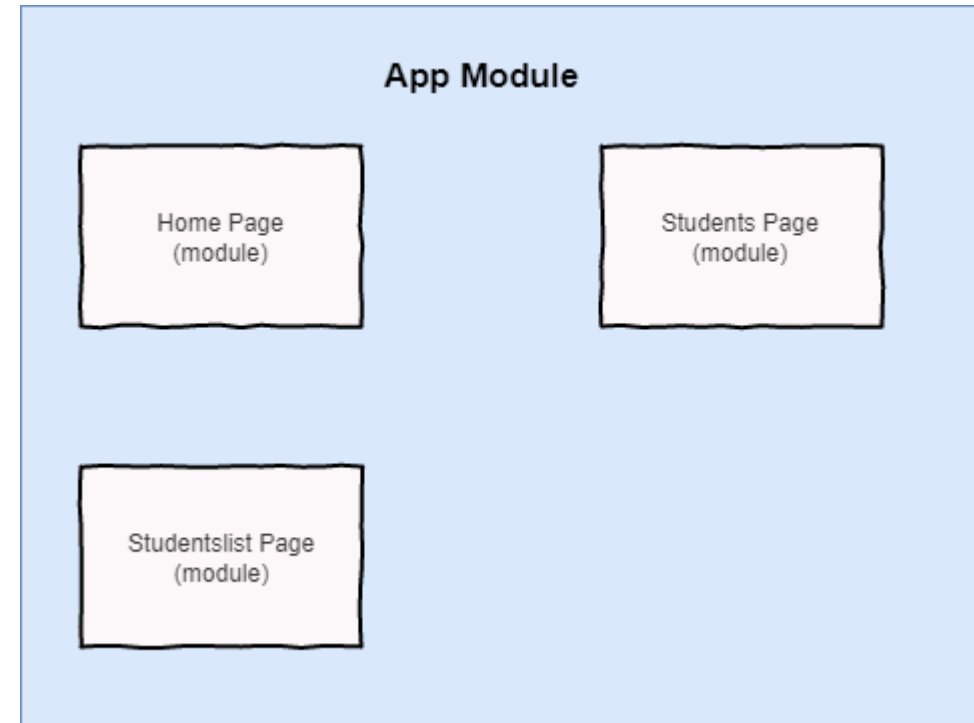
---

- Angular provides us with resusability. But how does it works?
- What if we write a piece of code and we want to use it in many different pages?
- In Angular, we have the concept of Shared Module

# Using a Shared Module

---

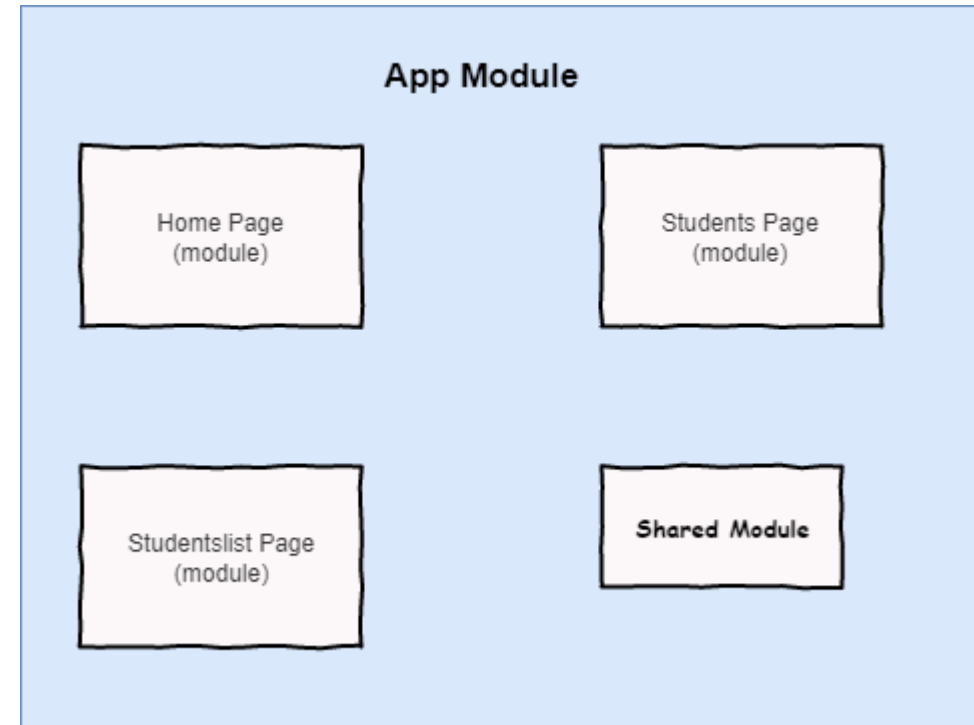
- This is what our structure looks like.
- We have 3 modules inside main App Module.
- what if we want something that will be used in all three modules?



# Using a Shared Module

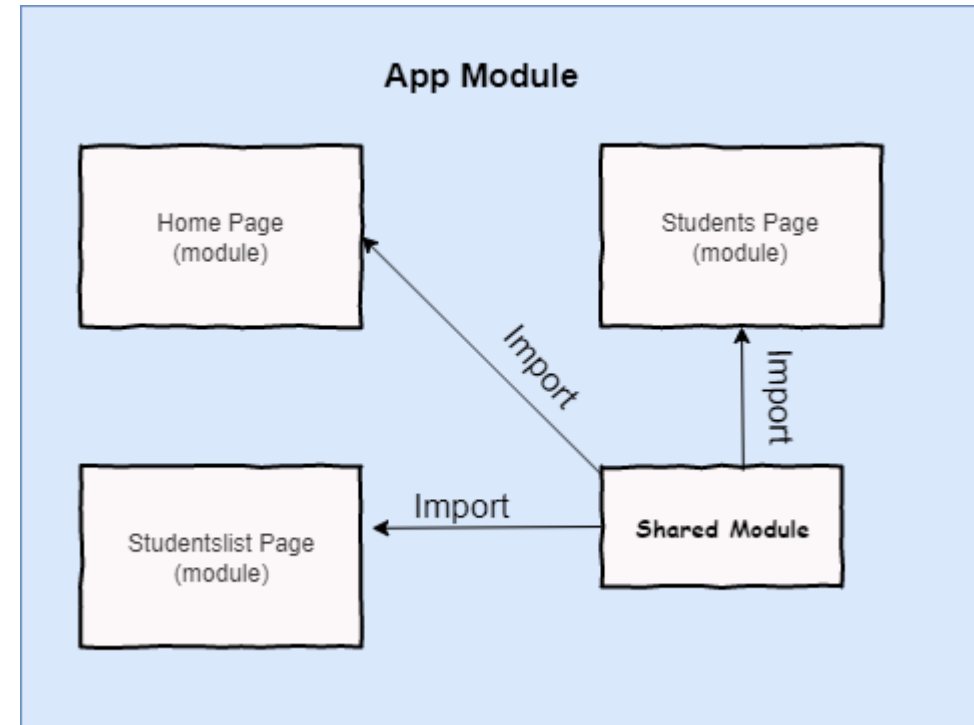
---

- We create a Shared Module.
- How will this work? This Shared Module will be accessible in any other module that we want.

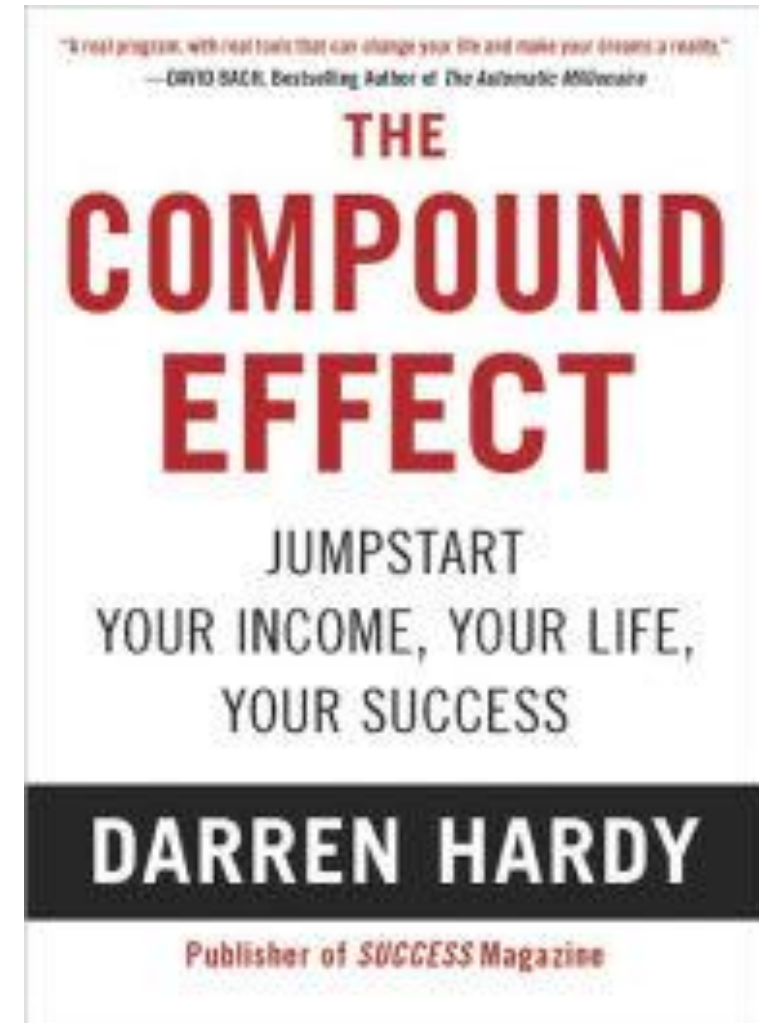


# Using a Shared Module

- But how will it be accessible in other modules? By Importing it in **Imports**.



*You know successful people aren't necessarily more intelligent or more talented than anyone else. But their habits take them in the direction of becoming more informed, more knowledgeable, more competent, better skilled, and better prepared.*



# Using a Shared Module

---

- Lets create a Shared Module.



```
C:\Users\Alamgir\Desktop\MAD-Workbooks\mad-workbooks (lecture9 -> origin)
λ ionic g
? What would you like to generate? module
? Name/path of module: shared
> ng.cmd generate module shared
CREATE src/app/shared/shared.module.ts (192 bytes)
[OK] Generated module!
```

# Using a Shared Module

---

- Our Shared Module has been created. It's empty however. Lets add a component which we will use in different modules. Right now, we are displaying profile for students, we also have to display profile for teachers. So let create profile-detail component inside shared.

```
C:\Users\Alamgir\Desktop\MAD-Workbooks\mad-workbooks (lecture9 -> origin)
λ ionic g
? What would you like to generate? component
? Name/path of component: shared/profile-detail
> ng.cmd generate component shared/profile-detail
CREATE src/app/shared/profile-detail/profile-detail.component.html (33 bytes)
CREATE src/app/shared/profile-detail/profile-detail.component.spec.ts (776 bytes)
CREATE src/app/shared/profile-detail/profile-detail.component.ts (299 bytes)
CREATE src/app/shared/profile-detail/profile-detail.component.scss (0 bytes)
[OK] Generated component!
```



# Using a Shared Module

---

- We have created a component but haven't declared it in any module.
- Lets add it to declaration of shared.module.ts

```
import { ProfileDetailComponent } from './profile-detail/profile-detail.component';
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';

@NgModule({
  declarations: [ProfileDetailComponent],
  imports: [
    CommonModule
  ]
})
export class SharedModule { }
```

# Using a Shared Module

---

- Now in order to use profile-detail component in Students.component.ts, we need to add it to **Exports** in the module as well.
- These exports components will be available only.

```
import { ProfileDetailComponent } from './profile-detail/profile-detail.component';
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';

@NgModule({
  declarations: [ProfileDetailComponent],
  exports: [ProfileDetailComponent],
  imports: [
    CommonModule
  ]
})
export class SharedModule { }
```

# Using a Shared Module

---

- Now we import Shared Module inside students.module.ts

```
@NgModule({  
  imports: [  
    CommonModule,  
    FormsModule,  
    IonicModule,  
    SharedModule,  
    RouterModule.forChild(routes)  
  ],  
  declarations: [StudentPage]  
})  
export class StudentPageModule {}
```

# Using a Shared Module

- Now if we display `<app-profile-detail> </app-profile-detail>` inside the html of `students.component.html`, it will work.
- But it will give an error in `studentslist.component.html` because it is not imported in that module.



# Using a Shared Module

---

- We would get this error. We can resolve this by importing shared module in

studentslist.module.ts

```
✖ ERROR Error: Uncaught (in promise): Error: Template parse errors:
'app-profile-detail' is not a known element:
1. If 'app-profile-detail' is an Angular component, then verify that it is part of this module.
2. If 'app-profile-detail' is a Web Component then add 'CUSTOM_ELEMENTS_SCHEMA' to the '@NgModule.schemas' of this component to suppress this message. ("

<ion-content>
  [ERROR ->]<app-profile-detail></app-profile-detail>

  <ion-card *ngFor="let item of students">
    "): ng:///StudentslistPageModule/StudentslistPage.html@10:2
Error: Template parse errors:
'app-profile-detail' is not a known element:
1. If 'app-profile-detail' is an Angular component, then verify that it is part of this module.
2. If 'app-profile-detail' is a Web Component then add 'CUSTOM_ELEMENTS_SCHEMA' to the '@NgModule.schemas' of this component to suppress this message. ("

<ion-content>
  [ERROR ->]<app-profile-detail></app-profile-detail>
```