

Angular

V Lecture

Modules

A Basic Use of Modules

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

```
@NgModule({  
  imports: [ ... ],  
  declarations: [ ... ],  
  bootstrap: [ ... ]  
})  
export class AppModule { }
```

Adding Components, Pipes and Services to a Module

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

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

import { CreditCardMaskPipe } from './credit-card-mask.pipe';
import { CreditCardService } from './credit-card.service';
import { CreditCardComponent } from './credit-card.component';

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

Creating a Feature Module

```
├── app
│   ├── app.component.ts
│   └── app.module.ts
├── credit-card
│   ├── credit-card-mask.pipe.ts
│   ├── credit-card.component.ts
│   ├── credit-card.module.ts
│   └── credit-card.service.ts
├── index.html
└── main.ts
```

```
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';

import { CreditCardMaskPipe } from './credit-card-mask.pipe';
import { CreditCardService } from './credit-card.service';
import { CreditCardComponent } from './credit-card.component';

@NgModule({
  imports: [CommonModule],
  declarations: [
    CreditCardMaskPipe,
    CreditCardComponent
  ],
  providers: [CreditCardService],
  exports: [CreditCardComponent]
})
export class CreditCardModule {}
```

Creating a Feature Module

```
├── app
│   ├── app.component.ts
│   └── app.module.ts
├── credit-card
│   ├── credit-card-mask.pipe.ts
│   ├── credit-card.component.ts
│   ├── credit-card.module.ts
│   └── credit-card.service.ts
├── index.html
└── main.ts
```

```
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';

import { CreditCardMaskPipe } from './credit-card-mask.pipe';
import { CreditCardService } from './credit-card.service';
import { CreditCardComponent } from './credit-card.component';

@NgModule({
  imports: [CommonModule],
  declarations: [
    CreditCardMaskPipe,
    CreditCardComponent
  ],
  providers: [CreditCardService],
  exports: [CreditCardComponent]
})
export class CreditCardModule {}
```

Services and Lazy Loaded Modules

```
import { NgModule, ModuleWithProviders } from '@angular/core';  
/* ...other imports... */
```

```
@NgModule({  
  imports: [CommonModule],  
  declarations: [  
    CreditCardMaskPipe,  
    CreditCardComponent  
  ],  
  exports: [CreditCardComponent]  
})  
export class CreditCardModule {  
  static forRoot(): ModuleWithProviders {  
    return {  
      ngModule: CreditCardModule,  
      providers: [CreditCardService]  
    }  
  }  
}
```

```
/* ...imports... */
```

```
@NgModule({  
  imports: [  
    BrowserModule,  
    CreditCardModule.forRoot()  
  ],  
  declarations: [AppComponent],  
  bootstrap: [AppComponent]  
})  
export class AppModule { }
```

Lazy Loading a Module

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

```
import { ModuleWithProviders } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';

import { EagerComponent } from './eager.component';

const routes: Routes = [
  { path: '', redirectTo: 'eager', pathMatch: 'full' },
  { path: 'eager', component: EagerComponent },
  { path: 'lazy', loadChildren: 'lazy/lazy.module#LazyModule' }
];

export const routing: ModuleWithProviders =
  RouterModule.forRoot(routes);
```


Lazy Loading a Module

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

```
import { ModuleWithProviders } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';

import { EagerComponent } from './eager.component';

const routes: Routes = [
  { path: '', redirectTo: 'eager', pathMatch: 'full' },
  { path: 'eager', component: EagerComponent },
  { path: 'lazy', loadChildren: 'lazy/lazy.module#LazyModule' }
];

export const routing: ModuleWithProviders =
  RouterModule.forRoot(routes);
```

Shared Modules and Dependency Injection

```
import { NgModule } from '@angular/core';
import { CounterService } from './counter.service';
```

```
@NgModule({
  providers: [CounterService]
})
export class SharedModule {}
```

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

```
@NgModule({
  imports: [
    SharedModule,
    ...
  ],
  declarations: [LazyComponent]
})
export class LazyModule {}
```

```
...
```

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

```
@NgModule({
  imports: [
    SharedModule,
    ...
  ],
  declarations: [
    EagerComponent,
    ...
  ]
})
export class AppModule {}
```

Routing

Configuring Routes

Base URL Tag

```
<base href="/">
```

Route Definition Object

- path - URL to be shown in the browser when application is on the specific route
- component - component to be rendered when the application is on the specific route
- redirectTo - redirect route if needed; each route can have either component or redirect attribute defined in the route (covered later in this chapter)
- pathMatch - optional property that defaults to 'prefix'; determines whether to match full URLs or just the beginning. When defining a route with empty path string set pathMatch to 'full', otherwise it will match all paths.
- children - array of route definitions objects representing the child routes of this route (covered later in this chapter).

```
const routes: Routes = [  
  { path: 'component-one', component: ComponentOne },  
  { path: 'component-two', component: ComponentTwo }  
];
```

RouterModule

```
...  
import { RouterModule, Routes } from '@angular/router';  
  
const routes: Routes = [  
  { path: 'component-one', component: ComponentOne },  
  { path: 'component-two', component: ComponentTwo }  
];  
  
export const routing = RouterModule.forRoot(routes);
```

```
...  
import { routing } from './app.routes';  
  
@NgModule({  
  imports: [  
    BrowserModule,  
    routing  
  ],  
  declarations: [  
    AppComponent,  
    ComponentOne,  
    ComponentTwo  
  ],  
  bootstrap: [ AppComponent ]  
})  
export class AppModule {  
}
```

Redirecting the Router to Another Route

```
export const routes: Routes = [  
  { path: '', redirectTo: 'component-one', pathMatch: 'full' },  
  { path: 'component-one', component: ComponentOne },  
  { path: 'component-two', component: ComponentTwo }  
];
```


Defining Links Between Routes

RouterLink

```
<a routerLink="/component-one">Component One</a>
```

Navigating Programmatically

```
this.router.navigate(['/component-one']);
```

Dynamically Adding Route Components

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

@Component({
  selector: 'app',
  template: `
    <nav>
      <a routerLink="/component-one">Component One</a>
      <a routerLink="/component-two">Component Two</a>
    </nav>

    <router-outlet></router-outlet>
    <!-- Route components are added by router here -->
  `
})
export class AppComponent {}
```

Using Route Parameters

- add a route parameter ID
- link the route to the parameter
- add the service that reads the parameter.

Declaring Route Parameters

```
export const routes: Routes = [  
  { path: '', redirectTo: 'product-list', pathMatch: 'full' },  
  { path: 'product-list', component: ProductList },  
  { path: 'product-details/:id', component: ProductDetails }  
];
```

Linking to Routes with Parameters

```
<a *ngFor="let product of products"  
  [routerLink]="['/product-details', product.id]">  
  {{ product.name }}  
</a>
```

```
goToProductDetails(id) {  
  this.router.navigate(['/product-details', id]);  
}
```

Reading Route Parameters

```
import { Component, OnInit, OnDestroy } from '@angular/core';
import { ActivatedRoute } from '@angular/router';

@Component({
  selector: 'product-details',
  template: `
    <div>
      Showing product details for product: {{id}}
    </div>
  `,
})
export class LoanDetailsPage implements OnInit, OnDestroy {
  id: number;
  private sub: any;

  constructor(private route: ActivatedRoute) {}

  ngOnInit() {
    this.sub = this.route.params.subscribe(params => {
      this.id = +params['id']; // (+) converts string 'id' to a number

      // In a real app: dispatch action to load the details here.
    });
  }

  ngOnDestroy() {
    this.sub.unsubscribe();
  }
}
```

Defining Child Routes

```
export const routes: Routes = [  
  { path: '', redirectTo: 'product-list', pathMatch: 'full' },  
  { path: 'product-list', component: ProductList },  
  { path: 'product-details/:id', component: ProductDetails,  
    children: [  
      { path: '', redirectTo: 'overview', pathMatch: 'full' },  
      { path: 'overview', component: Overview },  
      { path: 'specs', component: Specs }  
    ]  
  }  
];
```

Accessing a Parent's Route Parameters

```
export default class Overview {  
  parentRouteId: number;  
  private sub: any;  
  
  constructor(private router: Router,  
    private route: ActivatedRoute) {}  
  
  ngOnInit() {  
    // Get parent ActivatedRoute of this route.  
    this.sub = this.router.routerState.parent(this.route)  
      .params.subscribe(params => {  
        this.parentRouteId = +params["id"];  
      });  
  }  
  
  ngOnDestroy() {  
    this.sub.unsubscribe();  
  }  
}
```


Links

Prefix	Looks in
/	Root of the application
none	Current component children routes
../	Current component parent routes

Registering the Route Guards with Routes

```
import { Routes, RouterModule } from '@angular/router';
import { AccountPage } from './account-page';
import { LoginRouteGuard } from './login-route-guard';
import { SaveFormsGuard } from './save-forms-guard';
```

```
const routes: Routes = [
  { path: 'home', component: HomePage },
  {
    path: 'accounts',
    component: AccountPage,
    canActivate: [LoginRouteGuard],
    canDeactivate: [SaveFormsGuard]
  }
];
```

```
export const appRoutingProviders: any[] = [];
```

```
export const routing = RouterModule.forRoot(routes);
```

Implementing CanActivate

```
import { CanActivate } from '@angular/router';
import { Injectable } from '@angular/core';
import { LoginService } from '../login-service';

@Injectable()
export class LoginRouteGuard implements CanActivate {

  constructor(private loginService: LoginService) {}

  canActivate() {
    return this.loginService.isLoggedIn();
  }
}
```

Implementing CanDeactivate

```
import { CanDeactivate } from '@angular/router';
import { Injectable } from '@angular/core';
import { AccountPage } from './account-page';

@Injectable()
export class SaveFormsGuard implements CanDeactivate<AccountPage> {

  canDeactivate(component: AccountPage) {
    return component.areFormsSaved();
  }
}
```

Passing Optional Parameters

localhost:3000/product-list?page=2

Passing Query Parameters

```
<a [routerLink]="['product-list']" [queryParams]="{ page: 99 }">Go to Page 99</a>
```

```
goToPage(pageNum) {  
  this.router.navigate(['/product-list'], { queryParams: { page: pageNum } });  
}
```

Reading Query Parameters

```
import { Component } from '@angular/core';  
import { ActivatedRoute, Router } from '@angular/router';
```

```
@Component({  
  selector: 'product-list',  
  template: `<!-- Show product list -->`  
})
```

```
export default class ProductList {  
  constructor(  
    private route: ActivatedRoute,  
    private router: Router) {}
```

```
  ngOnInit() {  
    this.sub = this.route  
      .queryParams  
      .subscribe(params => {  
        // Defaults to 0 if no query param provided.  
        this.page = +params['page'] || 0;  
      });  
  }
```

```
  ngOnDestroy() {  
    this.sub.unsubscribe();  
  }
```

```
  nextPage() {  
    this.router.navigate(['product-list'], { queryParams: { page:  
this.page + 1 } }));  
  }  
}
```

Home task

Add Lazy Loading Modules and routing for home task from lecture2

<https://blackrockdigital.github.io/startbootstrap-clean-blog/contact.html>

<https://angular-2-training-book.rangle.io/handout/modules/>

<https://angular.io/guide/router>

<https://angular-2-training-book.rangle.io/handout/routing/>