Routing and Navigation in Angular SPA

Trayan Iliev

IPT – Intellectual Products & Technologies
e-mail: tiliev@iproduct.org
web: http://www.iproduct.org

Agenda

1. Bootstrapping rooter: *@angular/router*, setting the base href
2. Configuring the router
3. Router navigation using *<a routerLink=”...”>*
4. Programmatic navigation using *Router.navigate()*
5. Retrieving route information
6. Animating transitions for route components
7. confirming or canceling navigation with guards
8. *CanActivate, CanActivateChild, CanDeactivate, CanDeactivateChild* guards
9. Using *resolve* to pre-fetch data before activating a route
10. Lazy loading of feature modules

Where is The Code?

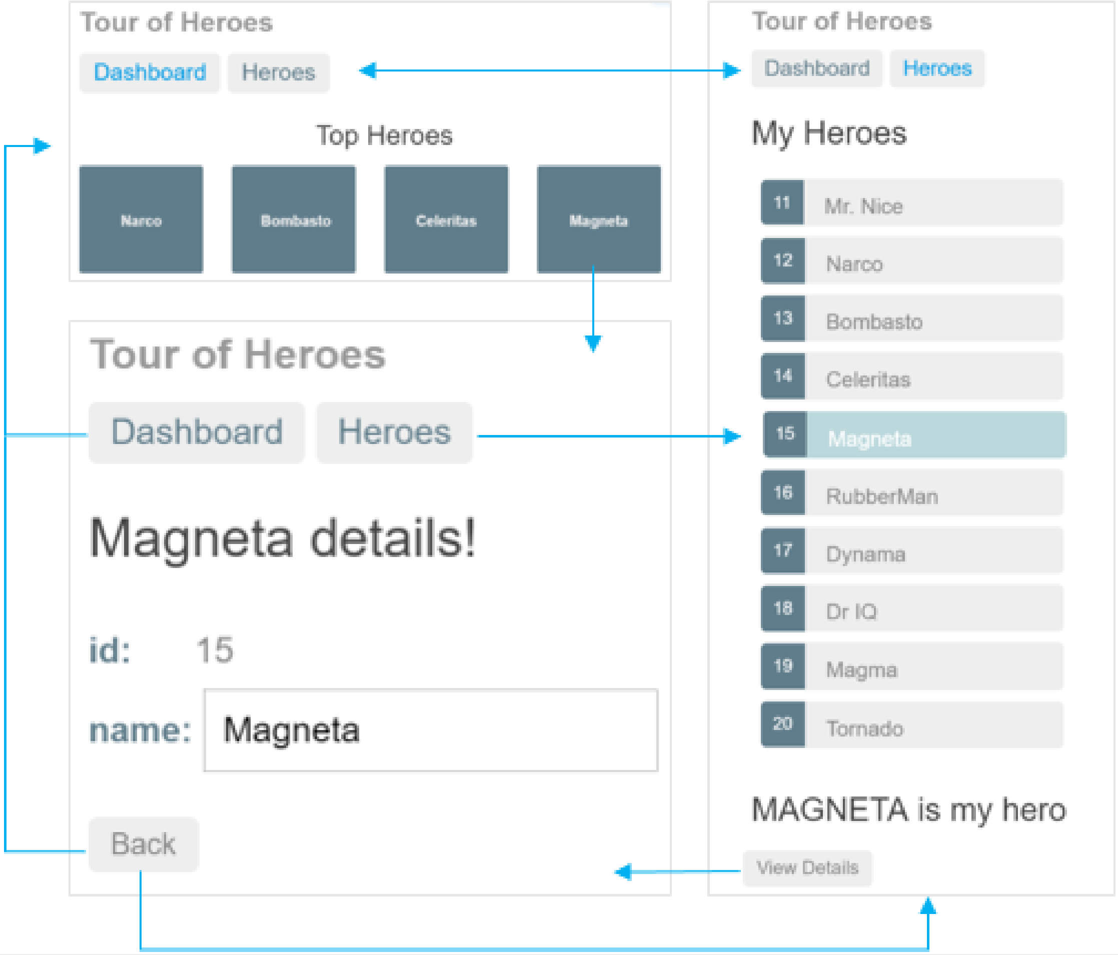
Angular 2 and TypeScript Web App Development code is available @GitHub:

https://github.com/iproduct/course-angular2

Contemporary Web Applications

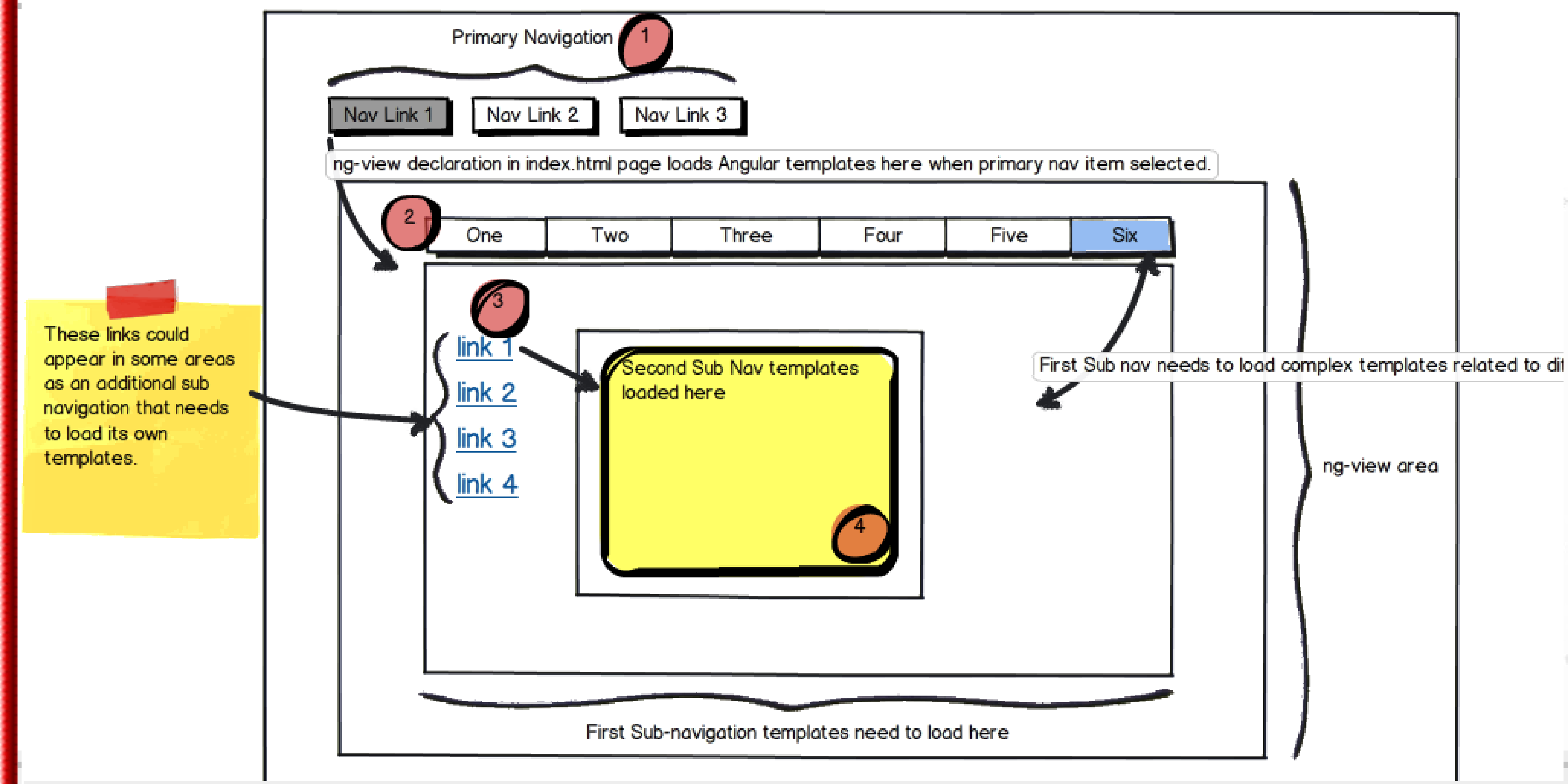
- Provide better User Experience (UX) by:
  - more interactive
  - loading and reacting faster in response (or even anticipation) of user's moves
  - able to work offline
  - supporting multiple devices and screen resolutions (responsive design)
  - are following design metaphors consistently (e.g. Google Material Design - MD)
  - looking more like desktop application than static web page

Single Page Applications (SPA)



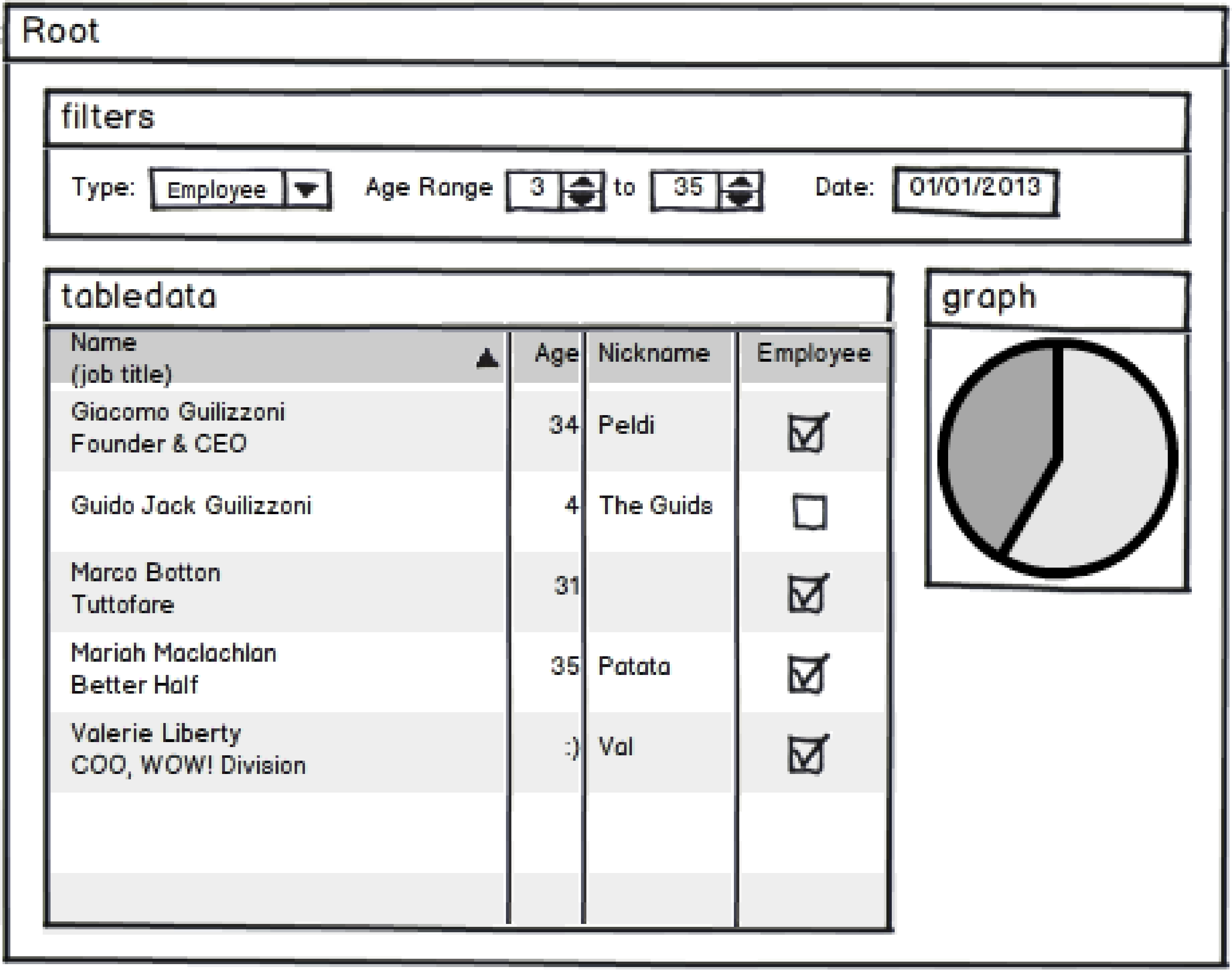
Source: Angular 2 Tutorial: Routing https://angular.io/docs/ts/latest/tutorial/toh-pt5.html
License: CC BY 4.0.

Hierarchical Routing


Primary Navigation
1
Nav Link 1
Nav Link 2
Nav Link 3
ng-view declaration in index.html page loads Angular templates here when primary nav item selected.
2
One
Two
Three
Four
Five
Six
3
These links could appear in some areas as an additional sub navigation that needs to load its own templates.
link 1
link 2
link 3
link 4
Second Sub Nav templates loaded here
4
First Sub nav needs to load complex templates related to di
ng-view area
First Sub-navigation templates need to load here

Source: http://stackoverflow.com/questions/12863663/complex-nesting-of-partials-and-templates
Author: PhillipKregg

SPA with Multiple Router Outlets



Root

filters

Type: Employee ▼ Age Range 3 to 35 Date: 01/01/2013

tabledata

Name (job title) ▲	Age	Nickname	Employee
Giacomo Guilizzoni Founder & CEO	34	Peldi	☑
Guido Jack Guilizzoni	4	The Guids	□
Marco Botton Tuttofare	31		☑
Mariah Maclachlan Better Half	35	Patata	☑
Valerie Liberty COO, WOW! Division	:)	Val	☑

graph

Why SPA?

- Page does not flicker – seamless (or even animated) transitions
- Less data transferred – responses are cached
- Only raw data, not markup
- Features can be loaded on demand (lazy) or in background
- Most page processing happens on the client offloading the server: REST data services + snapshops for crawlers (SEO)
- Code reuse – REST endopints are general purpose
- Supporting multiple platforms (hybrid) → Ionic 2, NativeScript

Bootstrapping & Configuring Angular Router

- **index.html:** `<base href="/" />`
- **app-routing-module.ts:**

```
import { RouterModule } from '@angular/router';
...
@NgModule({
 imports: [ RouterModule.forRoot([
   { path: '', redirectTo: '/home', pathMatch: 'full' },
   { path: 'home', component: HomeComponent },
   { path: 'products', component: ProductListComponent },
   { path: 'users', component: UserListComponent }
   { path: 'users/:userId', component: UserDetailComponent },
   { path: '**', component: PageNotFoundComponent } ]) ],
 exports: [ RouterModule ]
})
export class AppRoutingModule {}
```

Navigation Using **RouterLink** Directive

- **Router links:**

```
<li><a routerLink="/users" routerLinkActive="active">Users</a></li>
<li><a routerLink="/admin" routerLinkActive="active">Admin</a></li>
<li><a routerLink="/login" routerLinkActive="active">Login</a></li>
```

- **Using proprty bindings (** `/users/jim#qualifications?details=true` **):**

```
<a [routerLink]="['/users/jim']" [queryParams]="{details: true}"
   fragment="qualifications">
   link to user component
</a>
```

- **Using parameters array(** `/team/11/users/jim;details=true` **):**

```
<a [routerLink]="['/team', teamId, 'users', userName, {details: true}]">
   link to user component
</a>
```

Router Outlets

- **app.component.html :**

```
<div class="container">
  <app-nav></app-nav>
  <!-- PRIMARY_OUTLET - here are displayed routed views by default -->
  <router-outlet></router-outlet>
  <router-outlet name='left'></router-outlet>
  <router-outlet name='right'></router-outlet>
  <router-outlet name='details' (activate)='onActivate($event)'
     (deactivate)='onDeactivate($event)'></router-outlet>
</div>
```

Handling **ActivatedRoute.params** Observable

[https://angular.io/docs/ts/latest/guide/router.html#!#activated-route]

```
export class ProductListComponent implements OnInit, OnDestroy {
  constructor(private service: ProductService, private route:
    ActivatedRoute, private router: Router, private location:Location){}
  public ngOnInit() {
    this.route.params
      // (+) converts string 'id' to a number
      .switchMap((params: Params) =>this.service.getHero(+params['id']))
      .subscribe((hero: Hero) => this.hero = hero);
  }
  ...
}
```

Programmatic Navigation using **Router.navigate()**

```
export class ProductListComponent implements OnInit, OnDestroy {
  constructor(private service: ProductService, private route:
    ActivatedRoute, private router: Router, private location:Location){}
  public ngOnInit() {
    this.route.params   // highlight previously selected product
      .forEach((params: Params) => {
        this.selectedId = +params['selectedId'];
      });
  }
  public selectItem(product: Product) {
    this.selectedId = product.id;
    this.router.navigate(
         ['.', {selectedId: product.id }], {replaceUrl: true})
      .then(isSucces => this.router.navigate(['/product', product.id]));
  }
```

Routing with Animations I

```
import { animate, AnimationEntryMetadata, state, style, transition,
trigger } from '@angular/core';
export const slideInDownAnimation: AnimationEntryMetadata =
  trigger('routeAnimation', [
    state('*',
      style({ opacity: 1, transform: 'translateX(0)' })
    ),
    transition(':enter', [
      style({ opacity: 0, transform: 'translateX(-100%)' }),
      animate('0.6s ease-in')
    ]),
    transition(':leave', [
      animate('0.6s ease-out',
        style({ opacity: 0, transform: 'translateY(100%)' }))
    ])
  ]);
```

Source: Angular 2 Developer Guide: Routing & Navigation
https://angular.io/docs/ts/latest/guide/router.html#!#route-animation

Routing with Animations II

```
import { Component, HostBinding } from '@angular/core';
import { slideInDownAnimation } from '../common/animations';

@Component({
  selector: 'simple-form',
  templateUrl: './home.component.html',
  animations: [ slideInDownAnimation ]
})
export class HomeComponent {
  @HostBinding('@routeAnimation') routeAnimation = true;
  @HostBinding('style.display')   display = 'block';
  @HostBinding('style.width')   width = '100%';
  @HostBinding('style.position')  position = 'absolute';

  public imageBox = require('../../assets/img/ipt-box.png');
}
```

Routing Guards: CanDeactivate - I

```
@NgModule({
  imports: [
    RouterModule.forChild([
      { path: 'products',  component: ProductListComponent },
       {
        path: 'product/:id',
        component: ProductDetailComponent,
        canDeactivate: [CanDeactivateGuard],
        data: {
          title: 'Edit Product'
        }
      }
    ])],
  Exports: [ RouterModule ]
})
export class ProductRoutingModule {}
```

Routing Guards: CanDeactivate - II

```
@Component({
  selector: 'product-detail',
  templateUrl: './product-detail.component.html'
})
export class ProductDetailComponent implements OnInit, OnChanges,
                                               CanComponentDeactivate {
  public product: Product = { id: undefined };
   constructor(private fb: FormBuilder, private route: ActivatedRoute,
    private router: Router, private location: Location, private service:
    ProductService, private dialogService: DialogService) { }

    canDeactivate(): Promise<boolean> | boolean {
      if ( shallowEquals(this.product, this.productForm.getRawValue())){
        return true;
      }
      return this.dialogService.confirm('Discard changes?');
    }  ...
```

Resolving Route Data Asynchronously I

```
@NgModule({
  imports: [
    RouterModule.forChild([
      { path: 'products',  component: ProductListComponent },
       {
        path: 'product/:id',
        component: ProductDetailComponent,
        canDeactivate: [CanDeactivateGuard],
        data: { title: 'Edit Product' },
        resolve: { product: ProductResolver }
      }
  exports: [RouterModule ],
  providers: [ ProductResolver ]
})
export class ProductRoutingModule {}
```

Resolving Route Data Asynchronously II

```
import { Injectable } from '@angular/core';
import { Router, Resolve, ActivatedRouteSnapshot } from
                                                       '@angular/router';
import { Product } from './product.model';
import { ProductService } from './product.service';

@Injectable()
export class ProductResolver implements Resolve<Product> {

  constructor(private service: ProductService, private router: Router){}

  public resolve(route: ActivatedRouteSnapshot): Promise<Product> {
    let id = +route.params['id'];
    return this.service.refreshProducts().then(() =>
      this.service.getProductObservable(id).take(1).toPromise());
  }
}
```

Lazy Loading of Feature Modules

```
import { Routes } from '@angular/router';
import { NotFoundComponent } from './ui/components/not-found';
export const routes: Routes = [
  {
    path: '',
    redirectTo: '/users',
    pathMatch: 'full' },
  {
    path: 'tests',
    loadChildren: './tests/test.module#TestModule'
  },
  {
    path: 'users',
    loadChildren: './users/user.module#UserModule'
  },
  {
    path: '**',
    component: NotFoundComponent
  }
];
```

Lazy Loading – Child RoutingModule (1)

```
@NgModule({
  imports: [
    RouterModule.forChild([
      {
        path: '',
        component: UserListComponent,
        children: [{
            path: 'new',
            pathMatch: 'full',
            component: UserDetailComponent,
            data: { title: 'Add New User' }
          }, {
            path: ':id',
            component: UserDetailComponent,
            canDeactivate: [CanDeactivateGuard],
            data: { title: 'Edit User' },
            resolve: { user: UserResolver }
          }]
      }]) ],
```

Lazy Loading – Child RoutingModule (2)

```
  ...

  exports: [
    RouterModule
  ],
  providers: [
    UserResolver
  ]
})
export class UserRoutingModule { }
```

Thanks for Your Attention!

Questions?