11강. Routing

DEVELOP ACROSS ALL PLATFORMS

Learn one way to build applications with Angular and reuse your code and abilities to build apps for any deployment target. For web, mobile web, native mobile and native desktop.

11 강.

라우팅 (routing)

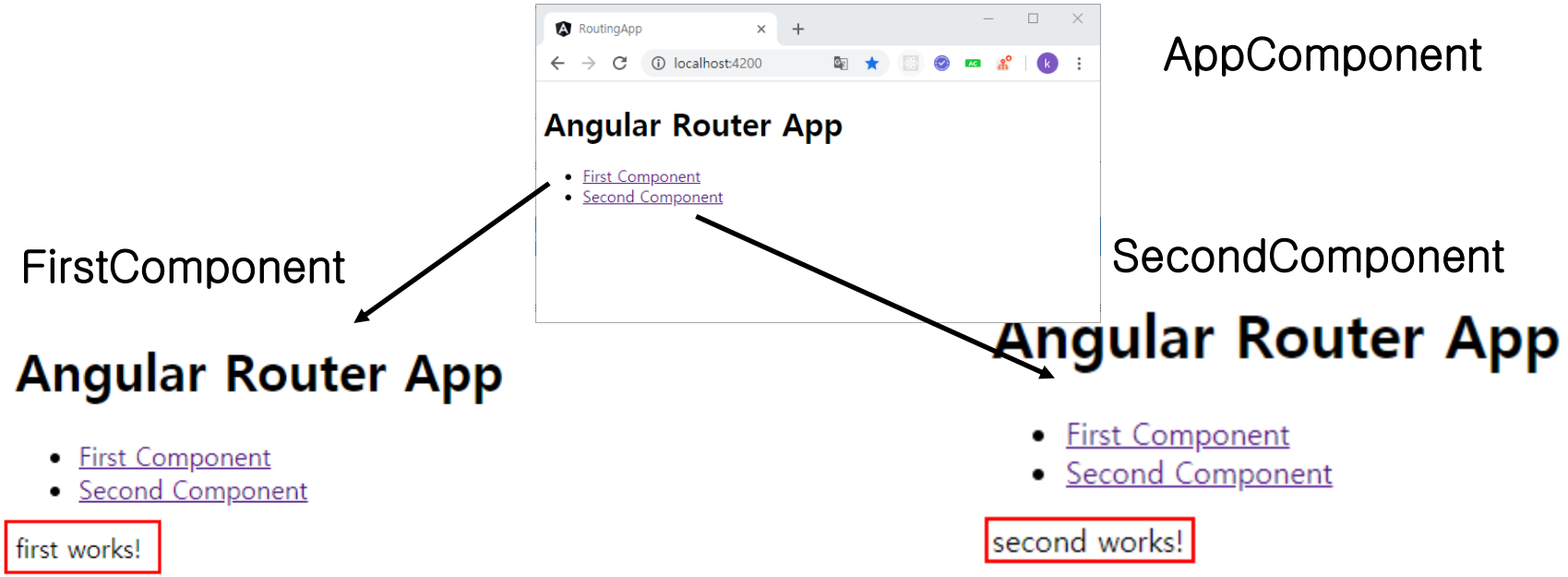
❑ 1) 라우팅 (routing) 개요

•라우팅 역할

- SPA (Single Page Application) 개발시 **네비게이션 기능으로** 사용 가능.
- 애플리케이션 상태 유지 가능
- 모듈화 된 애플리케이션 개발 가능 (기능별)
- Role 기반의 애플리케이션 개발 가능

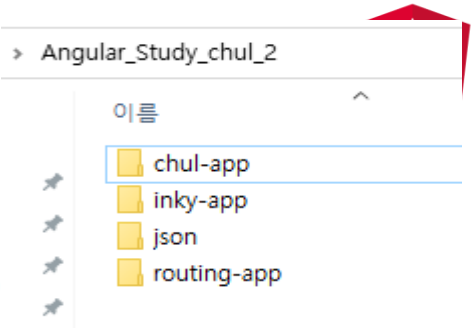
* 기본 실습

- first 링크와 second 링크를 이용한 화면 전환 애플리케이션 구축



❑ 2) 라우팅 (routing) 실습

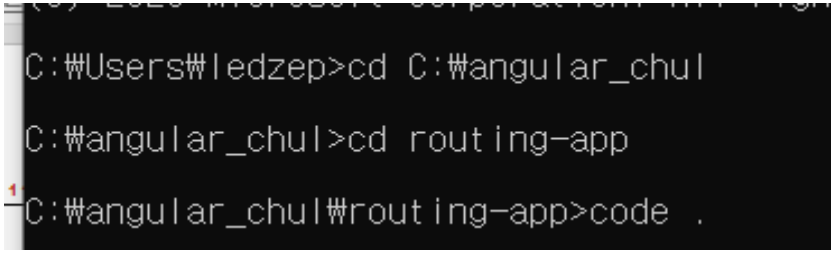
1) 라우팅 기능을 포함하는 프로젝트 새로 생성



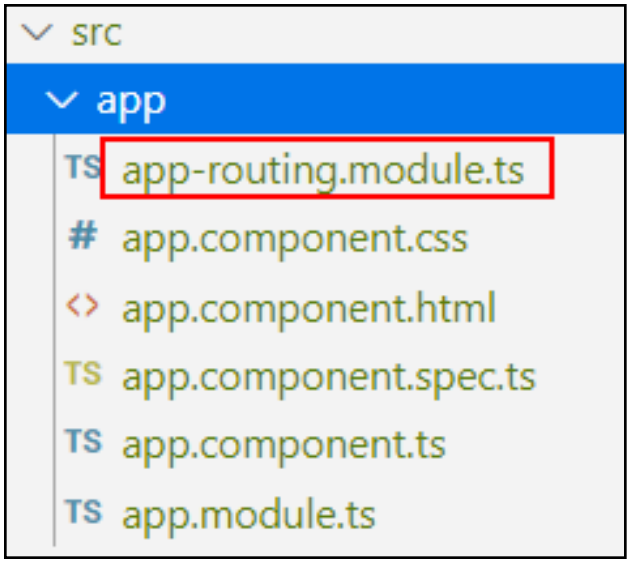

```
ng new routing-app --routing
```

C:\angular_chul\chul-app>cd..

C:\angular_chul>ng new routing-app --routing



https://angular.io/guide/router



app-routing.module.ts

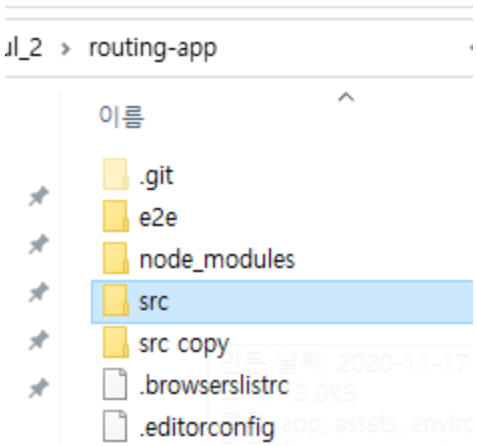
uting-app > src > app > TS app-routing.module.ts > ...

```
1   import { NgModule } from '@angular/core';
2   import { Routes, RouterModule } from '@angular/router';
3
4
5   const routes: Routes = [];
6
7   @NgModule({
8     imports: [RouterModule.forRoot(routes)],
9     exports: [RouterModule]
10  })
11  export class AppRoutingModule { }
```

2) FirstComponent 와 SecondComponent 생성


```
C:\angular\routing-app>ng g component First
CREATE src/app/first/first.component.html (20 bytes)
CREATE src/app/first/first.component.spec.ts (621 bytes)
CREATE src/app/first/first.component.ts (271 bytes)
CREATE src/app/first/first.component.css (0 bytes)

C:\angular\routing-app>ng g component Second
CREATE src/app/second/second.component.html (21 bytes)
CREATE src/app/second/second.component.spec.ts (628 bytes)
CREATE src/app/second/second.component.ts (275 bytes)
CREATE src/app/second/second.component.css (0 bytes)
```



❑ 2) 라우팅 (routing) 실습

3) Routes 설정

- Angular의 Routes는 사용자가 요청한 URL을 해석하고 출력을 담당하는 컴포넌트와 연결하는 역할을 담당. app-routing.module.ts 에 추가

```
const routes: Routes = [
  { path: 'first-component', component: FirstComponent },
  { path: 'second-component', component: SecondComponent },
];
```

4) template에서 routerLink와 router-outlet(보여지는 부분) 설정

```
<h1>Angular Router App</h1>
<!-- This nav gives you links to click, which tells the router which route to use (defined in th
AppRoutingModule) -->
<nav>
  <ul>
    <li><a routerLink="/first-component" routerLinkActive="active">First Component</a></li>
    <li><a routerLink="/second-component" routerLinkActive="active">Second Component</a></li>
  </ul>
</nav>
<!-- The routed views render in the <router-outlet>-->
<router-outlet></router-outlet>
```

❑ App-routing.module.ts

TS app-routing.module.ts × | app.component.html | TS page-not-found-component.component.ts | first.

src > app > TS app-routing.module.ts > ...

```
1  import { NgModule } from '@angular/core';
2  import { Routes, RouterModule } from '@angular/router';
3  import {FirstComponent} from './first/first.component';
4  import {SecondComponent} from './second/second.component';
5  import {PageNotFoundComponentComponent} from "./page-not-found-component/pa
6  //html의 링크값을 설정
7  const routes: Routes = [
8    {path:'first-component', component:FirstComponent}, //컴포넌트 주소등록
9    {path:'second-component', component:SecondComponent}, //컴포넌트 주소등록
10   {path:"**", component:PageNotFoundComponentComponent}
11 ];
12
13 @NgModule({
14   imports: [RouterModule.forRoot(routes)],
15   exports: [RouterModule]
16 })
```

❑ App.component.html 수정

(R) 터미널(T) 도움말(H) app.component.html - routing-app - Visual Studio Code

TS app-routing.module.ts | app.component.html × | TS page-not-found-component.component.ts

src > app > app.component.html > router-outlet

```
1  <h1>Angluar Router App</h1>
2  <nav>
3    <ul>
4      <li><a routerLink="/first-component">First-component</a></li>
5      <li><a routerLink="/second-component">Second-component</a></li>
6    </ul>
7  </nav>
8  <!-- 아래 영역에 링크된 컴포넌트가 보여진다. -->
9  <router-outlet></router-outlet>
```

RoutingApp × RoutingApp

localhost:4200/first-component

앱 3D프린터 교육과정 기타강좌 데이터

Angluar Router App

- First-component
- Second-component

first Component

RoutingApp × RoutingApp

localhost:4200/second-component

앱 3D프린터 교육과정 기타강좌 데이터베

Angluar Router App

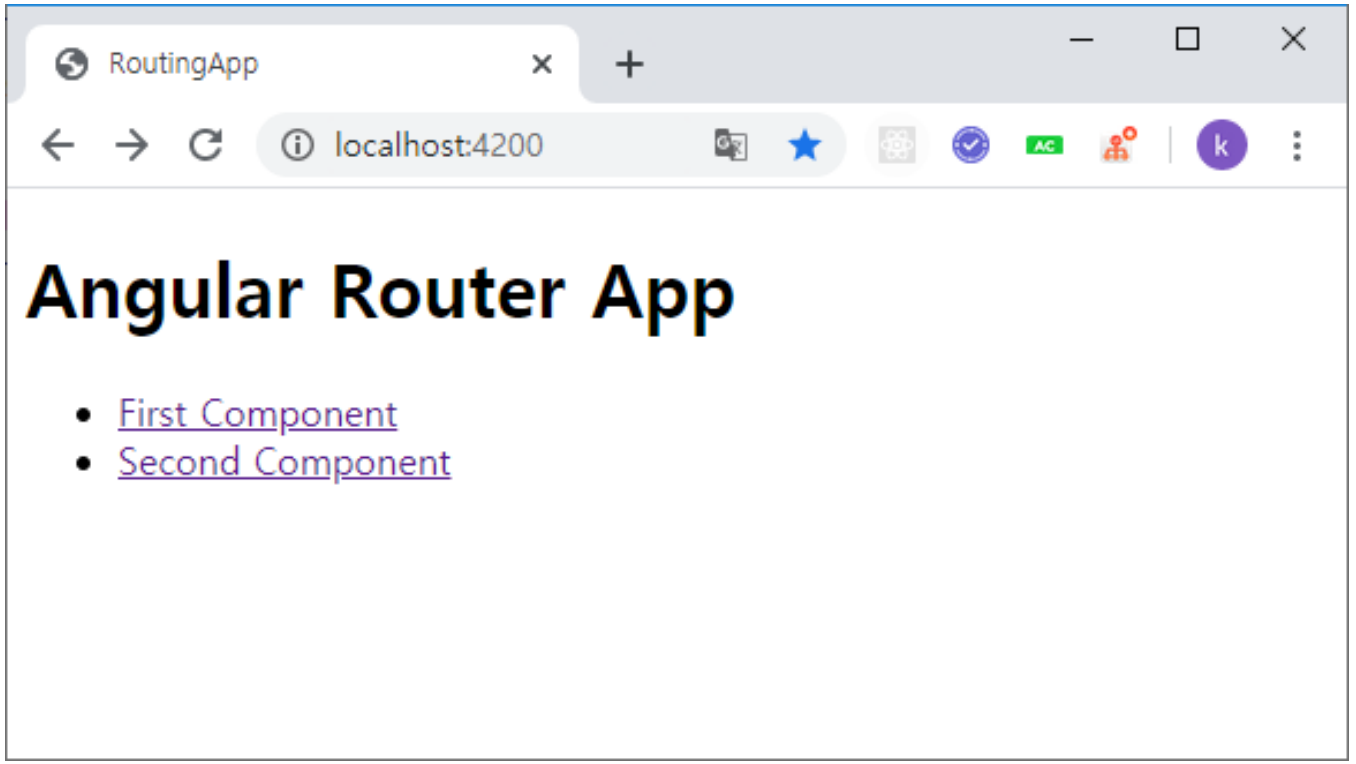
- First-component
- Second-component

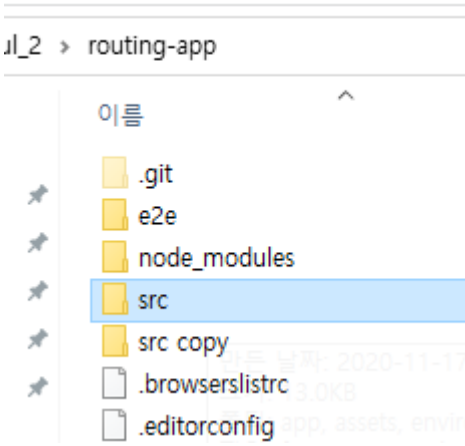
second component

```
PS C:\Angular_Study_chul_2\routing-app> ng serve
```

❑ 2) 라우팅 (routing) 실습

5) 실행





❑ 3) FileNotFound - 404 처리

- ** 와일드 카드 route 설정

사용자가 존재하지 않는 path를 요청할 때 정상적으로 처리하는 방법이다.
즉, 요청된 URL이 라우터 path와 일치하지 않을 때 이 경로가 선택된다.

```
C:\angular\routing-app>ng g component PageNotFound
CREATE src/app/page-not-found/page-not-found.component.html (29 bytes)
CREATE src/app/page-not-found/page-not-found.component.spec.ts (672 bytes)
CREATE src/app/page-not-found/page-not-found.component.ts (305 bytes)
CREATE src/app/page-not-found/page-not-found.component.css (0 bytes)
UPDATE src/app/app.module.ts (663 bytes)
```

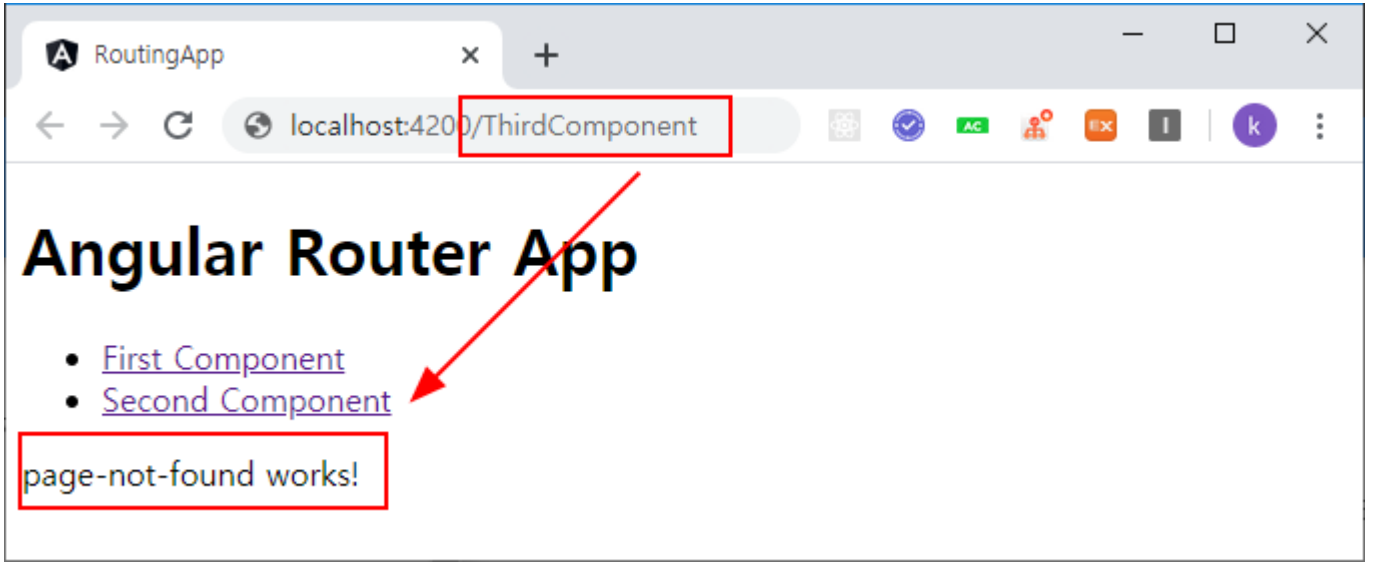
```
4  import {SecondComponent} from './sec d/page-not-found.component"
5  import {PageNotFoundComponent} from './page-not-found/page-not-found.component';
```

```
const routes: Routes = [
  { path: 'first-component', component: FirstComponent },
  { path: 'second-component', component: SecondComponent },
  { path: '',   redirectTo: '/first-component', pathMatch: 'full' }, // redirect to `first-component`
  { path: '**', component: FirstComponent },
  { path: '**', component: PageNotFoundComponent },  // Wildcard route for a 404 page
];
```

❑ 3) FileNotFound - 404 처리

```
import { FirstComponent } from './first/first.component';
import { SecondComponent } from './second/second.component';
import { PageNotFoundComponent } from './page-not-found/page

const routes: Routes = [
  { path: 'first-component', component: FirstComponent },
  { path: 'second-component', component: SecondComponent },
  { path: '**', component: PageNotFoundComponent }
];
```



❑ 4) 리다이렉션 처리

*리다이렉션(redirection) ?

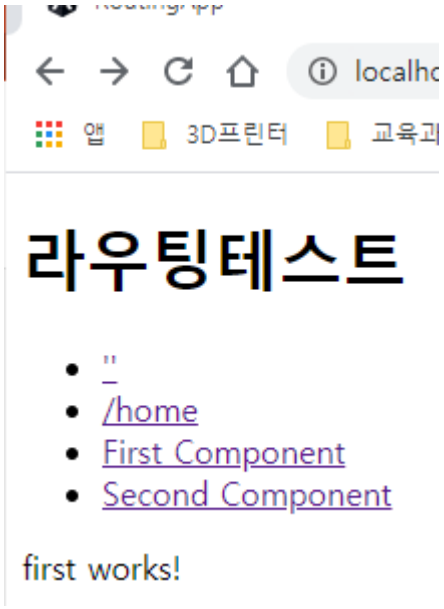
- 리다이렉션이란 사용자의 요청을 우회시키는 방법을 의미하다.
- Routes의 redirectTo 속성과 pathMatch 속성을 이용하여 구현한다.

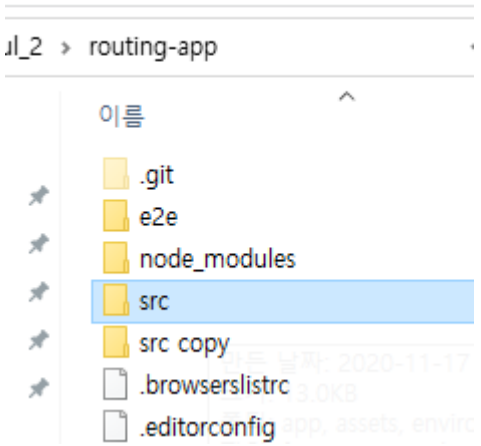
app-routing.module.ts | page-not-found.component.html

src > app > app-routing.module.ts > ...

```
1  import { NgModule } from '@angular/core';
2  import { Routes, RouterModule } from '@angular/router';
3  import {FirstComponent} from './first/first.component';
4  import {SecondComponent} from './second/second.component';
5  import {PageNotFoundComponent} from './page-not-found/page-not-found.component';
6  const routes: Routes = [
7  {path:'', redirectTo:'/first-component',pathMatch:'full'},//주소지정, 컴포넌트지정
8  {path:'home', redirectTo:'/first-component'  , pathMatch:'full'},//주소지정, 컴포넌트지정
9  {path:'first-component', component:FirstComponent},//주소지정, 컴포넌트지정
10 {path:'second-component', component:SecondComponent},
11 {path:'**', component:PageNotFoundComponent}  //404처리를 위한 처리
12 ];
13
```

❑ 리다이렉트의 실행

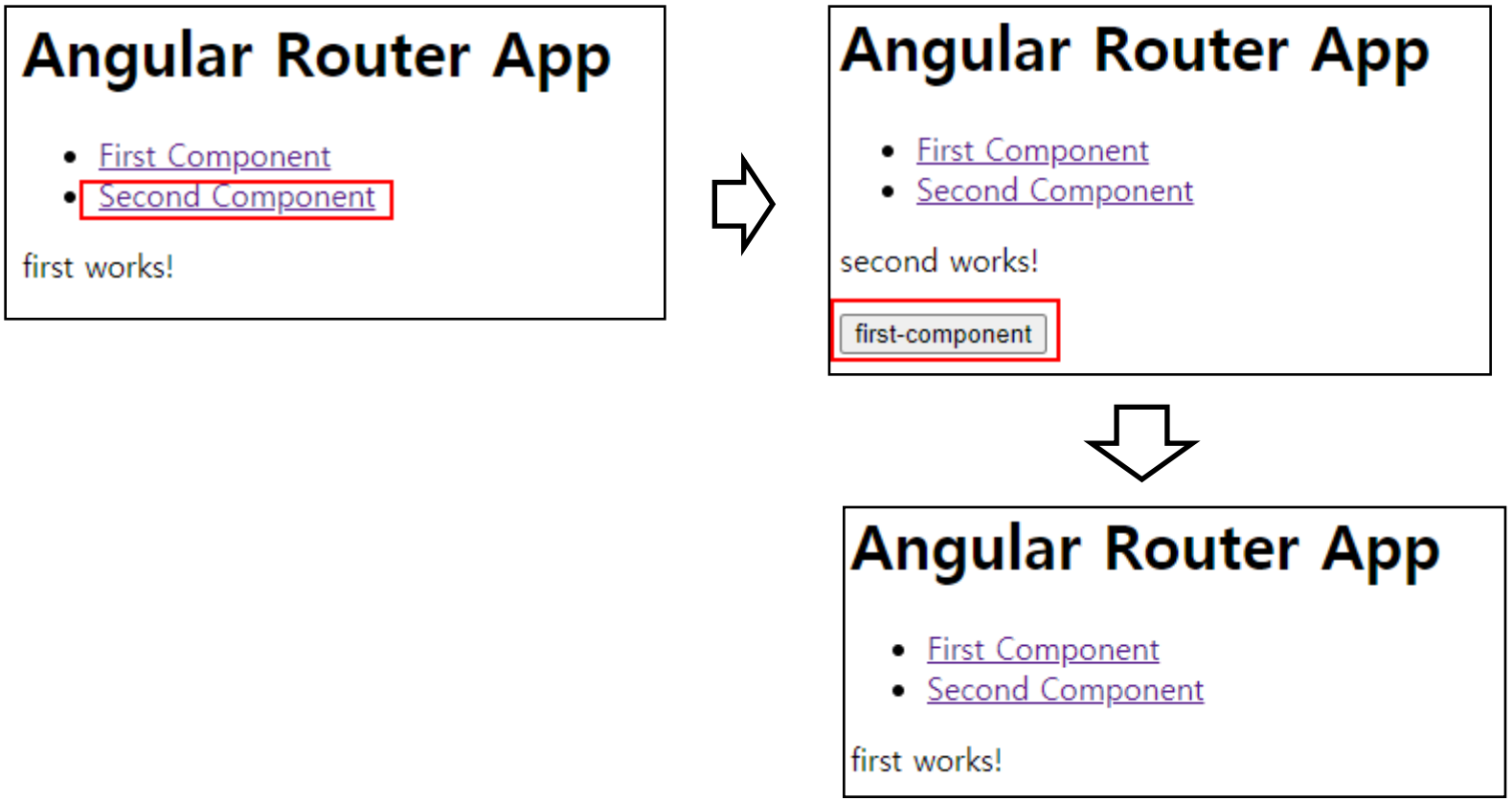

localhc
앱
3D프린터
교육과
라우팅테스트
''
/home
First Component
Second Component
first works!


routing-app
이름
.git
e2e
node_modules
src
src copy
.browserslistrc
.editorconfig

❑ 5) 코드 명령어를 사용한 라우팅 처리

문법:

```
this.router.navigate([타겟path]);
```



❑ 5) 코드 명령어를 사용한 라우팅 처리

second.component.html

```
uting-app > src > app > second > <> second.compo
1   <p>second works!</p>
2   <button (click)="goFirst()">
3       first-component</button>
```

second.component.ts

```
uting-app > src > app > second > TS second.component.ts > ...
1   import { Component, OnInit } from '@angular/core';
2   import { Router} from '@angular/router';
3   @Component({
4     selector: 'app-second',
5     templateUrl: './second.component.html',
6     styleUrls: ['./second.component.css']
7   })
8   export class SecondComponent{
9
10    constructor(private router:Router) { }
11
12    goFirst(){
13      console.log("goFirst")
14      this.router.navigate(['first-component'])
15    }
16  }
```

```
PS C:\Angular_Study_chul_2\routing-app> ng serve
```

❑ 6) 파라미터 전송 1- URL 형식

1) template에서 파라미터값을 설정한다.

Restful 형식으로 파라미터를 전달할 수 있다.

first.component.ts | first.component.html | app.component.html × | app-routing.module.ts

src > app > app.component.html > nav > ul > li > a

```
1  <h1>Angluar Router App</h1>
2  <nav>
3    <!--
4      파라미터 전송시url 형태로 :Restful서비스 <==>SOAP
5      SOAP방식 :8080/target?key=value?key2=value
6      RestFul방식 : 8080/target/key/value/key2/value
7      <li><a routerLink="/first-componen/seoul/1000" routerLinkActive="active">First-component/seoul
8      first-compoenent/:city/:pop
9    -->
10   <ul>
11     <li><a routerLink="/first-component/seoul/1000" routerLinkActive="active">First-component/seou
12     <li><a routerLink="/first-component/pusan/500" routerLinkActive="active">First-component2</a><
13     <li><a routerLink="/second-component" routerLinkActive="active">Second-component</a></li>
```

❑ 파라미터 전송 1- URL 형식

2) Routes에서 요청 path와 일치하도록 경로 설정.

형식: 요청path/:변수명

p.component.html | TS app-routing.module.ts × | TS first.component.ts

app > TS app-routing.module.ts > routes > path

```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';
import {FirstComponent} from './first/first.component';
import {SecondComponent} from './second/second.component';
import {PageNotFoundComponentComponent} from "./page-not-found-component/page-not-fc
//html의 링크값을 설정
const routes: Routes = [
  {path:'first-component/:city/:pop', component:FirstComponent}, //컴포넌트 주소등록
```

❑ 6) 파라미터 전송 1 - URL 형식

3) ActivatedRoute 사용한 파라미터 얻기

app.component.html | first.component.ts × | app-routing.module.ts

src > app > first > first.component.ts > FirstComponent

```
1  import { Component, OnInit } from '@angular/core';
2  import {ActivatedRoute} from '@angular/router';
3  @Component({
4    selector: 'app-first',
5    templateUrl: './first.component.html',
6    styleUrls: ['./first.component.css']
7  })
8  export class FirstComponent implements OnInit{
9    constructor(public rotue:ActivatedRoute) { }
10   params;
11   params2;
12   //FirstComponent가 초기화 될 때 호출
13   ngOnInit(): void {
14    this.rotue.paramMap.subscribe(res=>{this.params=res.get("city"); this.params2=res.get("pop");})
15   }
16
17 }
```

```
PS C:\Angular_Study_chul_2\routing-app> ng serve
```

❑ 6) 파라미터 전송 1- URL 형식

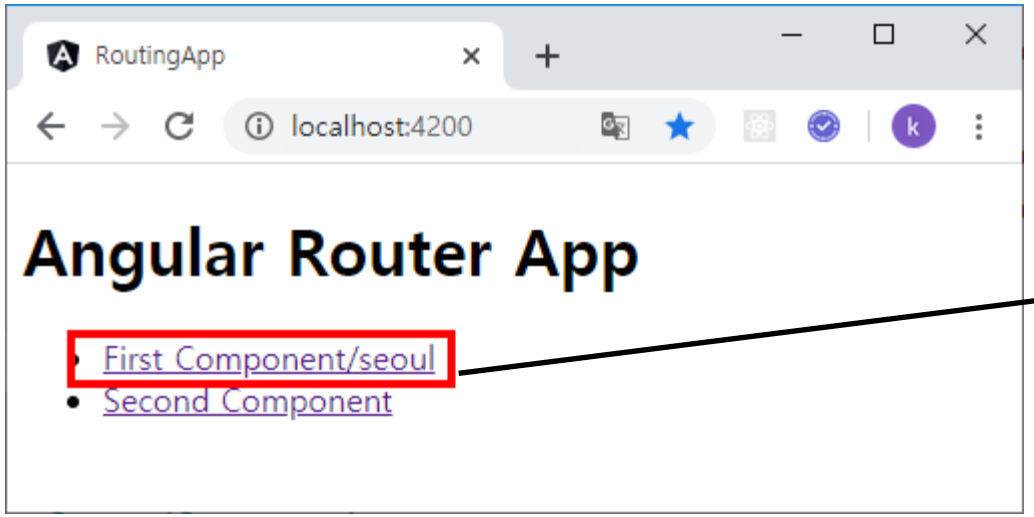
4) Template 에서 파라미터 출력

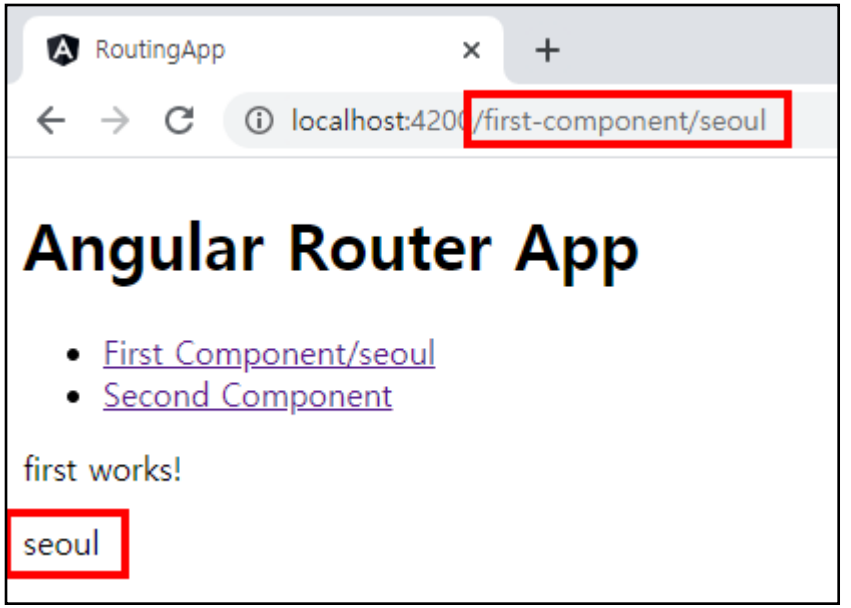
TS first.component.ts | <> first.component.html × | <> app.co

src > app > first > <> first.component.html > ...

```
1   <p>first Component</p>
2   {{params}} : {{params2}}
```

5) 실행





❑ 6) 파라미터 전송 2- 세그먼트 형식

1) template에서 파라미터값을 설정한다.

JSON 형식으로 파라미터를 전달할 수 있다.

app.component.html

```
<h1>Angular Router App</h1>
<!-- This nav gives you links to click, which tells the router which route to use (
<nav>
  <ul>
    <li><a routerLink="/first-component/seoul" routerLinkActive="active">
      First Component/seoul</a></li>
    <li><a [routerLink]="['/second-component', {userid:'master',passwd:'1234'}]"
            routerLinkActive="active">
      Second Component</a></li>
  </ul>
</nav>
<!-- The routed views render in the <router-outlet>-->
<router-outlet></router-outlet>
```

❑ 6) 파라미터 전송 2- 세그먼트 형식

2) Routes에서 요청 path와 일치하도록 경로 설정

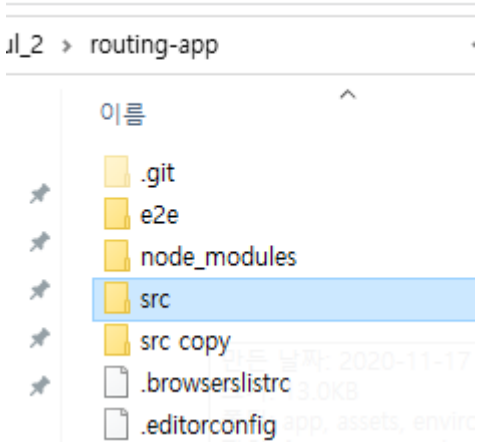
```
const routes: Routes = [
  { path: 'first-component/:city', component: FirstComponent },
  { path: 'second-component', component: SecondComponent },
];
```

3) ActivatedRoute 사용한 파라미터 얻기

```
import { ActivatedRoute } from '@angular/router';
```

```
export class SecondComponent implements OnInit {

  constructor(private route:ActivatedRoute) { }
  user={
    userid:"",
    passwd:""
  }
  ngOnInit(): void {
    this.route.paramMap.subscribe(
      params=>{this.user.userid=params.get("userid");
               this.user.passwd=params.get("passwd");})
  }
}
```



❑ 6) 파라미터 전송 - URL 형식

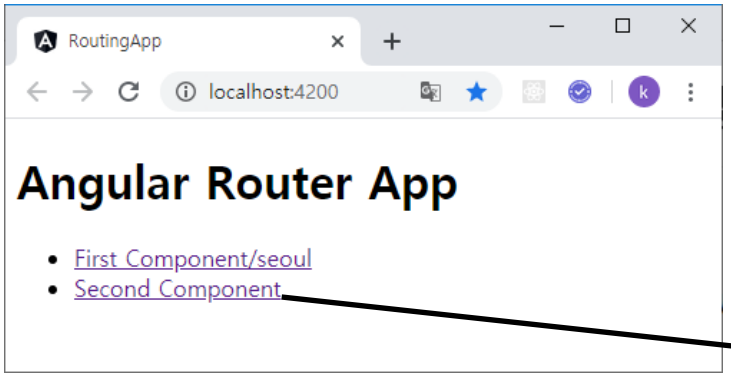
4) Template 에서 파라미터 출력

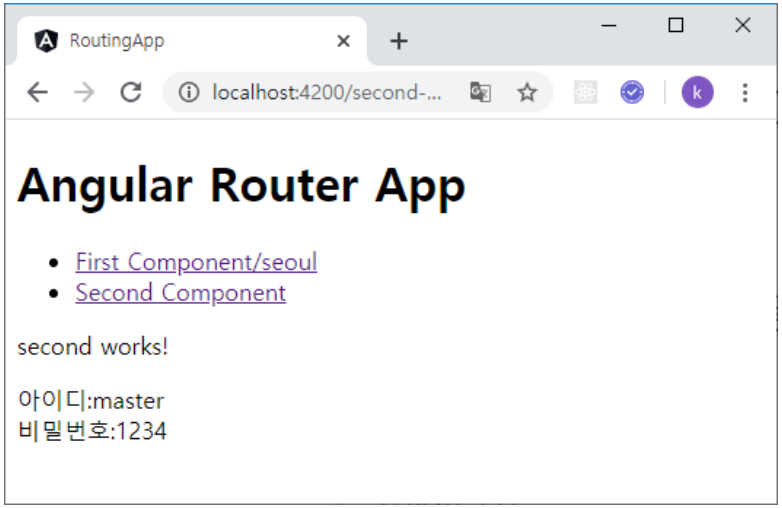
second.component.html

routing-app > src > app > second > second.componen

```
1   <p>second works!</p>
2   아이디:{{user.userid}}<br>
3   비밀번호:{{user.passwd}}<br>
```

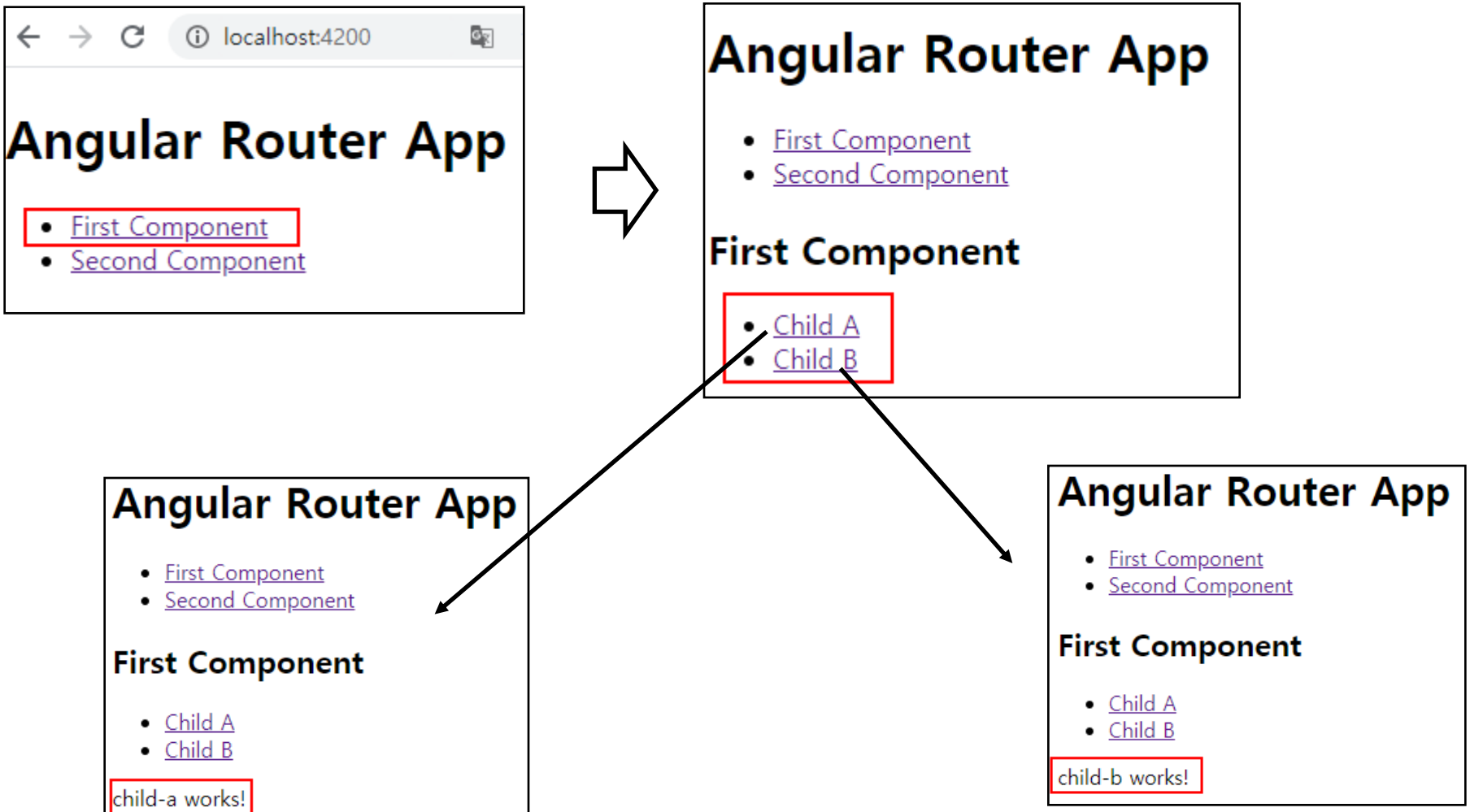
5) 실행





❑ 가 sub링크 구축

1) 실행 결과



❑ 가 sub링크 구축

2) FirstComponent의 자식으로 ChildA 와 ChildB 컴포넌트 생성

```
C:\angular_chul\routing-app>ng g component first/ChildA
CREATE src/app/first/child-a/child-a.component.html (22 bytes)
CREATE src/app/first/child-a/child-a.component.spec.ts (627 bytes)
CREATE src/app/first/child-a/child-a.component.ts (278 bytes)
CREATE src/app/first/child-a/child-a.component.css (0 bytes)
UPDATE src/app/app.module.ts (791 bytes)

C:\angular_chul\routing-app>ng g component first/ChildB
CREATE src/app/first/child-b/child-b.component.html (22 bytes)
CREATE src/app/first/child-b/child-b.component.spec.ts (627 bytes)
CREATE src/app/first/child-b/child-b.component.ts (278 bytes)
CREATE src/app/first/child-b/child-b.component.css (0 bytes)
UPDATE src/app/app.module.ts (881 bytes)
```

❑ 가 sub링크 구축

3) Routes에서 Child 컴포넌트를 FirstComponent 의 children으로 추가

```
const routes: Routes = [
  { path: 'first-component',
    component: FirstComponent, // this is the component with the <router-outlet> in the template
    children: [
      {
        path: 'child-a', // child route path
        component: ChildAComponent // child route component that the router renders
      },
      {
        path: 'child-b',
        component: ChildBComponent // another child route component that the router renders
      }
    ] },
  { path: 'second-component', component: SecondComponent },
];
```

❑ 가 sub링크 구축

4) FirstCompnent의 template에서 sub링크 추가

first.component.html

outing-app > src > app > first > first.component.html > ...

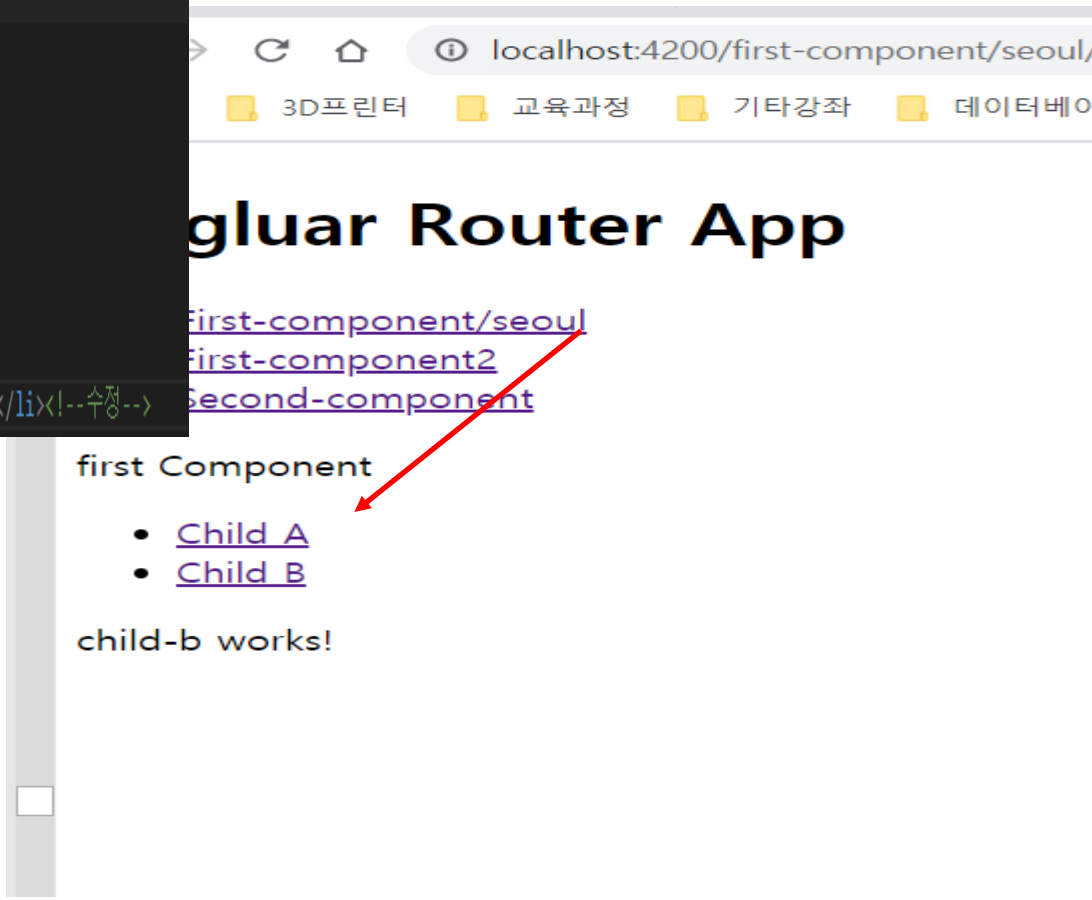
```
1   <h2>First Component</h2>
2
3   <nav>
4     <ul>
5       <li><a routerLink="child-a">Child A</a></li>
6       <li><a routerLink="child-b">Child B</a></li>
7     </ul>
8   </nav>
9
10  <router-outlet></router-outlet>
```

❑ 테스트

app.component.html | app.module.ts | app-routing.module.ts | first.component.html

src > app > app.component.html > nav > ul

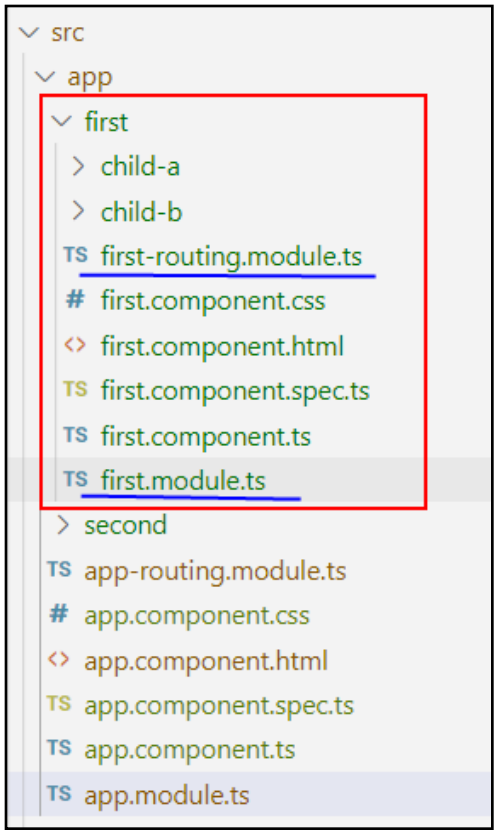
```
1  <h1>{{title}}</h1>
2  <!-- nav태그를 이용한 네비 링크 -->
3  <nav>
4    <ul>
5      <li><a routerLink='' routerLinkActive="active">''</a></li>
6      <li><a routerLink='/home' routerLinkActive="active">/home</a></li>
7      <li><a routerLink="/first-component" routerLinkActive="active">First Component</a></li><!--수정-->
```



❑ 8) sub 모듈

1) sub링크 되는 각 컴포넌트들을 모듈로 관리 가능

```
C:\angular\routing-app>ng g module first/first --routing --flat
CREATE src/app/first/first-routing.module.ts (249 bytes)
CREATE src/app/first/first.module.ts (276 bytes)
```



❑ 8) sub 모듈

2) first 라우팅 모듈에 Routes 설정

App.routing.module.ts에서 주소 잘라내기

app-routing.module.ts

```
outing-app > src > app > TS app-routing.module.ts > ...
1   import { NgModule } from '@angular/core';
2   import { Routes, RouterModule } from '@angular/router';
3
4   import { SecondComponent } from './second/second.component';
5
6   const routes: Routes = [
7     { path: 'second-component', component: SecondComponent },
8   ];
9
10  @NgModule({
11    imports: [RouterModule.forRoot(routes)],
12    exports: [RouterModule]
13  })
14  export class AppRoutingModule { }
```

app-routing.module.ts | first-routing.module.ts | first.component.html

src > app > first > TS first-routing.module.ts > routes > path

```
1   import { NgModule } from '@angular/core';
2   import { Routes, RouterModule } from '@angular/router';
3   import {ChildAComponent} from './child-a/child-a.component';
4   import {ChildBComponent} from './child-b/child-b.component';
5   import {FirstComponent} from './first.component'
6
7   const routes: Routes = [
8     {path:'first-component', component:FirstComponent,
9     children:[
10      {
11        path: 'child-a', //child route path
12        component:ChildAComponent
13      },
14      {
15        path: 'child-b', //child route path
16        component:ChildBComponent
17      }
18    ]
19  }, //컴포넌트 주소등록
20
```

❑ 8) sub 모듈

3) first 모듈에 First관련 컴포넌트 등록

TS first.module.ts ×

src > app > first > TS first.module.ts > ...

```ts
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { FirstRoutingModule } from './first-routing.module';
import {ChildAComponent} from './child-a/child-a.component';
import {ChildBComponent} from './child-b/child-b.component';
import {FirstComponent} from './first.component';
@NgModule({
  declarations: [
    FirstComponent,
    ChildAComponent,
    ChildBComponent
  ],
  imports: [
    CommonModule,
    FirstRoutingModule,
    ]
})
export class FirstModule { }
```

4) first 모듈을 app 모듈에 등록하고 First관련 컴포넌트 삭제

app.module.ts

outing-app > src > app > TS app.module.ts > ...

```
10  import { FirstModule } from './first/first.module';
11
12
13  @NgModule({
14    declarations: [
15      AppComponent,
16      SecondComponent
17    ],
18    imports: [
19      BrowserModule,
20      AppRoutingModule,
21      FirstModule
22    ],
23    providers: [],
24    bootstrap: [AppComponent]
25  })
26  export class AppModule { }
```

❑ App-component.html의 수정

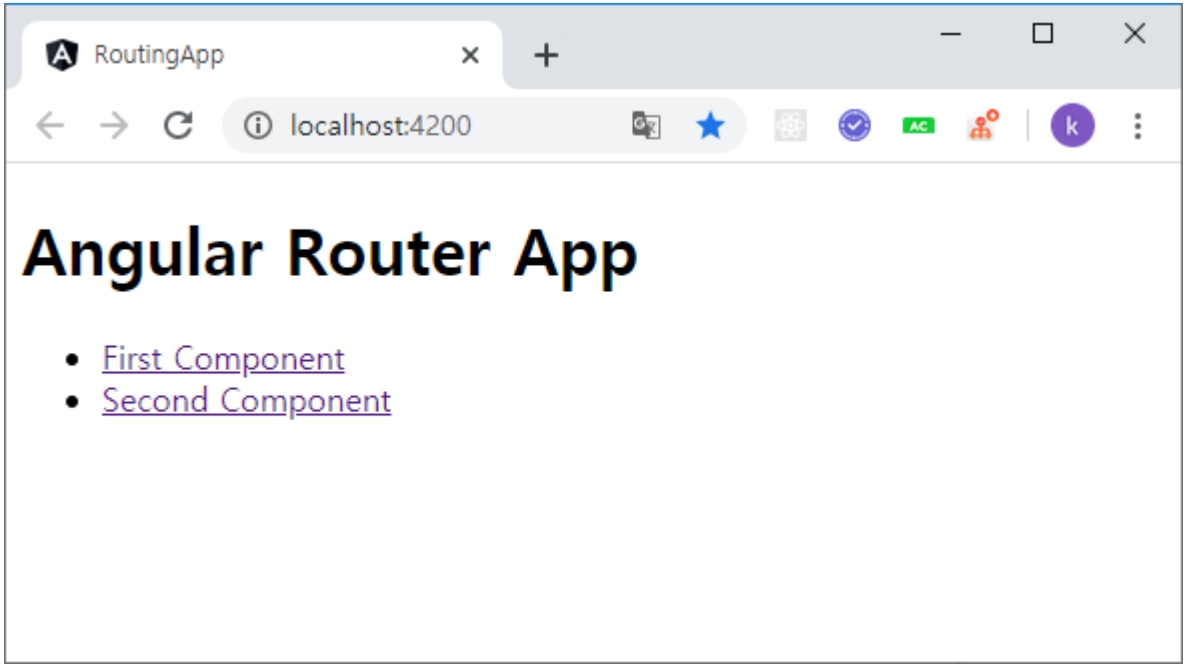
st.component.html | TS app.module.ts | TS app-routing.module.ts | app.component.html ×

app > app.component.html > nav

```
<h1>Angluar Router App</h1>
<nav>
  <!--
    파라미터 전송시url 형태로 :Restful서비스 <==>SOAP
    SOAP방식 :8080/target?key=value?key2=value
    RestFul방식 : 8080/target/key/value/key2/value
    <li><a routerLink="/first-componen/seoul/1000" routerLinkActive="active">First-component/se
    first-compoenent/:city/:pop
  -->
  <ul>
    <li><a routerLink="/first-component" routerLinkActive="active">First-component</a></li>
   <li><a routerLink="/second-component" rotuerLinkActive="active">Second-component</a></li>
```

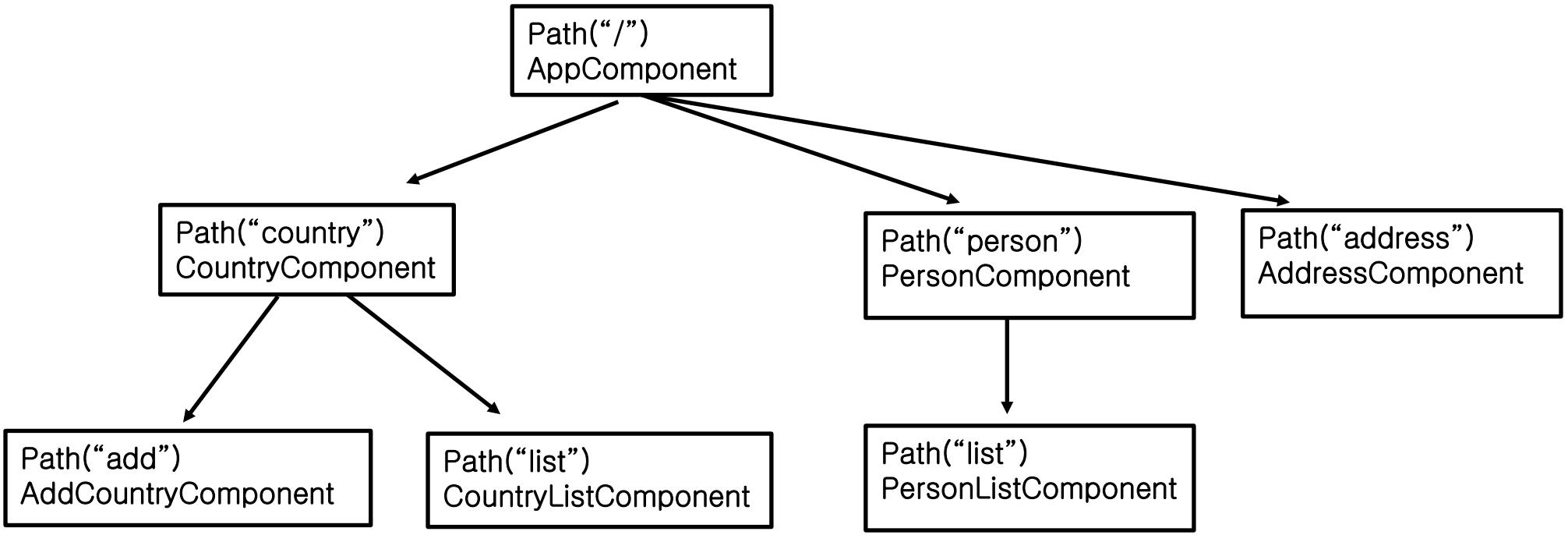
```
PS C:\Angular_Study_chul_2\routing-app> ng serve
```

5) 실행


RoutingApp
localhost:4200
Angular Router App
First Component
Second Component

❑ 실습 문제 1

다음 구조와 일치하는 라우팅 어플리케이션을 구현 하시오.



❑ 실습 문제 1

```
C:\angular\routing-app>ng g component country
? Would you like to share anonymous usage data about this project with the Angular Team at
Google under Google's Privacy Policy at https://policies.google.com/privacy? For more
details and how to change this setting, see http://angular.io/analytics. No
CREATE src/app/country/country.component.html (22 bytes)
CREATE src/app/country/country.component.spec.ts (635 bytes)
CREATE src/app/country/country.component.ts (279 bytes)
CREATE src/app/country/country.component.css (0 bytes)
UPDATE src/app/app.module.ts (479 bytes)

C:\angular\routing-app>ng g component person
CREATE src/app/person/person.component.html (21 bytes)
CREATE src/app/person/person.component.spec.ts (628 bytes)
CREATE src/app/person/person.component.ts (275 bytes)
CREATE src/app/person/person.component.css (0 bytes)
UPDATE src/app/app.module.ts (561 bytes)

C:\angular\routing-app>ng g component address
CREATE src/app/address/address.component.html (22 bytes)
CREATE src/app/address/address.component.spec.ts (635 bytes)
CREATE src/app/address/address.component.ts (279 bytes)
CREATE src/app/address/address.component.css (0 bytes)
UPDATE src/app/app.module.ts (647 bytes)
```

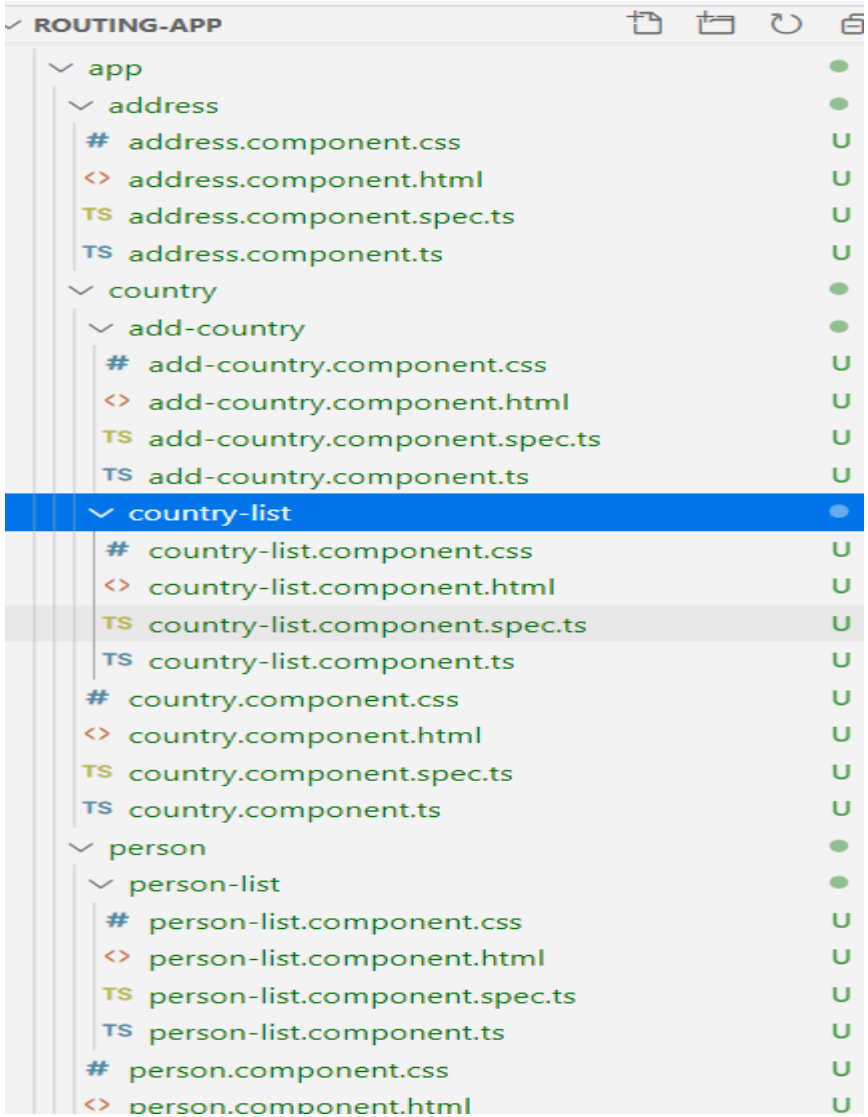
❑ 실습 문제 1

```
C:\angular\routing-app>ng g component country/addCountry
CREATE src/app/country/add-country/add-country.component.html (26 bytes)
CREATE src/app/country/add-country/add-country.component.spec.ts (657 bytes)
CREATE src/app/country/add-country/add-country.component.ts (294 bytes)
CREATE src/app/country/add-country/add-country.component.css (0 bytes)
UPDATE src/app/app.module.ts (833 bytes)

C:\angular\routing-app>ng g component country/countryList
CREATE src/app/country/country-list/country-list.component.html (27 bytes)
CREATE src/app/country/country-list/country-list.component.spec.ts (664 bytes)
CREATE src/app/country/country-list/country-list.component.ts (298 bytes)
CREATE src/app/country/country-list/country-list.component.css (0 bytes)
UPDATE src/app/app.module.ts (945 bytes)

C:\angular\routing-app>ng g component person/personList
CREATE src/app/person/person-list/person-list.component.html (26 bytes)
CREATE src/app/person/person-list/person-list.component.spec.ts (657 bytes)
CREATE src/app/person/person-list/person-list.component.ts (294 bytes)
CREATE src/app/person/person-list/person-list.component.css (0 bytes)
UPDATE src/app/app.module.ts (974 bytes)
```

❑ 실습


ROUTING-APP
app
address
address.component.css
address.component.html
address.component.spec.ts
address.component.ts
country
add-country
add-country.component.css
add-country.component.html
add-country.component.spec.ts
add-country.component.ts
country-list
country-list.component.css
country-list.component.html
country-list.component.spec.ts
country-list.component.ts
country.component.css
country.component.html
country.component.spec.ts
country.component.ts
person
person-list
person-list.component.css
person-list.component.html
person-list.component.spec.ts
person-list.component.ts
person.component.css
person.component.html

❑ 실습

app-routing.module.ts ● | country.component.html | person-list.component.html

src > app > app-routing.module.ts >

```
(alias) class AddressComponent
import AddressComponent
```

```
1   import {
2   import {                                    ngular/router';
3   import {AddressComponent} from './address/address.component';
4   import {CountryComponent} from './country/country.component';
5   import {AddCountryComponent} from './country/add-country/add-country.component';
6   import {CountryListComponent} from './country/country-list/country-list.component';
7   import {PersonListComponent} from './person/person-list/person-list.component'
8   const routes: Routes = [
9         { path:'country',
10      component:CountryComponent,
11      children:[
12       {path:'add',
13       component:AddCountryComponent
14       },
15       {path: 'list',
16       component: CountryListComponent
17       }
18      ]},//end country
19      {path:'person', component:PersonListComponent,
20      children:[
21       {path:'list',
22       component:PersonListComponent
23      } ]
24      },//end Person
25      {path:'address', component:AddressComponent},
26      {path:"**", redirectTo:"/"}
27  ];
28
```

app.component.html

src_11강실습문제1 > app > app.component.html > router-outlet

```
1  <h1>App 컴포넌트</h1>
2  <a routerLink="person">person</a> 
3  <a routerLink="address">address</a> 
4  <a routerLink="country">country</a> 
5  <router-outlet></router-outlet>
```

country.component.html

src_11강실습문제1 > app > country > country.component.html > p

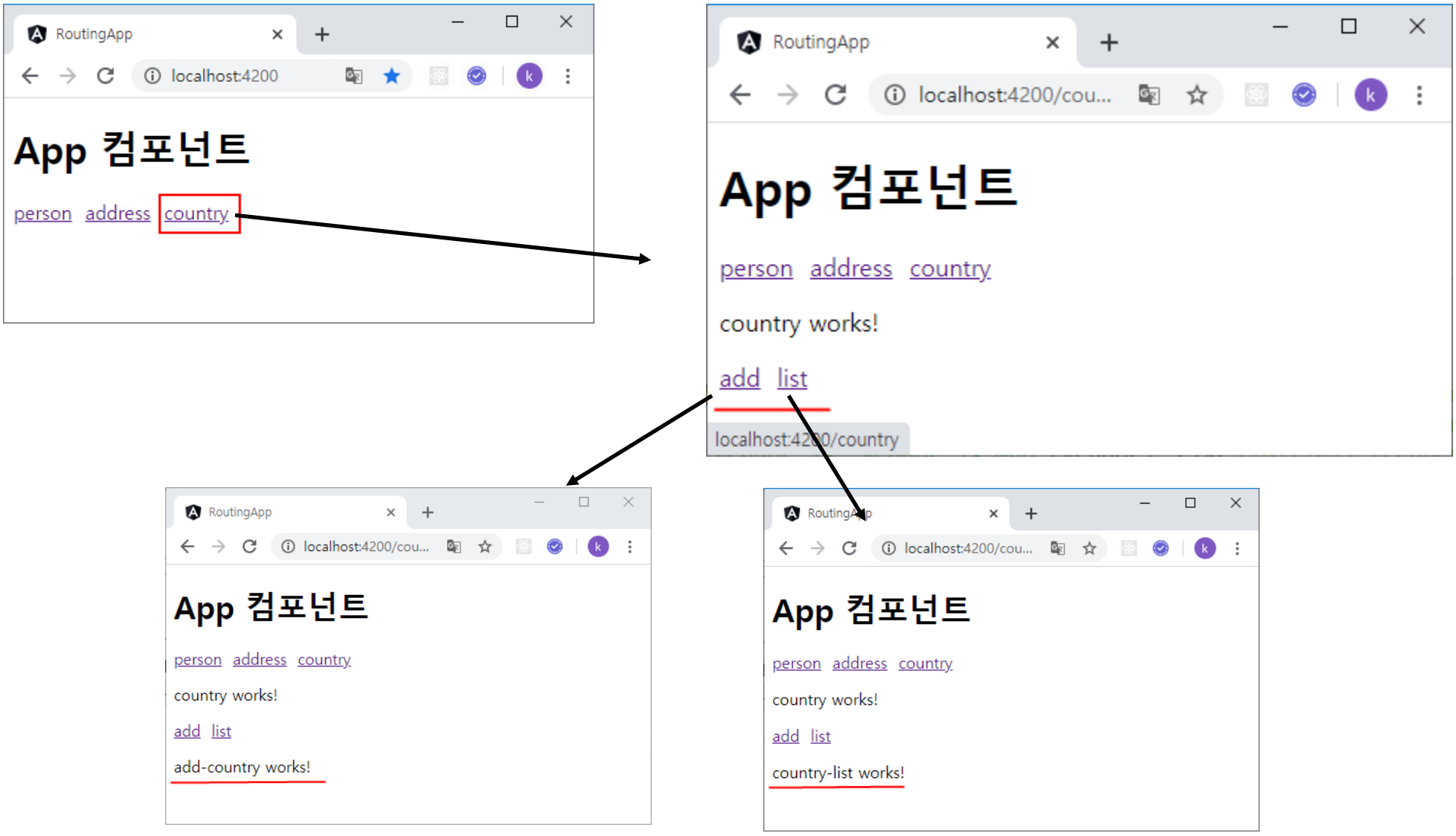
```
1  <p>
2      country works!
3  </p>
4  <a routerLink="add">add</a> 
5  <a routerLink="list">list</a> 
6  <router-outlet></router-outlet>
```

person.component.html

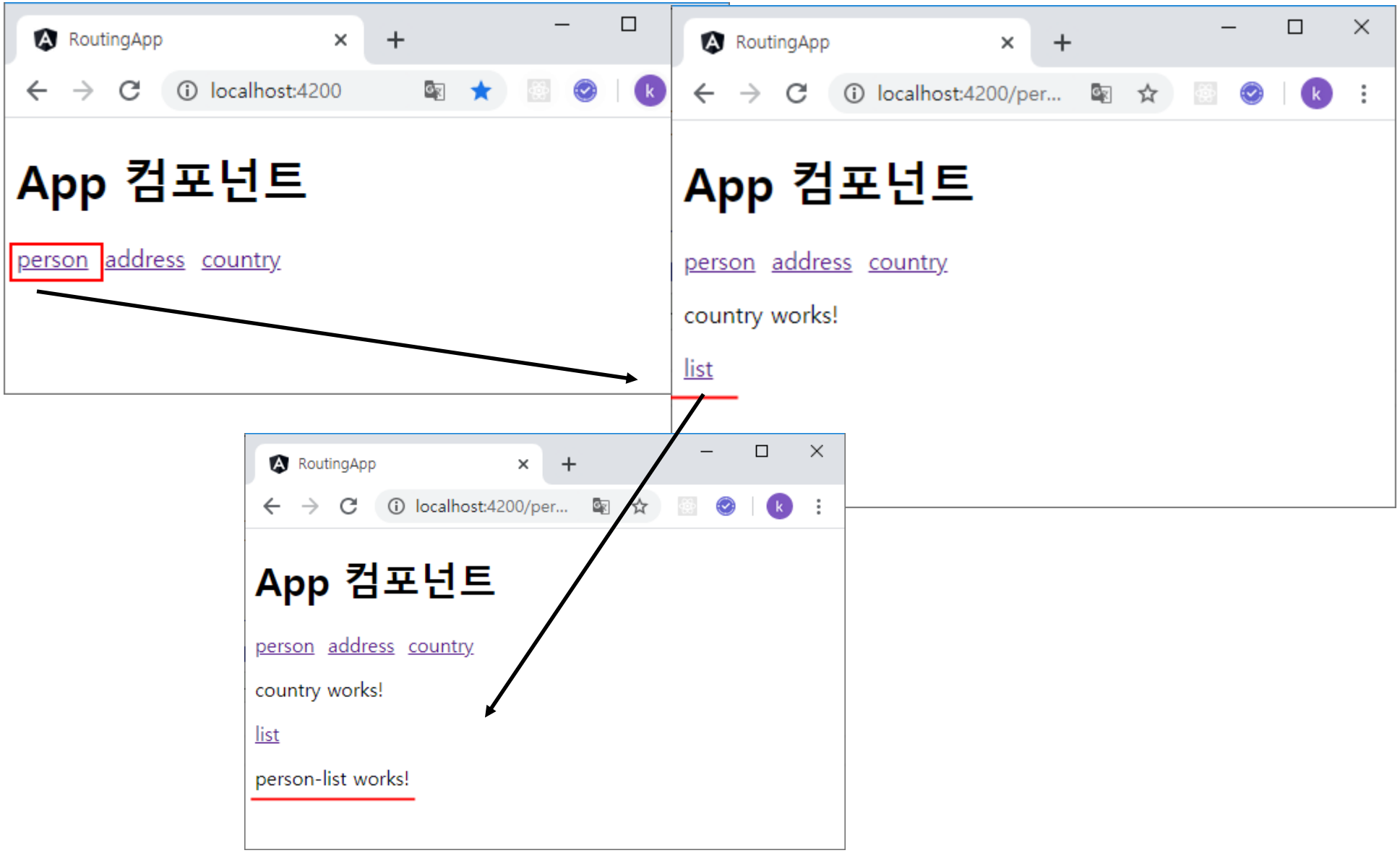
src_11강실습문제1 > app > person > person.component.html > ...

```
1  <p>
2      country works!
3  </p>
4  <a routerLink="list">list</a> 
5  <router-outlet></router-outlet>
6
```

❑ 실습 문제 1


RoutingApp
localhost:4200
App 컴포넌트
person address country
RoutingApp
localhost:4200/cou...
App 컴포넌트
person address country
country works!
add list
localhost:4200/country
RoutingApp
localhost:4200/cou...
App 컴포넌트
person address country
country works!
add list
add-country works!
RoutingApp
localhost:4200/cou...
App 컴포넌트
person address country
country works!
add list
country-list works!

❑ 실습 문제 1


RoutingApp
localhost:4200
App 컴포넌트
person address country
RoutingApp
localhost:4200/per...
App 컴포넌트
person address country
country works!
list
RoutingApp
localhost:4200/per...
App 컴포넌트
person address country
country works!
list
person-list works!

수고하셨습니다.