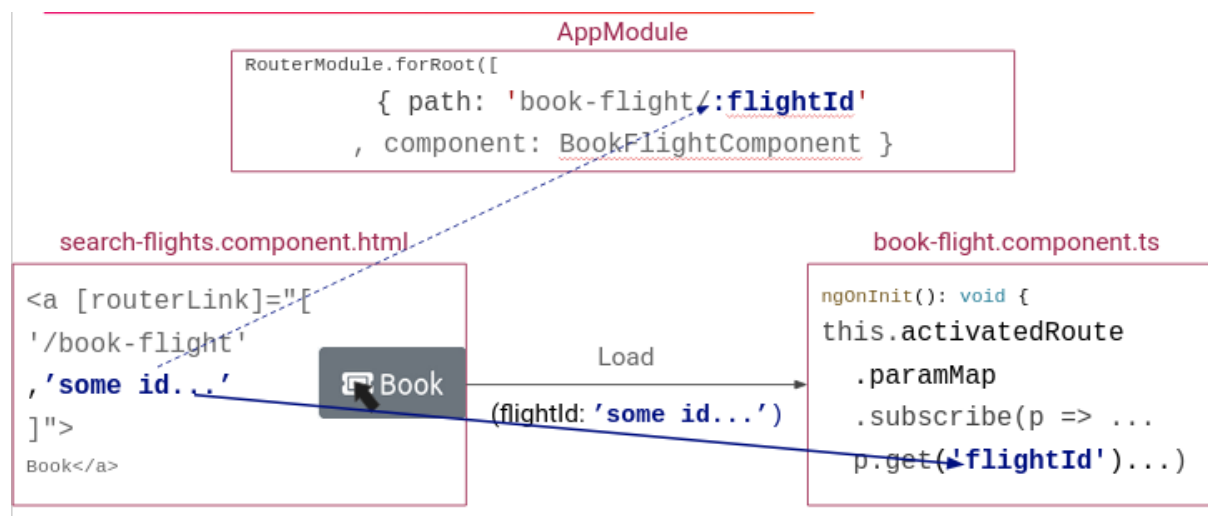


Day 7 - Angular & Asp in 21 days Handouts

Status

Day 7: Angular Components and Routing



Use the Angular CLI to generate a new component:

```
npx ng generate component <component name> --module app
```

To route a component, add the following to the Route section of the module:

```
RouterModule.forRoot([
  ...
  { path: 'component-path', component: MyComponent }
])
```

So by clicking the following action link, the component is loaded to the `router outlet`.

```
<a [routerLink]="['/component-path']"> Load the Component </a>
```

To parameterize a route, add `/:` and then the parameter's name to the end of the path definition.

```
{ path: 'component-path/:parameterName', component: MyComponent }
```

To pass the parameter to the component, add the value to the router link array. You can also bind the value to a property.

```
<a [routerLink]="['/component-path/:parameterName','parameter value']"> Load the Component </a>
```

The component can then read the parameter from the parameters of the activated route.

```
constructor(private route: ActivatedRoute) { }  
ngOnInit(): void {  
    this.route.paramMap  
        .subscribe(p => this.parameterName = p.get('parameterName') ?? 'not passed')  
}  
parameterName: any
```

To pass a route parameter to an action on the Web API you can define the parameter like this:

```
[HttpGet("{paramName}")]  
public ReadModel Find(<parameter type> paramName)
```

To find a single item in a collection, pass it to the `SingleOrDefault` extension method.

```
flights.SingleOrDefault(f => f.Id == "value");
```