

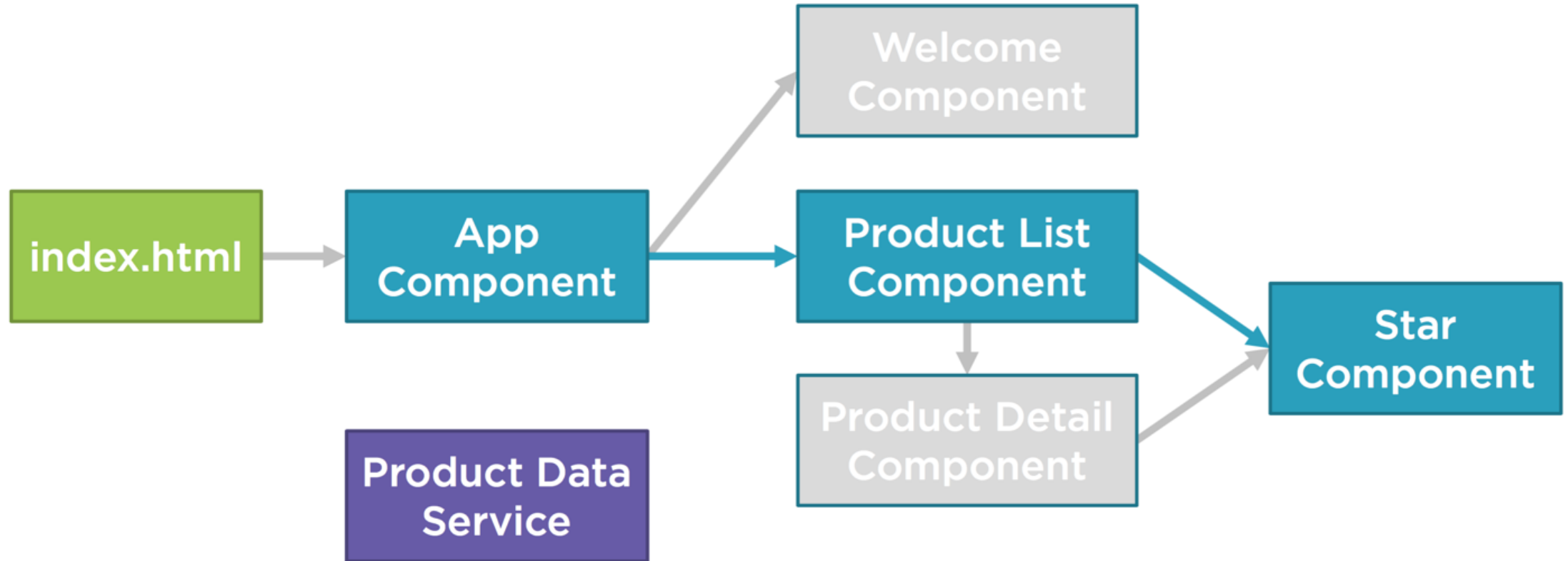
AngularJS

Navigation and Routing Basics

Module Overview

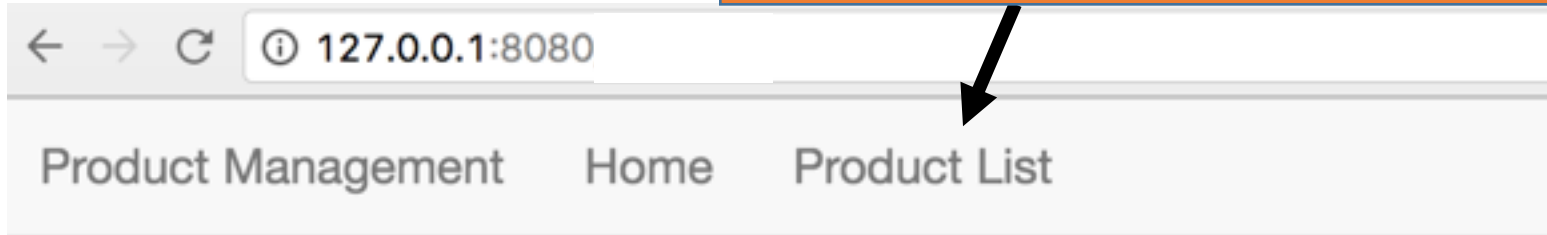
- How does Routing Work?
- Configuring Routes
- Tying routes to actions
- Placing the Views

Application Architecture



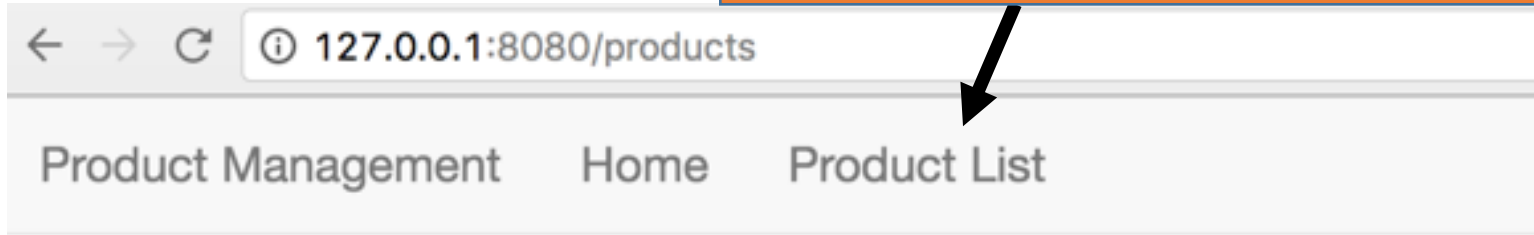
How does Routing Work?

```
<a routerLink="/products">Product List</a>
```



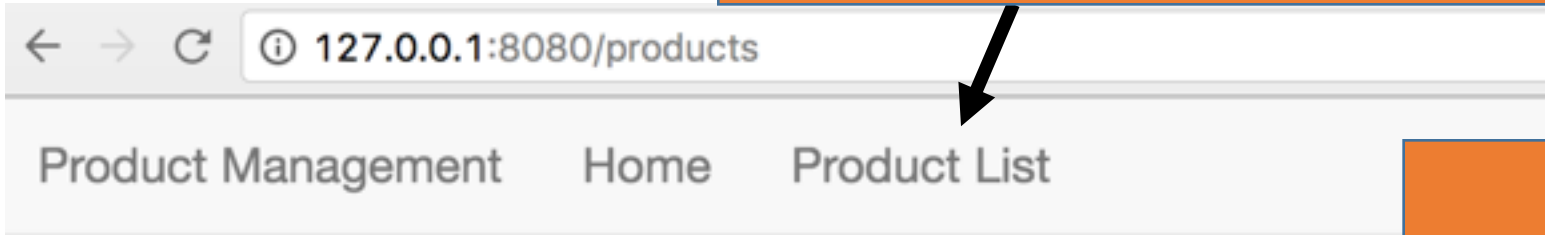
How does Routing Work?

```
<a routerLink="/products">Product List</a>
```



How does Routing Work?

```
<a routerLink="/products">Product List</a>
```



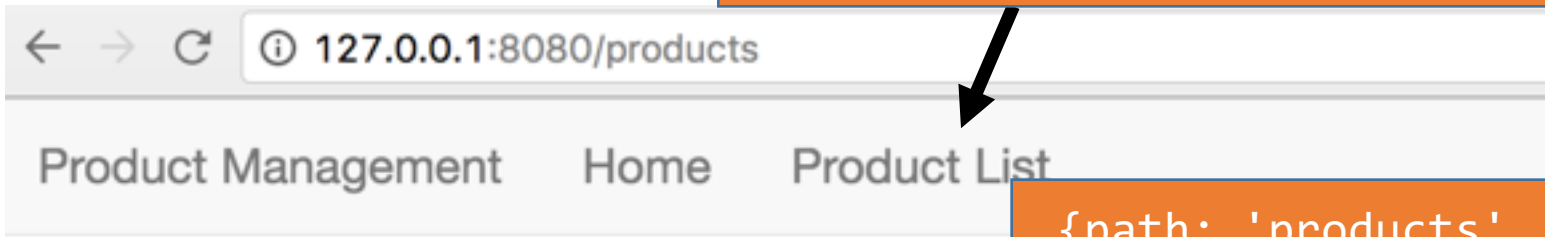
```
http://localhost/products
```

```
http://localhost/#/products
```

By default HTML5 is used (without #).
Webserver needs configuration to redirect all URL to index.html

How does Routing Work?

```
<a routerLink="/products">Product List</a>
```



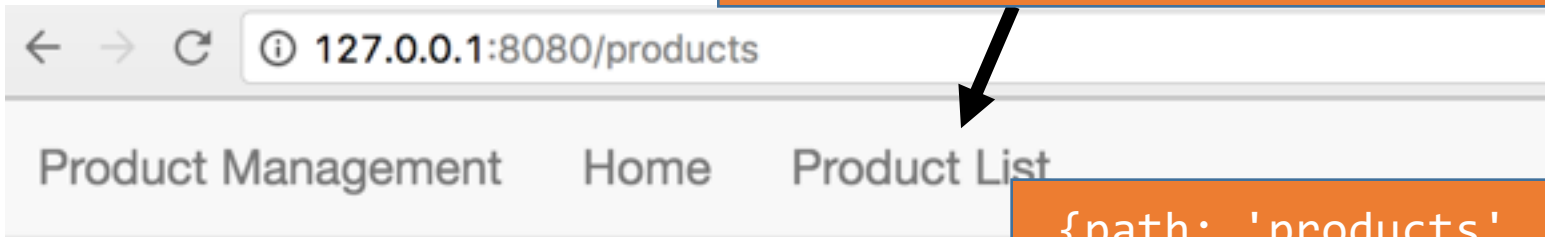
```
{path: 'products', component: ProductListComponent}
```

```
// product-list.components.ts

import { Component } from '@angular/core';
@Component({
  selector: 'nat-products',
  templateUrl: 'app/products/product-
list.component.html'
})
export class ProductListComponent {}
```

How does Routing Work?

```
<a routerLink="/products">Product List</a>
```



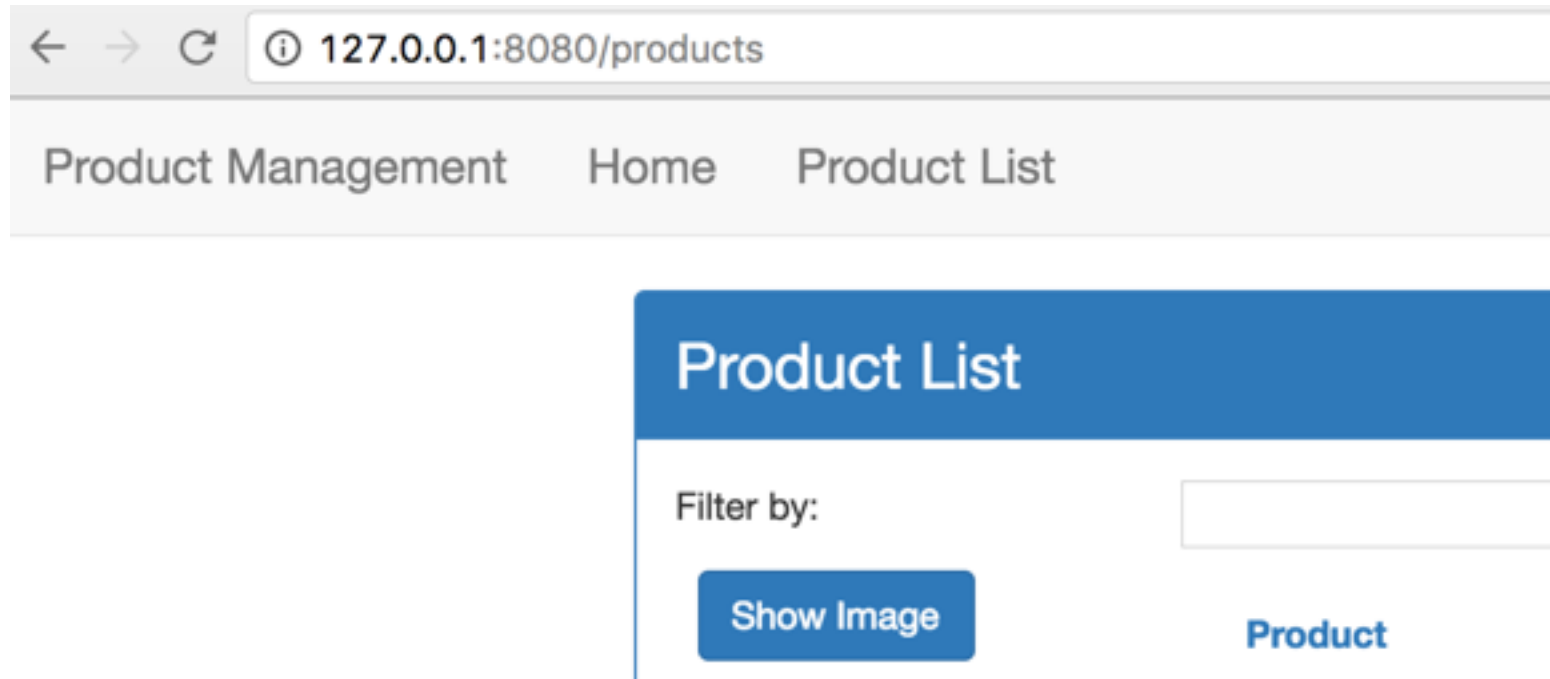
```
{path: 'products', component: ProductListComponent}
```

```
<router-outlet></router-outlet>
```

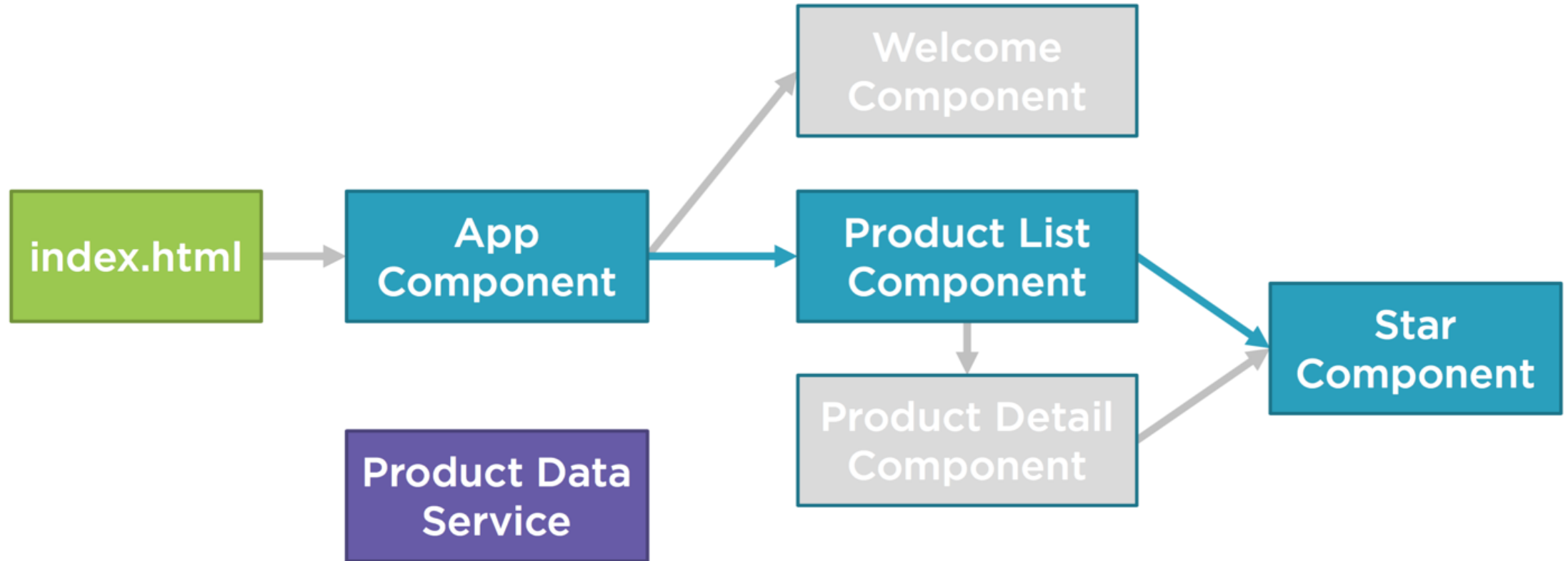
```
// product-list.components.ts

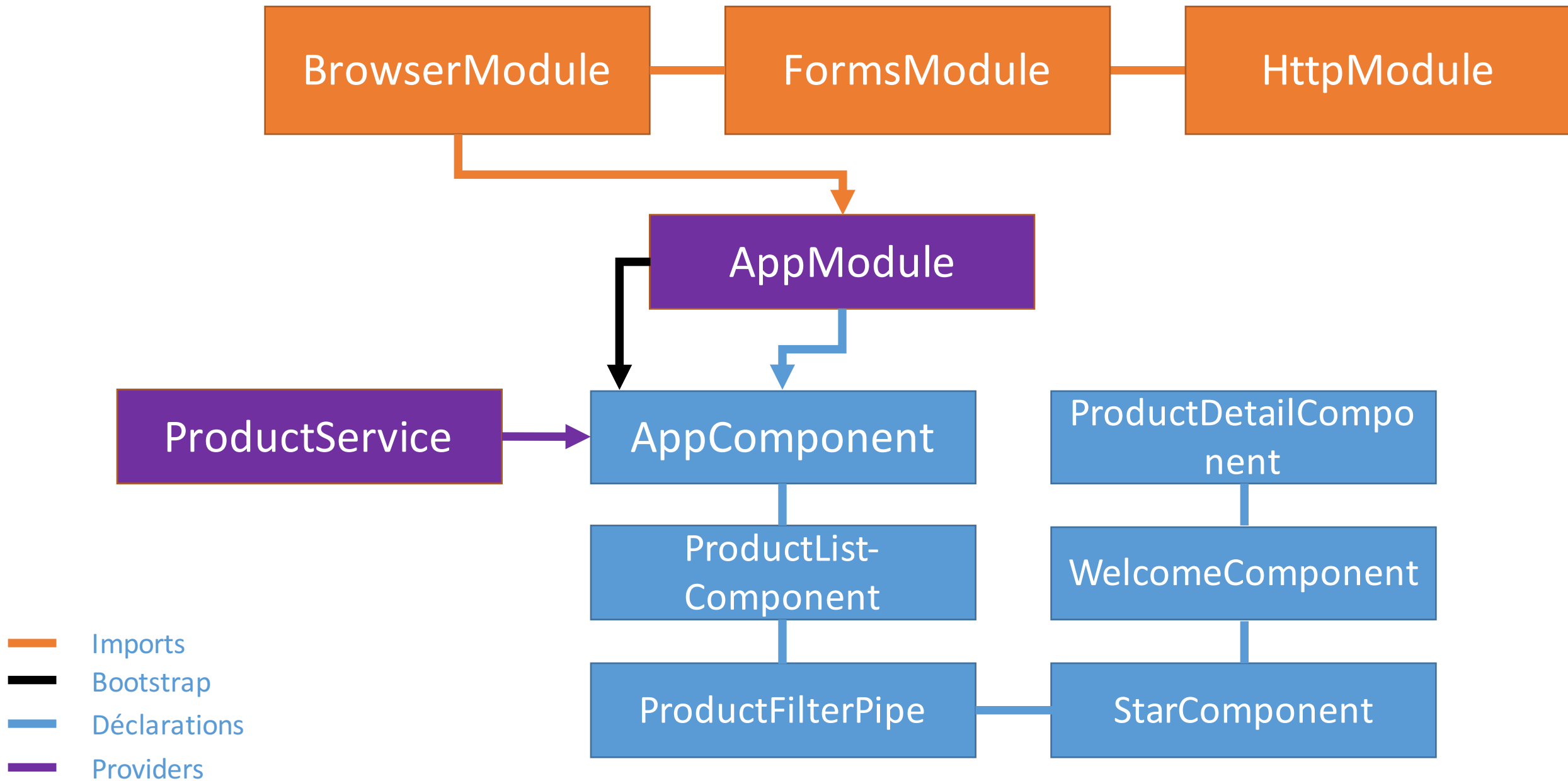
import { Component } from '@angular/core';
@Component({
  templateUrl: 'app/products/product-
list.component.html'
})
export class ProductListComponent {}
```


How does Routing Work?



Application Architecture





Routes into our application

- 'products' → ProductListComponent
- 'product/:id' → ProductDetailComponent
- 'welcome' → WelcomeComponent

Configuring Routes

```
// app.module.ts
import { RouterModule } from 'shared/star.component';
@NgModule({
  imports: [
    BrowserModule,
    FormsModule,
    HttpClientModule,
    RouterModule
  ],
  declarations: [
    ...
  ],
  bootstrap: [ AppComponent ]
})
export class AppModule {}
```

Configuring Routes

```
// app.module.ts
import { RouterModule } from 'shared/star.component';
@NgModule({
  imports: [
    BrowserModule,
    FormsModule,
    HttpClientModule,
    RouterModule.forRoot([])
  ],
  declarations: [
    ...
  ],
  bootstrap: [ AppComponent ]
})
export class AppModule {}
```

Configuring Routes

```
// app.module.ts
import { RouterModule } from 'shared/star.component';
@NgModule({
  imports: [
    BrowserModule,
    FormsModule,
    HttpClientModule,
    RouterModule.forRoot([], {useHash: true})
  ],
  declarations: [
    ...
  ],
  bootstrap: [ AppComponent ]
})
export class AppModule {}
```

Configuring Routes

```
[  
  { path: 'products', component: ProductListComponent },  
  { path: 'product/:id', component: ProductDetailComponent },  
  { path: 'welcome', component: WelcomeComponent },  
  { path: '', redirectTo: 'welcome', pathMatch: 'full' },  
  { path: '**', component: PageNotFoundComponent }  
]
```


Configuring Routes

- No leading slash in the path '/'
- Order matter, it's an array

Navigating the applications routes

- Menu option, link, image or button that activates a route
- Typing the Url in the address bar / bookmark
- The browser's forward or back buttons

Tying Routes to Actions

Product Management Home Product List

Hey

Tying Routes to Actions

```
@Component({  
  selector: 'nat-app',  
  template:`  
    <ul class="nav navbar-nav">  
      <li><a>Home</a></li>  
      <li><a>Product List</a></li>  
    </ul>  
  `,  
})
```

Tying Routes to Actions

```
@Component({
  selector: 'nat-app',
  template: `
    <ul class="nav navbar-nav">
      <li><a [routerLink]="['/welcome']">Home</a></li>
      <li><a [routerLink]="['/products']">Product List</a></li>
    </ul>
  `,
})
```

Placing the Views

```
@Component({
  selector: 'nat-app',
  template: `
    <ul class="nav navbar-nav">
      <li><a [routerLink]="['/welcome']">Home</a></li>
      <li><a [routerLink]="['/products']">Product List</a></li>
    </ul>
    <router-outlet></router-outlet>
  `
})
```

Checklist: Displaying Components

- ✓ Nest-able components
 - ✓ Define a selector
 - ✓ Next in another component
 - ✓ No route
- ✓ Routed components
 - ✓ No selector
 - ✓ Configure routes
 - ✓ Tie routes to actions

Checklist: Doing Routing

- ✓ Configure routes
- ✓ Tie routes to actions
- ✓ Place the view

Checklist: Configuring Routes

- ✓ Define the base element
- ✓ Add RouterModule
 - ✓ Add each route (RouterModule.forRoot)
 - ✓ Order matters
- ✓ path: Url segment for the route
 - ✓ No leading slash
 - ✓ '' for default route
 - ✓ '**' for wildcard route
- ✓ component
 - ✓ not string name, not enclosed in quotes

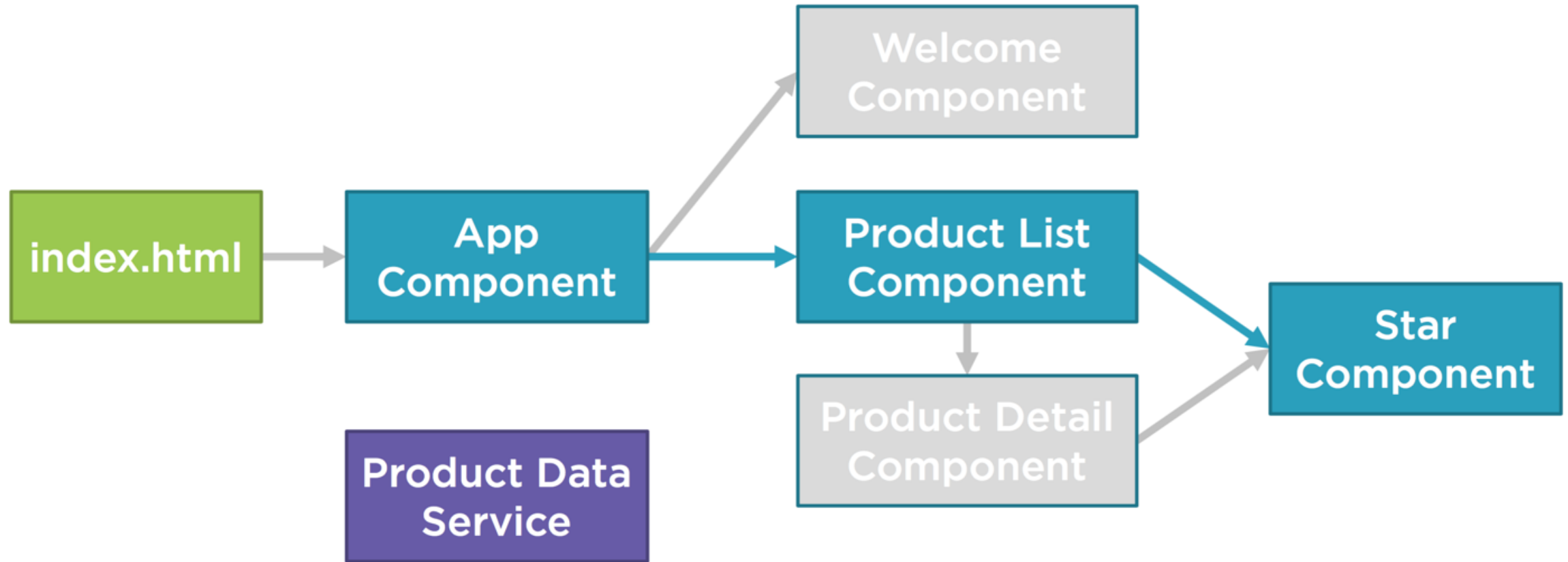
Checklist: Tying Routes to Actions

- ✓ Add the Routerlink Directive as an attribute
 - ✓ Clickable element
 - ✓ Enclose in square brackets
- ✓ Bind to a link parameters array
 - ✓ First element is the path
 - ✓ All other elements are route parameters

Module Overview

- How does Routing Work?
- Configuring Routes
- Tying routes to actions
- Placing the Views

Application Architecture



Application Architecture

