

Navigation and Routing Basics



Deborah Kurata

Consultant | Speaker | Author | MVP | GDE

@deborahkurata



Module Overview



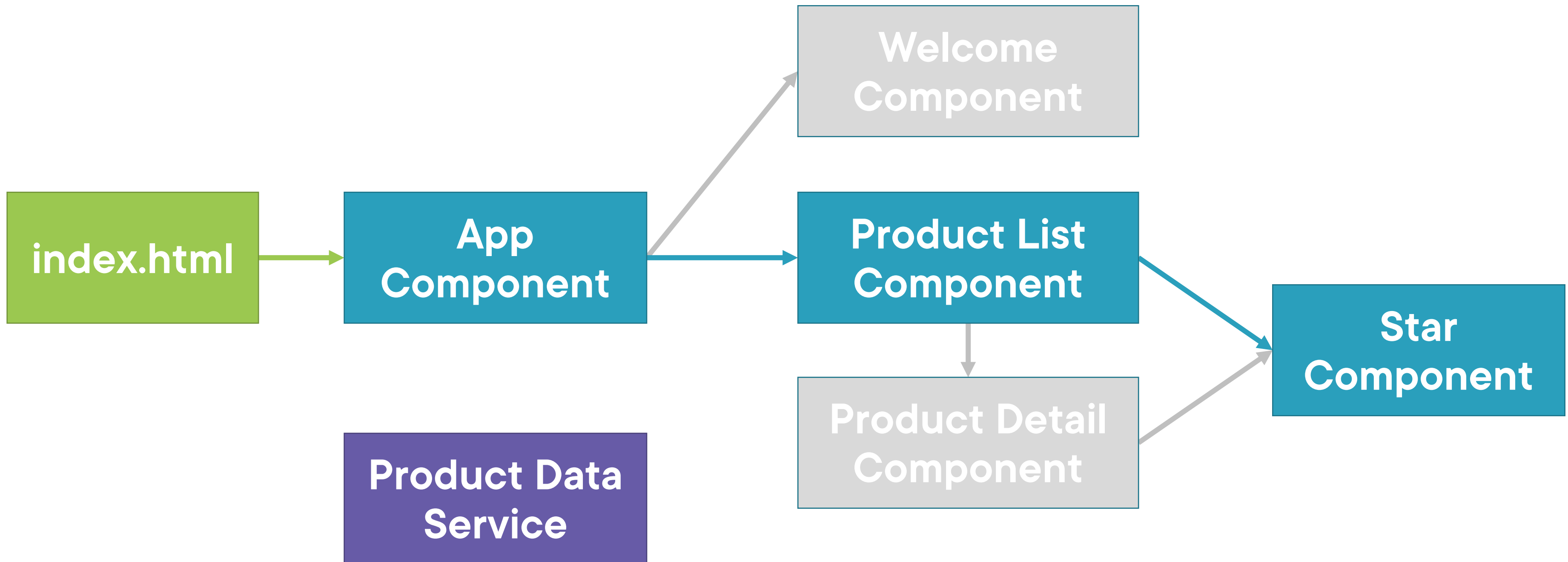
How does routing work?

Configuring routes

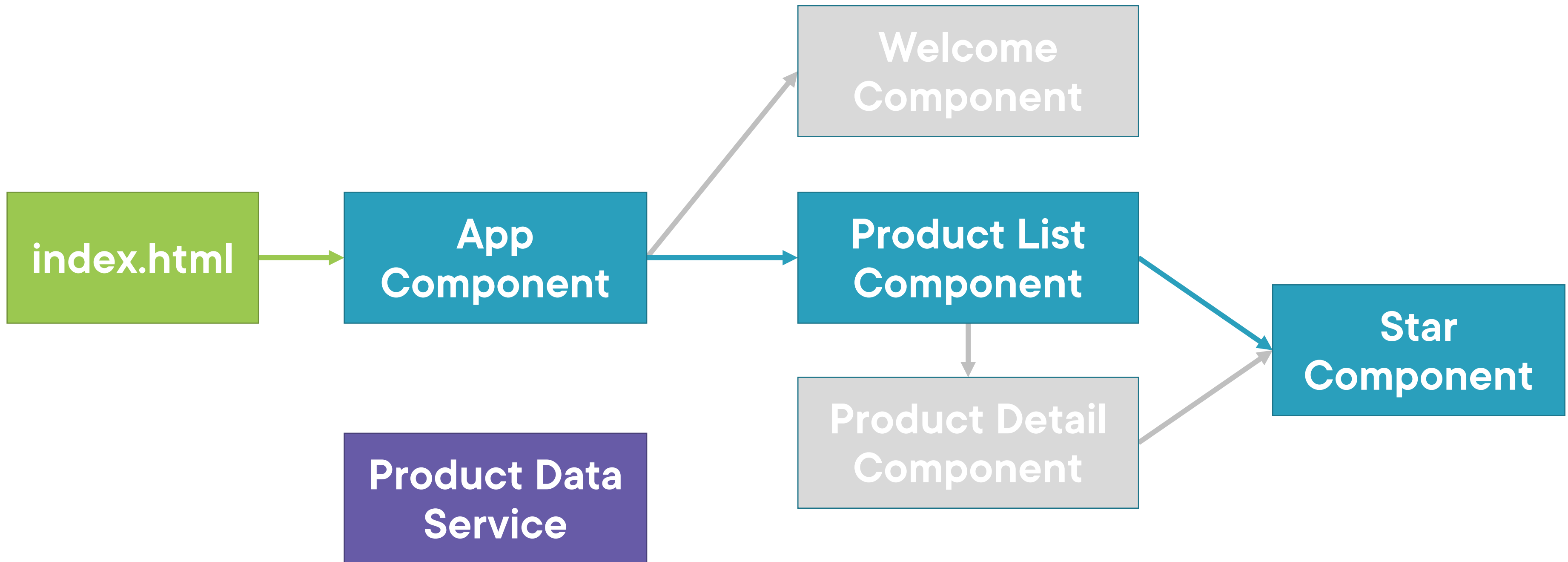
Tying routes to actions

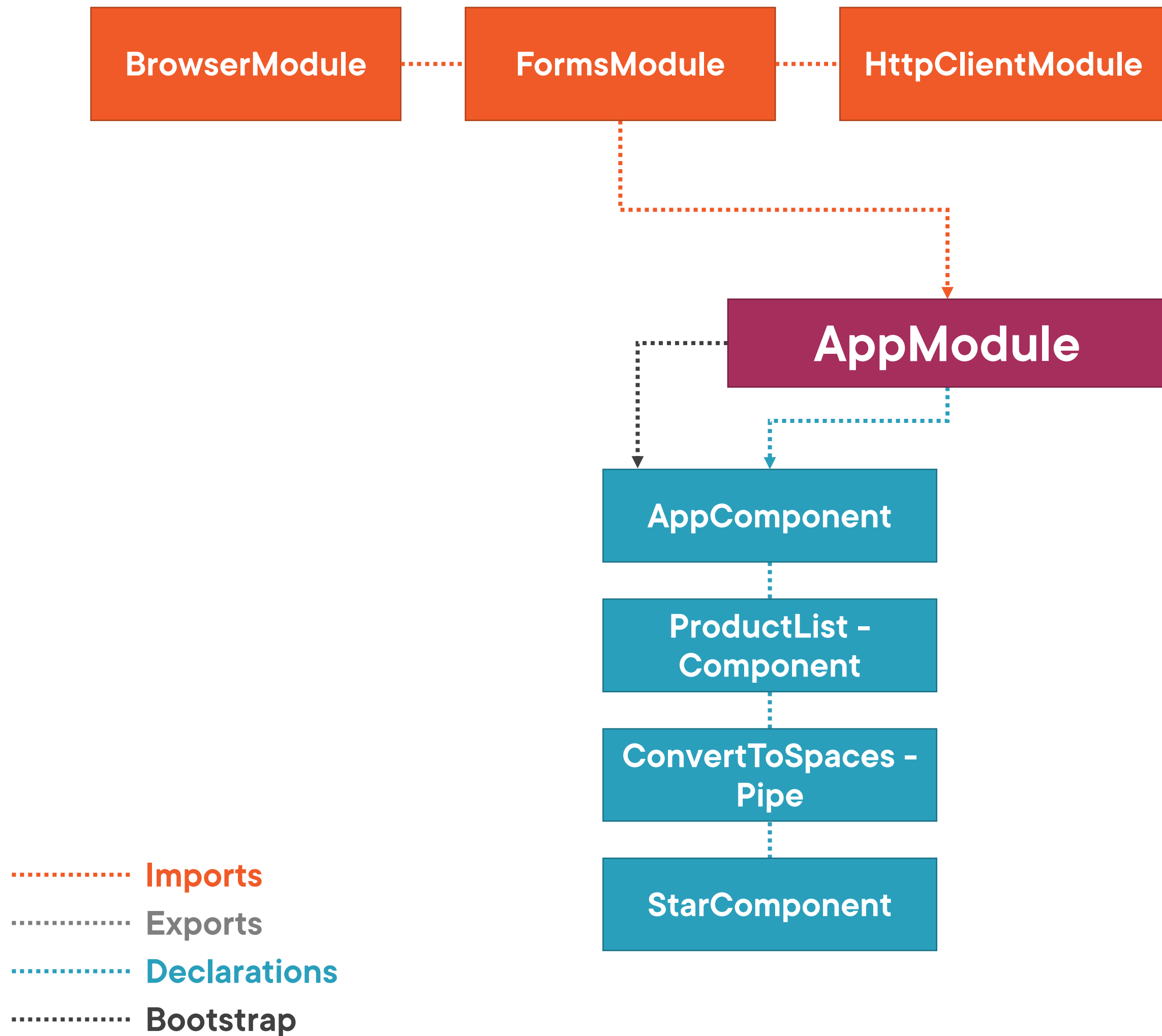
Placing the views

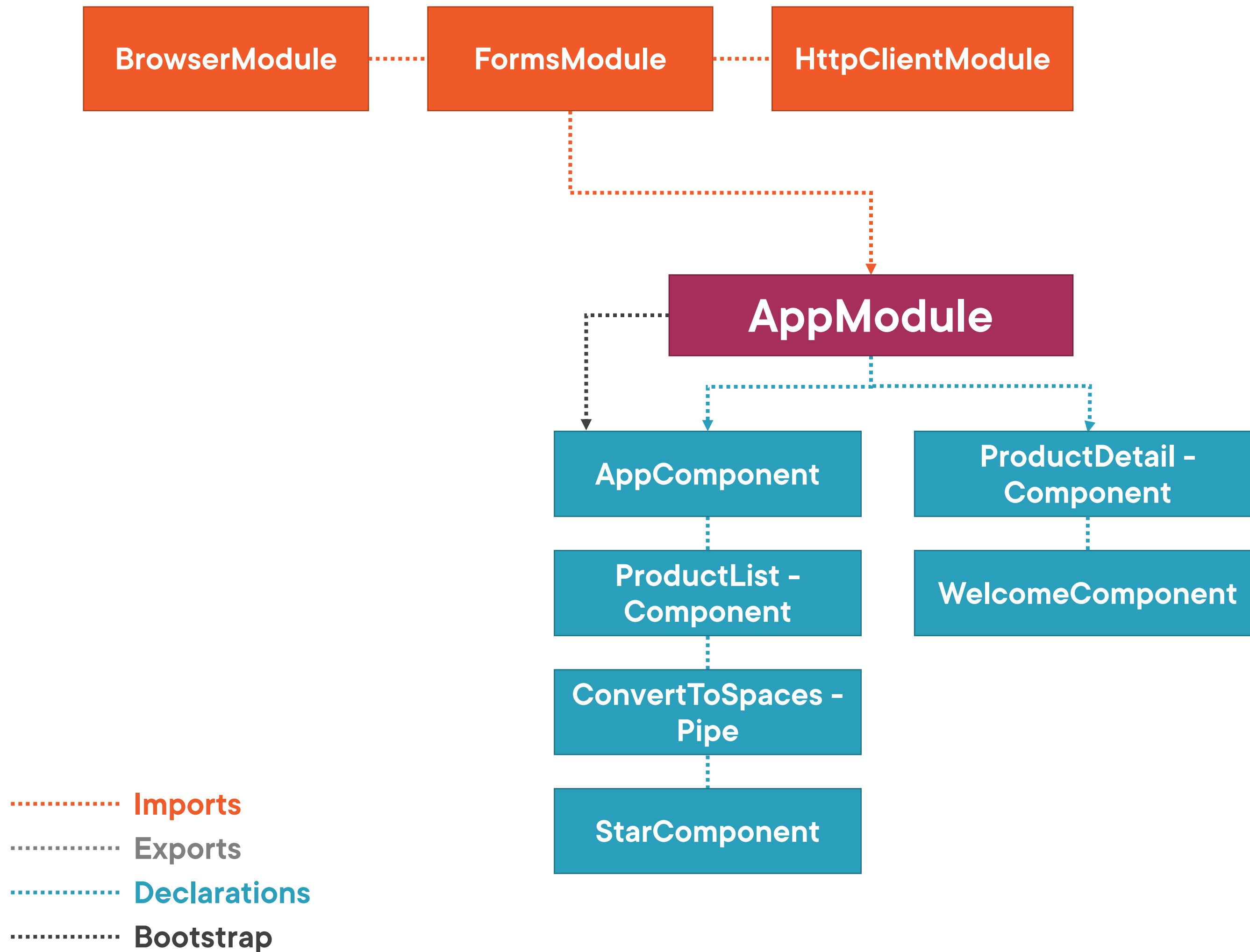
Application Architecture



Application Architecture







How Routing Works

```
▼<pm-root ng-version="11.2.0">
  ▶<nav class="navbar navbar-expand navbar-light bg-light">...</nav>
  <router-outlet></router-outlet>
  ▼<ng-component _ngghost-jvs-c47>
    ▼<div _ngcontent-jvs-c47 class="card">
      <div _ngcontent-jvs-c47 class="card-header"> Product List </div>
      ▼<div _ngcontent-jvs-c47 class="card-body">
        ▶<div _ngcontent-jvs-c47 class="row">...</div>
        ▶<div _ngcontent-jvs-c47 class="row">...</div>
        ▶<div _ngcontent-jvs-c47 class="table-responsive">...</div>
      </div>
    </div>
  </ng-component>
  <!--container-->
</pm-root>
```

Configure a route for each component

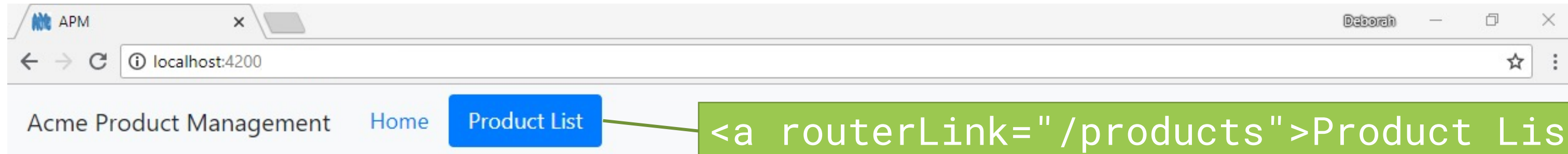
Define options/actions

Tie a route to each option/action

Activate the route based on user action

Activating a route displays the component's view

How Routing Works



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

product-list.component.ts

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

@Component({
  templateUrl: './product-list.component.html'
})
export class ProductListComponent { }
```

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




'products', ProductListComponent

'products/:id', ProductDetailComponent

'welcome', WelcomeComponent

Configuring Routes

app.module.ts

```
...
import { RouterModule } from '@angular/router';

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

Configuring Routes

app.module.ts

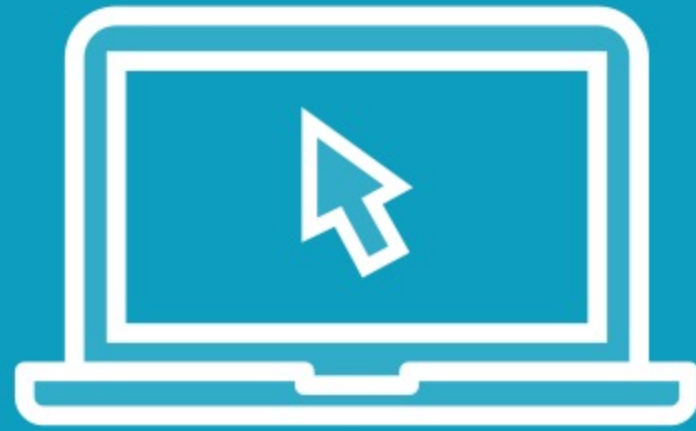
```
...
import { RouterModule } from '@angular/router';

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


Configuring Routes

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

Demo



Configuring routes

Navigating the Application Routes

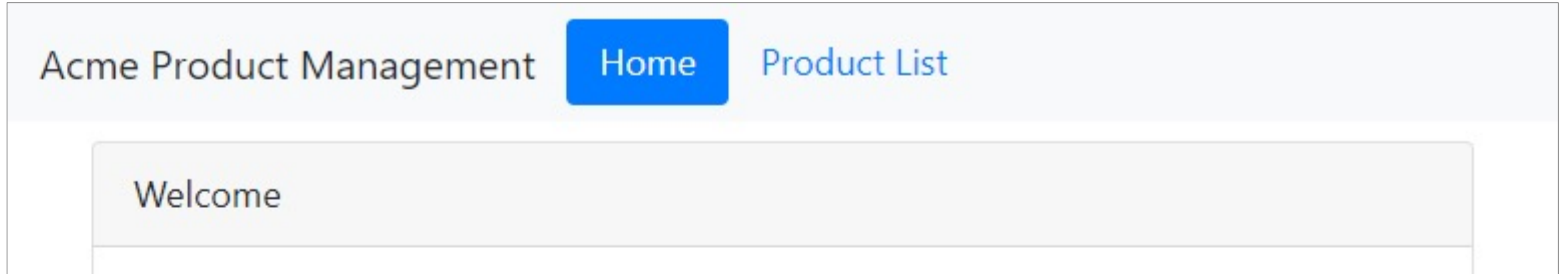


Menu option, link, image or button that activates a route

Typing the Url in the address bar / bookmark

The browser's forward or back buttons

Tying Routes to Actions



Tying Routes to Actions

app.component.ts

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

Tying Routes to Actions

app.component.ts

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

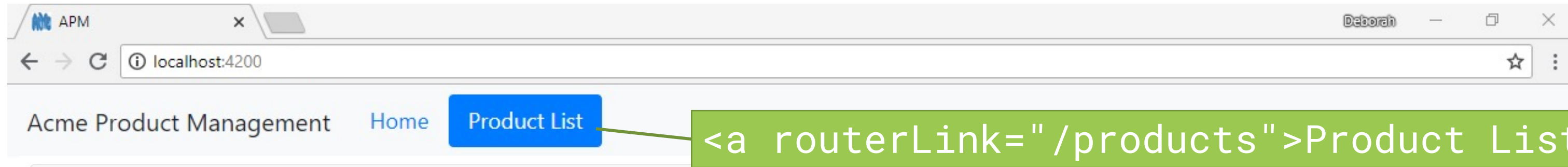
```
<li><a routerLink="/welcome">Home</a></li>
<li><a routerLink="/products">Product List</a></li>
```


Placing the Views

app.component.ts

```
@Component({
  selector: 'pm-root',
  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>
  `
})
```

How Routing Works



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

product-list.component.ts

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

@Component({
  templateUrl: './product-list.component.html'
})
export class ProductListComponent { }
```

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

Routing Checklist: Displaying Components



Nest-able components

- Define a selector

```
@Component({  
  selector: 'pm-products',  
  templateUrl: './product-list.component.html'  
})
```

- Nest in another component

```
<div><h1>{{pageTitle}}</h1>  
  <pm-products></pm-products>  
</div>
```

- No route

Routed components

- No selector or nesting

- Configure routes

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

- Tie routes to actions

Routing Checklist: Doing Routing



Configure routes

Tie routes to actions

Place the view

Routing Checklist:

Configuring Routes



Define the base element

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

Add RouterModule

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

Routing Checklist: Configuring Routes



Add each route to forRoot array

- 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

```
RouterModule.forRoot([
  { path: 'products', component: ProductListComponent }, ...
  { path: '', redirectTo: 'welcome', pathMatch: 'full' },
  { path: '**', redirectTo: 'welcome', pathMatch: 'full' }
])
```


Routing 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

```
<ul>
  <li><a [routerLink]="[' /welcome']">Home</a></li>
  <li><a [routerLink]="[' /products']">Product List</a><
/ul>
```

Routing Checklist:

Placing the View

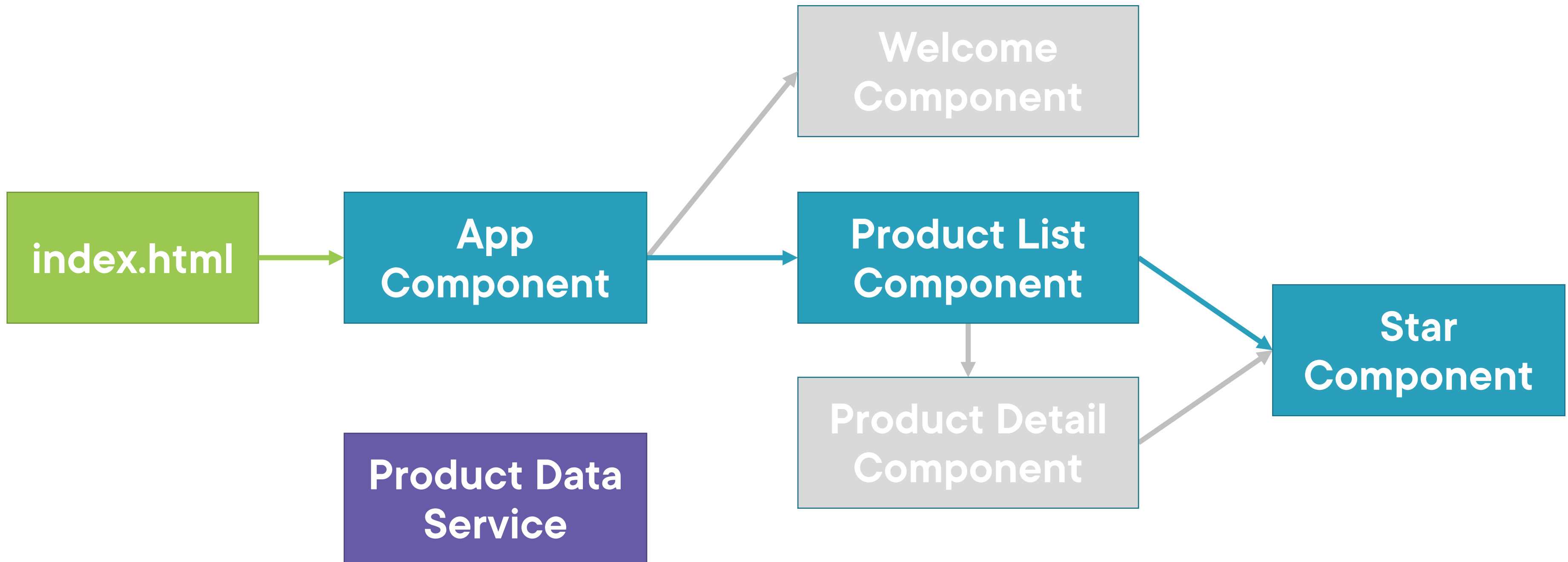


Add the RouterOutlet directive

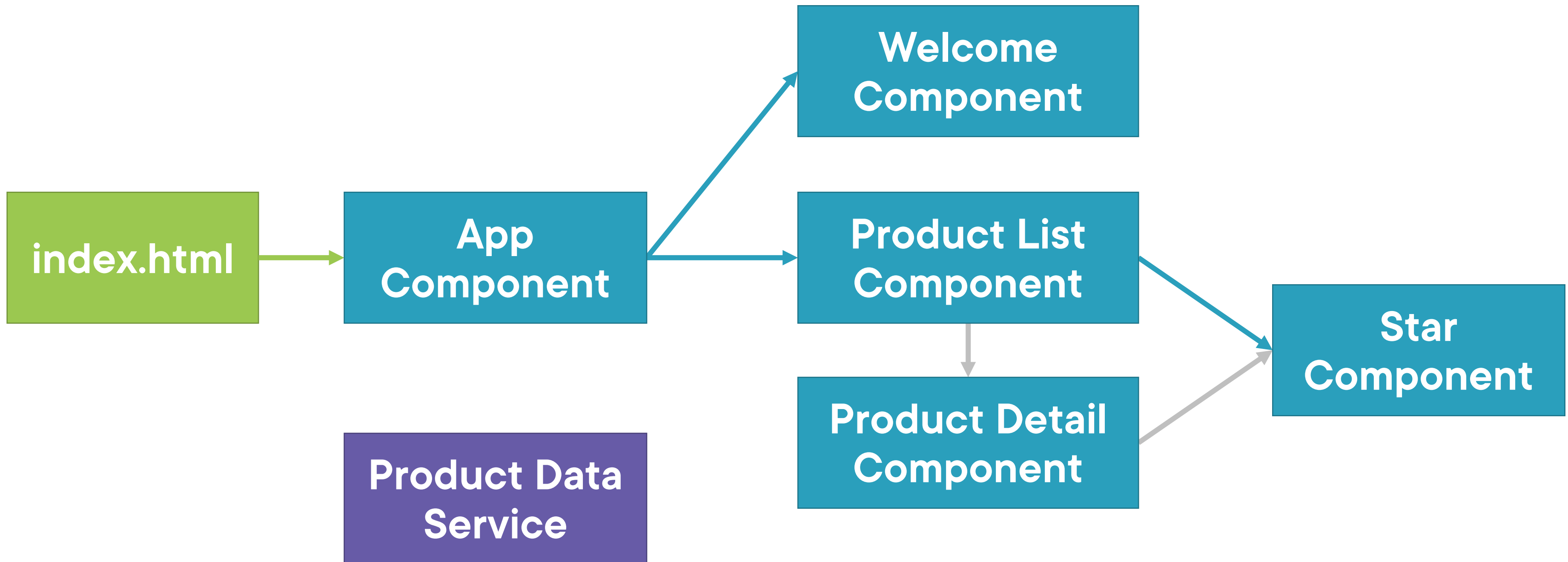
- Identifies where to display the routed component's view
- Specified in the host component's template

```
<ul>
  <li><a [routerLink]="[' /welcome' ]">Home</a></li>
  <li><a [routerLink]="[' /products' ]">Product List</a><
/ul>
</ul>
<router-outlet></router-outlet>
```

Application Architecture



Application Architecture





Coming up next ...

**Navigation and Routing
Additional Techniques**