

Navigation and Routing

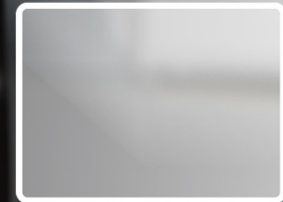
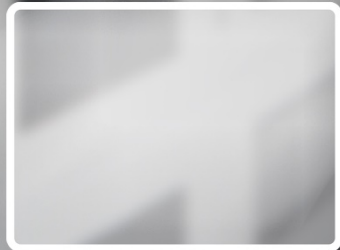


Deborah Kurata

CONSULTANT | SPEAKER | AUTHOR

@deborahkurata | blogs.msmvps.com/deborahk/





Routing

```
▼ <pm-app>
  ▼ <div>
    <h1>Acme Product Management</h1>
    ▼ <pm-products>
      ▼ <div class="panel panel-primary">
        <div class="panel-heading">
          Product List
        </div>
        ▼ <div class="panel-body">
          ::before
          ▶ <div class="row">...</div>
          ▶ <div class="row">...</div>
          ▼ <div class="table-responsive">
            <!--template bindings={}-->
            ▼ <table class="table">
              ▶ <thead>...</thead>
              ▶ <tbody>...</tbody>
            </table>
          </div>
          ::after
        </div>
      </div>
    </pm-products>
  </div>
</pm-app>
```

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



Module Overview



Setting up

Configuring Routes

Tying Routes to Actions

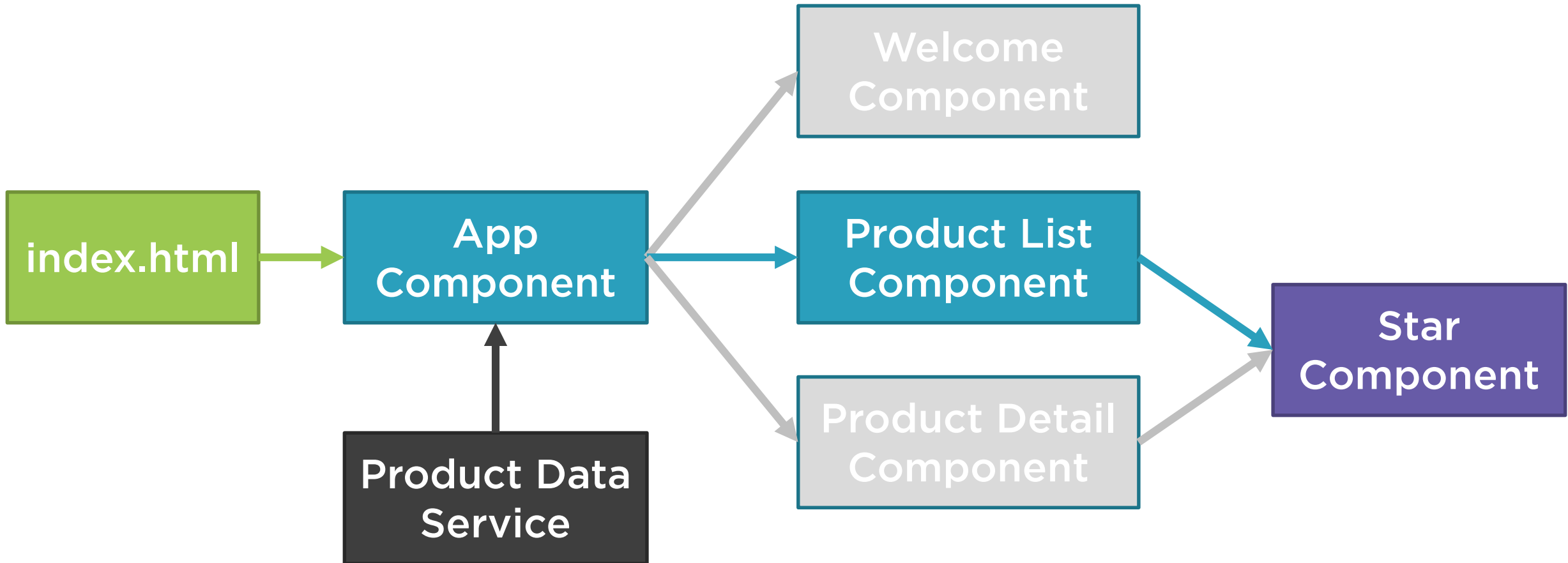
Placing the Views

Passing Parameters to a Route

Activating a Route with Code



Application Architecture



Setting Up



Include the *Angular 2* router script

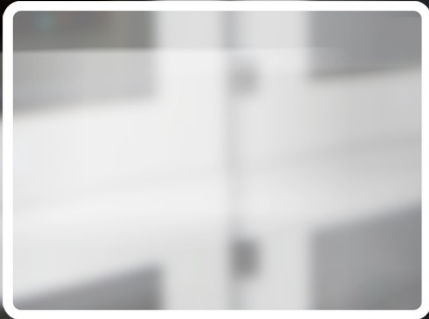
Define the base element

Register ROUTER_PROVIDERS





'Products', '/products', ProductListComponent



'Product', '/product', ProductDetailComponent



'Home', '/welcome', WelcomeComponent



Configuring Routes

app.component.ts (host component)

```
import { RouteConfig } from 'angular2/router';

import { WelcomeComponent } from 'welcome.component';
import { ProductListComponent } from 'product-list.component';

@RouteConfig([
  { path: '/welcome', name: 'Welcome', component: WelcomeComponent,
    useAsDefault: true },
  { path: '/products', name: 'Products', component: ProductListComponent }
])
```



Navigating the Application Routes

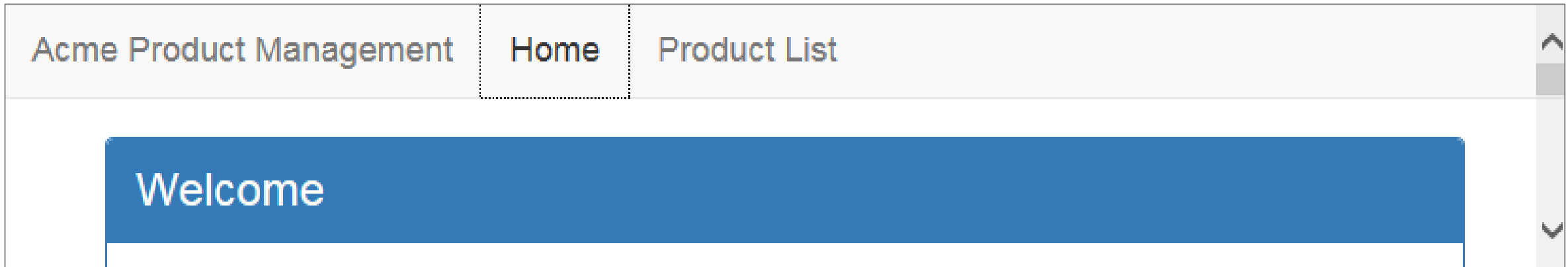


Menu option, link, image or button that activates a route

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

```
import { ROUTER_PROVIDERS } from 'angular2/router';

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



Tying Routes to Actions

app.component.ts

```
import { ROUTER_PROVIDERS, ROUTER_DIRECTIVES } from 'angular2/router';

@Component({
  selector: 'pm-app',
  template: `
    <ul class='nav navbar-nav'>
      <li><a [routerLink]="['Welcome']">Home</a></li>
      <li><a [routerLink]="['Products']">Product List</a></li>
    </ul>
  `,
  directives: [ROUTER_DIRECTIVES],
  providers: [ROUTER_PROVIDERS]
})
```



Placing the Views

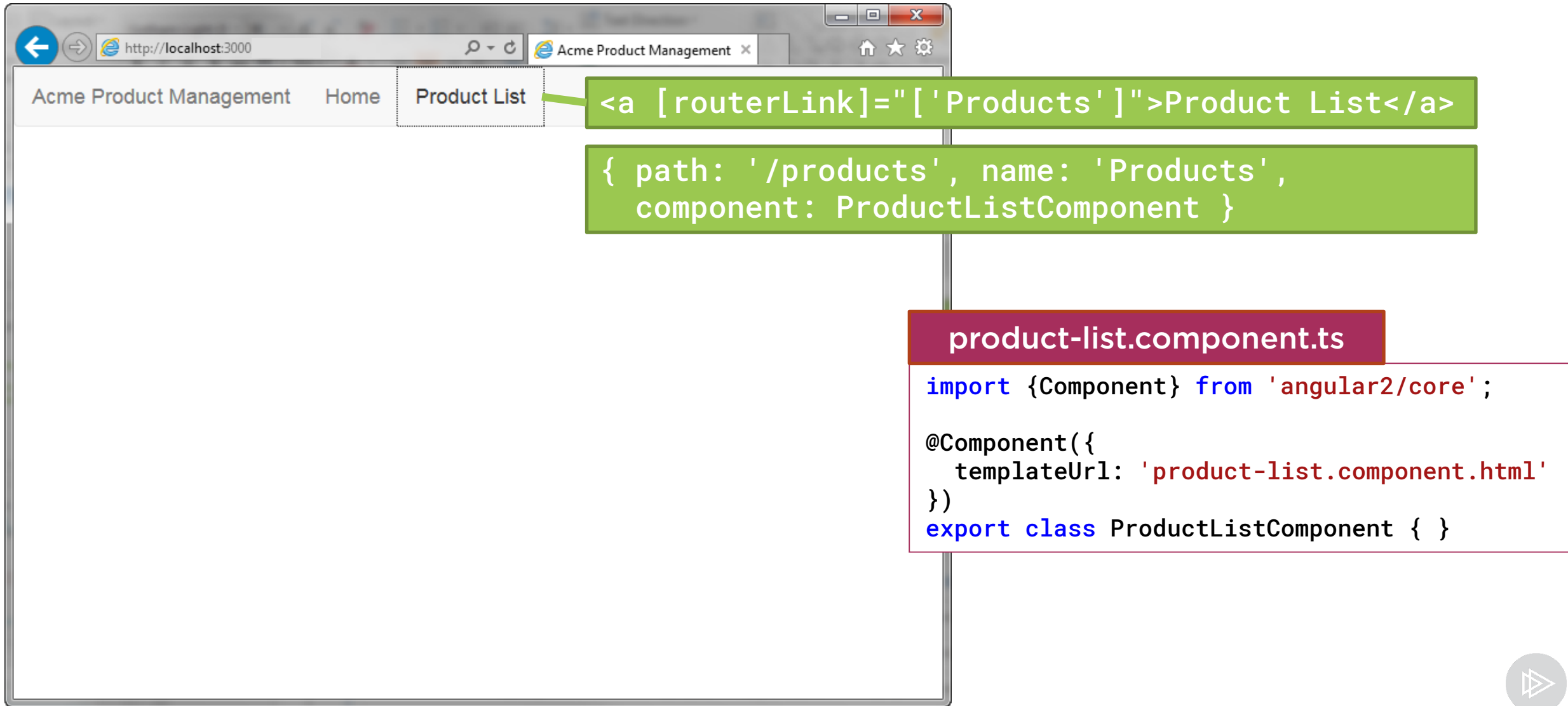
app.component.ts

```
import { ROUTER_PROVIDERS, ROUTER_DIRECTIVES } from 'angular2/router';

@Component({
  selector: 'pm-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>
  `,
  directives: [ROUTER_DIRECTIVES],
  providers: [ROUTER_PROVIDERS]
})
```



How Routing Works



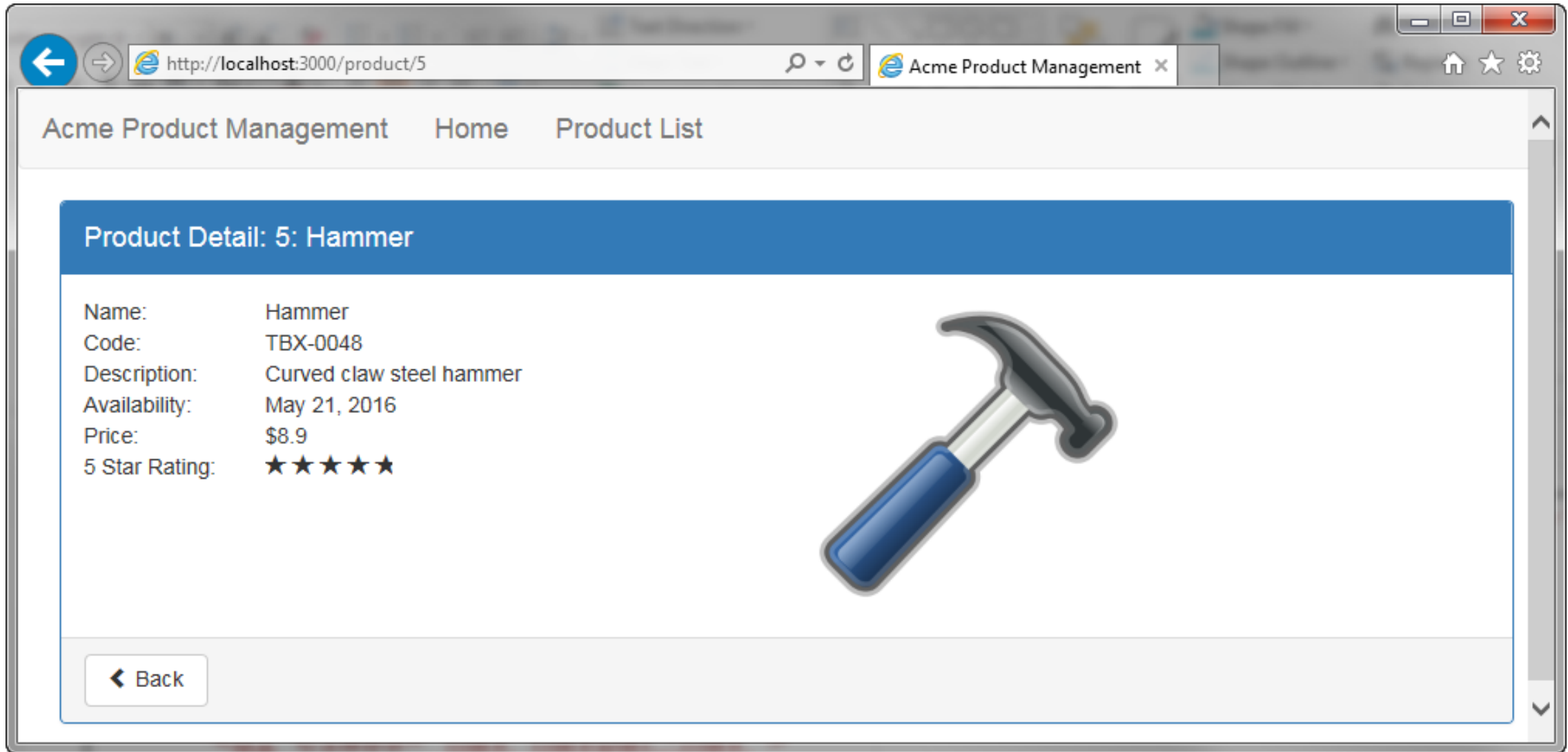
The screenshot shows a web browser at `http://localhost:3000` displaying the 'Acme Product Management' application. The navigation bar contains 'Acme Product Management', 'Home', and 'Product List'. The 'Product List' link is highlighted, and a callout box shows the routerLink attribute used in the link: `<a [routerLink]='['Products']">Product List`. Another callout box shows the routing configuration object: `{ path: '/products', name: 'Products', component: ProductListComponent }`. A third callout box shows the TypeScript code for the `product-list.component.ts` file:

```
import {Component} from 'angular2/core';

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



Passing Parameters to a Route



Passing Parameters to a Route

app.component.ts

```
import { RouteConfig } from 'angular2/router';

import { WelcomeComponent } from 'welcome.component';
import { ProductListComponent } from 'product-list.component';

@RouteConfig([
  { path: '/welcome', name: 'Welcome', component: WelcomeComponent,
    useAsDefault: true },
  { path: '/products', name: 'Products', component: ProductListComponent },
  { path: '/product/:id', name: 'ProductDetail',
    component: ProductDetailComponent }
])
```



Passing Parameters to a Route

product-list.component.html

```
<td>
  <a [routerLink]="['ProductDetail', {id: product.productId}]">
    {{product.productName}}
  </a>
</td>
```

app.component.ts

```
{ path: '/product/:id', name: 'ProductDetail',
                                component: ProductDetailComponent }
```



Passing Parameters to a Route

product-detail.component.ts

```
import { RouteParams } from 'angular2/router';

constructor(private _routeParams: RouteParams) {
  console.log(this._routeParams.get('id'));
}
```

app.component.ts

```
{ path: '/product/:id', name: 'ProductDetail',
  component: ProductDetailComponent }
```



Activating a Route with Code

product-detail.component.ts

```
import { Router } from 'angular2/router';

constructor(private _router: Router) { }

onBack(): void {
  this._router.navigate(['Products']);
}
```



Checklist: Displaying Components



Nest-able Components

- Define a selector
- Nest in another component
- No route

Routed Components

- No selector
- Configured route
- User actions to activate the route

Routing Checklist: Setup







Include the Angular 2 router script

Define the base element

Register ROUTER_PROVIDERS



Routing Checklist: Configuring Routes



path

- Url path segment for route

name

- Route name
- PascalCase

component

- Not string name; not enclosed in quotes
- Each component must be imported

useAsDefault



Routing Checklist: Tying Routes to Actions



Specify ROUTER_DIRECTIVES in the directives array

Add the RouterLink directive as an attribute

- Clickable element
- Enclose in square brackets

Bind to a link parameters array

- First element is the route name
- All other elements are route parameters



Routing Checklist: Place the View







RouterOutlet

Identifies where to display the routed component's view

Specified in the host component's template



Routing Checklist: Passing Parameters

app.component.ts

```
{ path: '/product/:id', name: 'ProductDetail',  
  component: ProductDetailComponent }
```

product-list.component.html

```
<a [routerLink]="['ProductDetail', {id: product.productId}]">  
  {{product.productName}}  
</a>
```

product-detail.component.ts

```
import { RouteParams } from 'angular2/router';  
  
constructor(private _routeParams: RouteParams) {  
  console.log(this._routeParams.get('id'));  
}
```

Routing Checklist: Activate a Route with Code



Use the Router service

- Import the service
- Define it as a dependency

Create a method that calls the navigate method of the Router service

- Pass in the link parameters

Add a user interface element

- Use event binding to bind to the created method



Summary



Setting up

Configuring Routes

Tying Routes to Actions

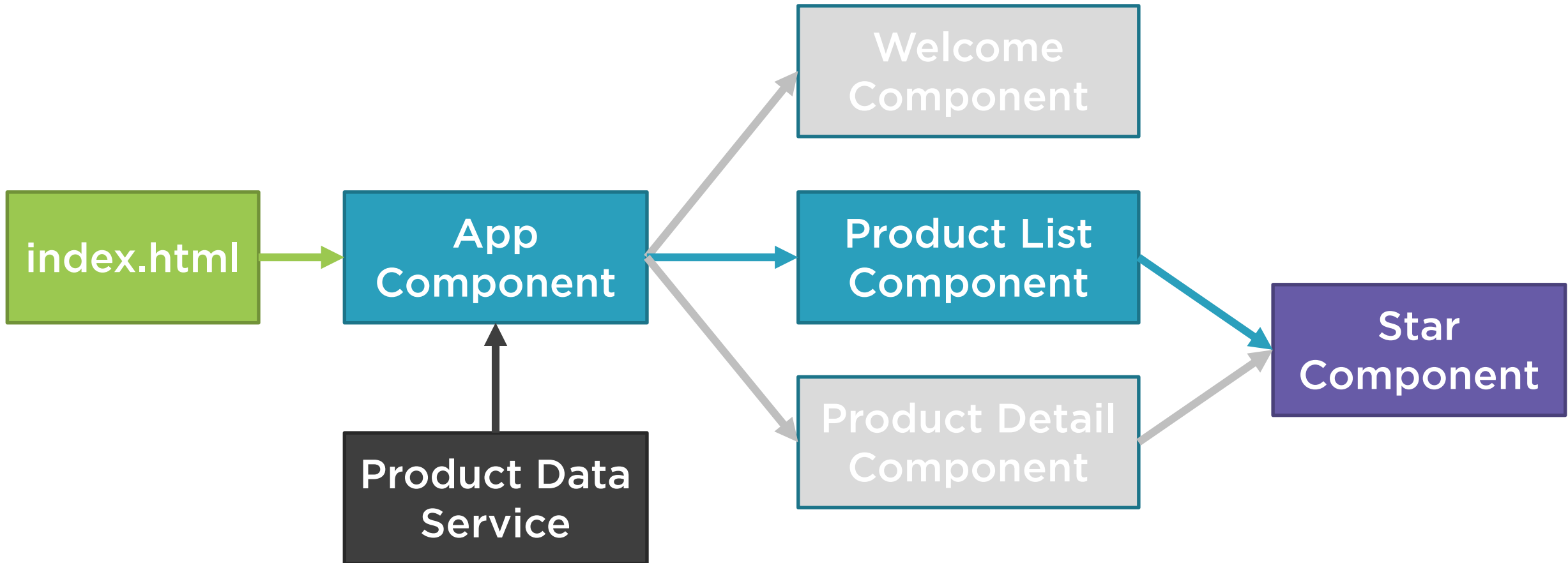
Placing the Views

Passing Parameters to a Route

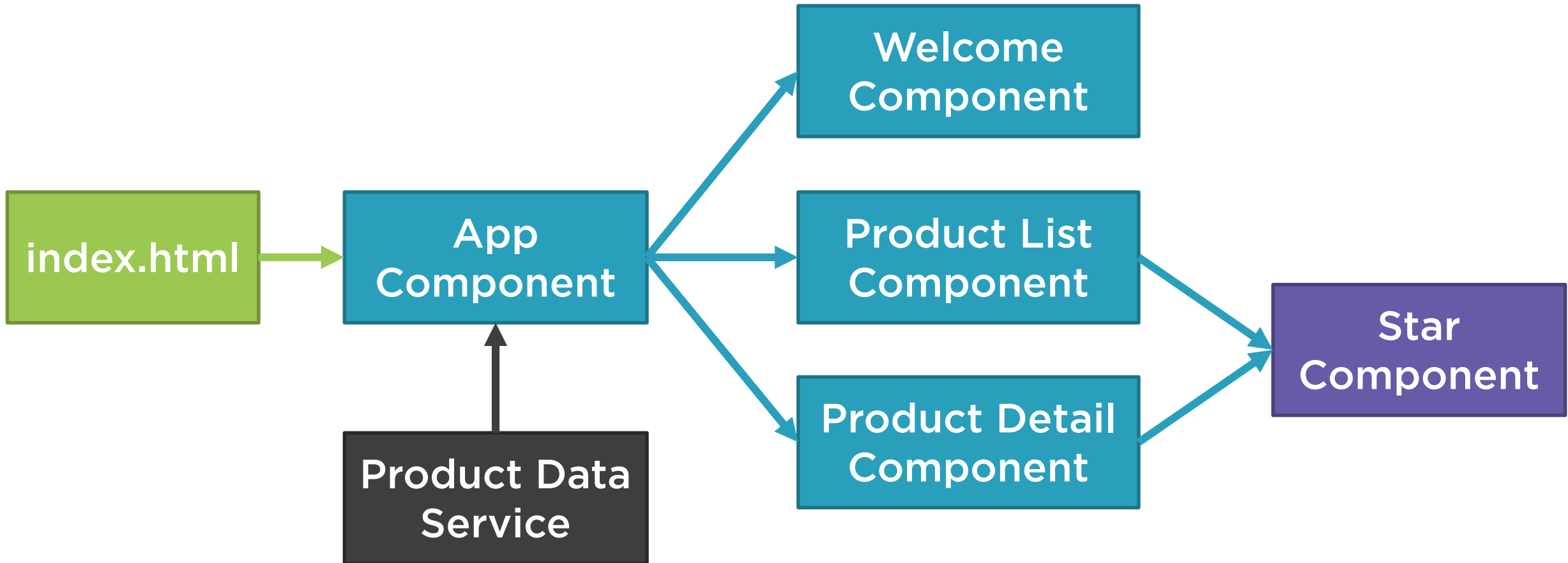
Activating a Route with Code



Application Architecture



Application Architecture



Up Next

