

Navigation and Routing Basics



Deborah Kurata

CONSULTANT | SPEAKER | AUTHOR | MVP | GDE

@deborahkurata | blogs.msmvps.com/deborahk/





Module Overview



How Does Routing Work?

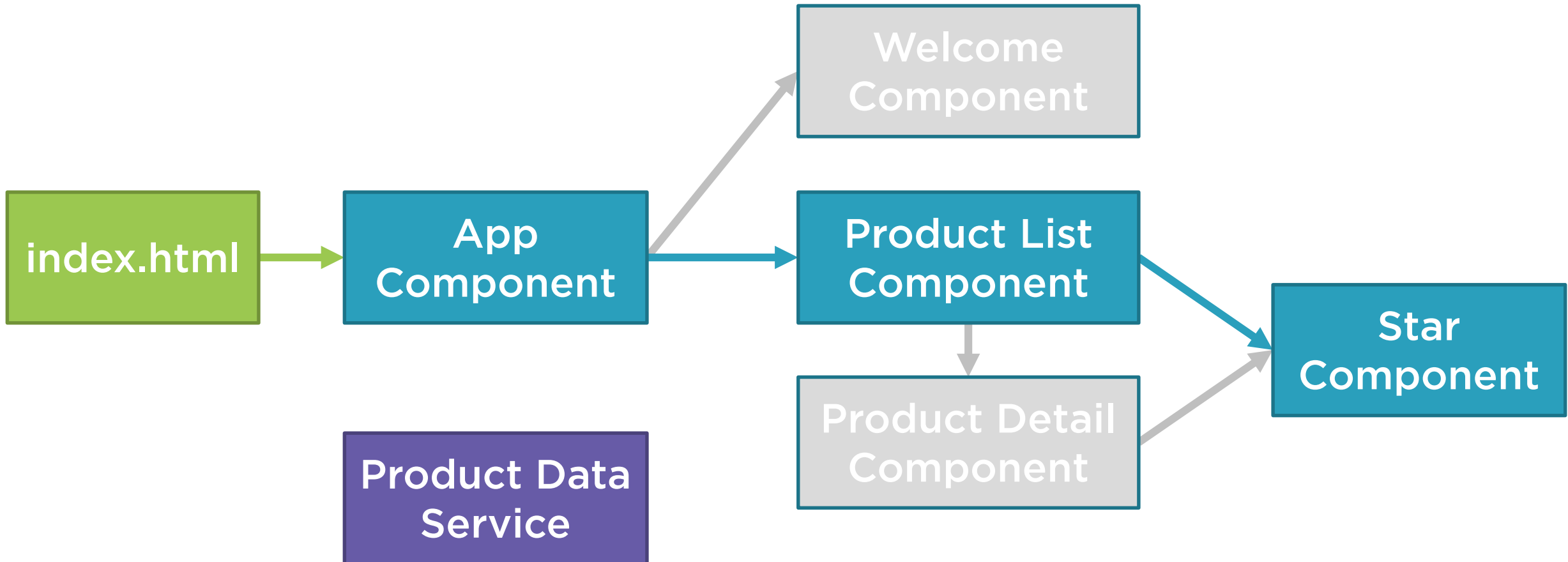
Configuring Routes

Tying Routes to Actions

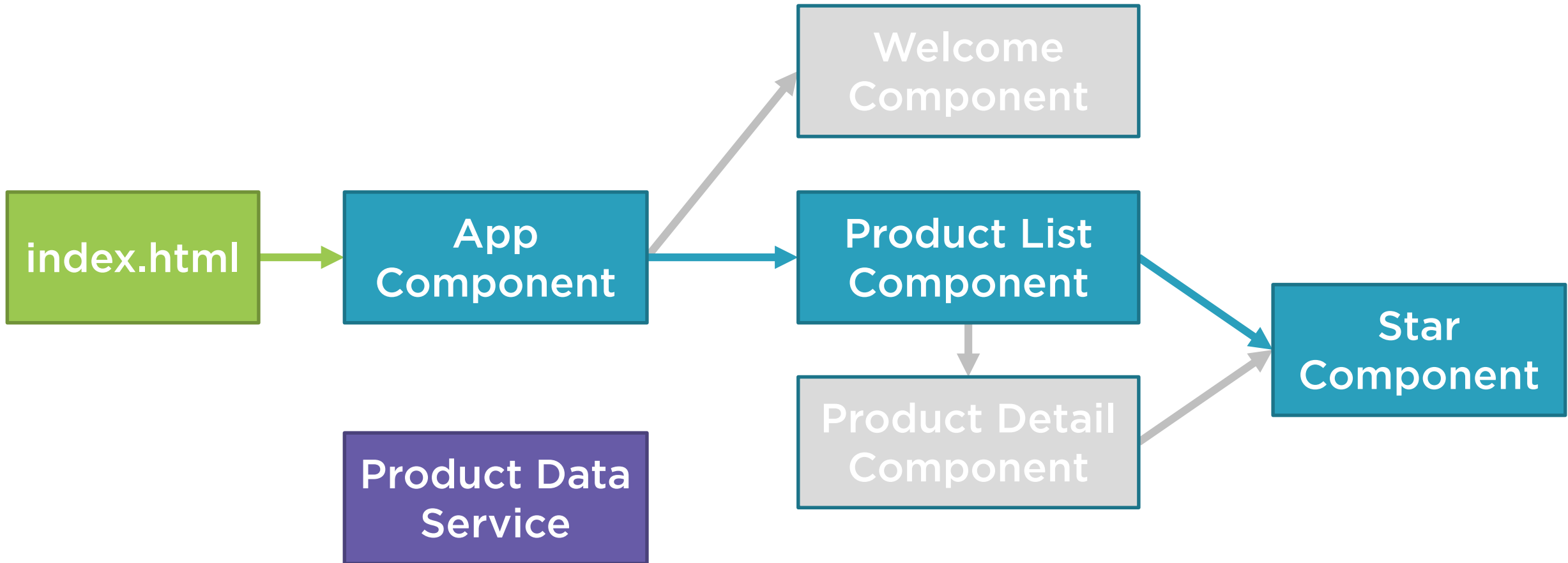
Placing the Views

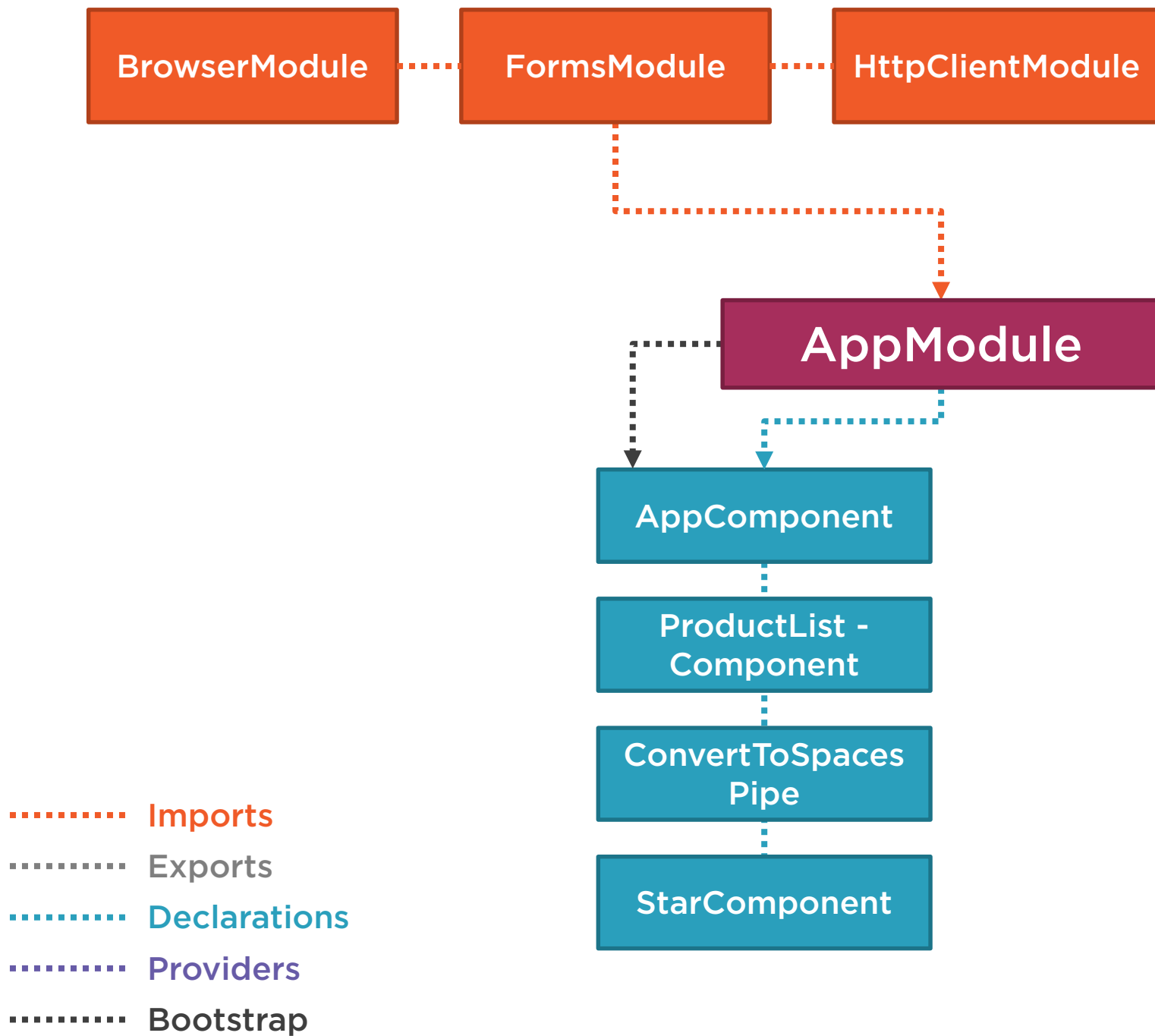


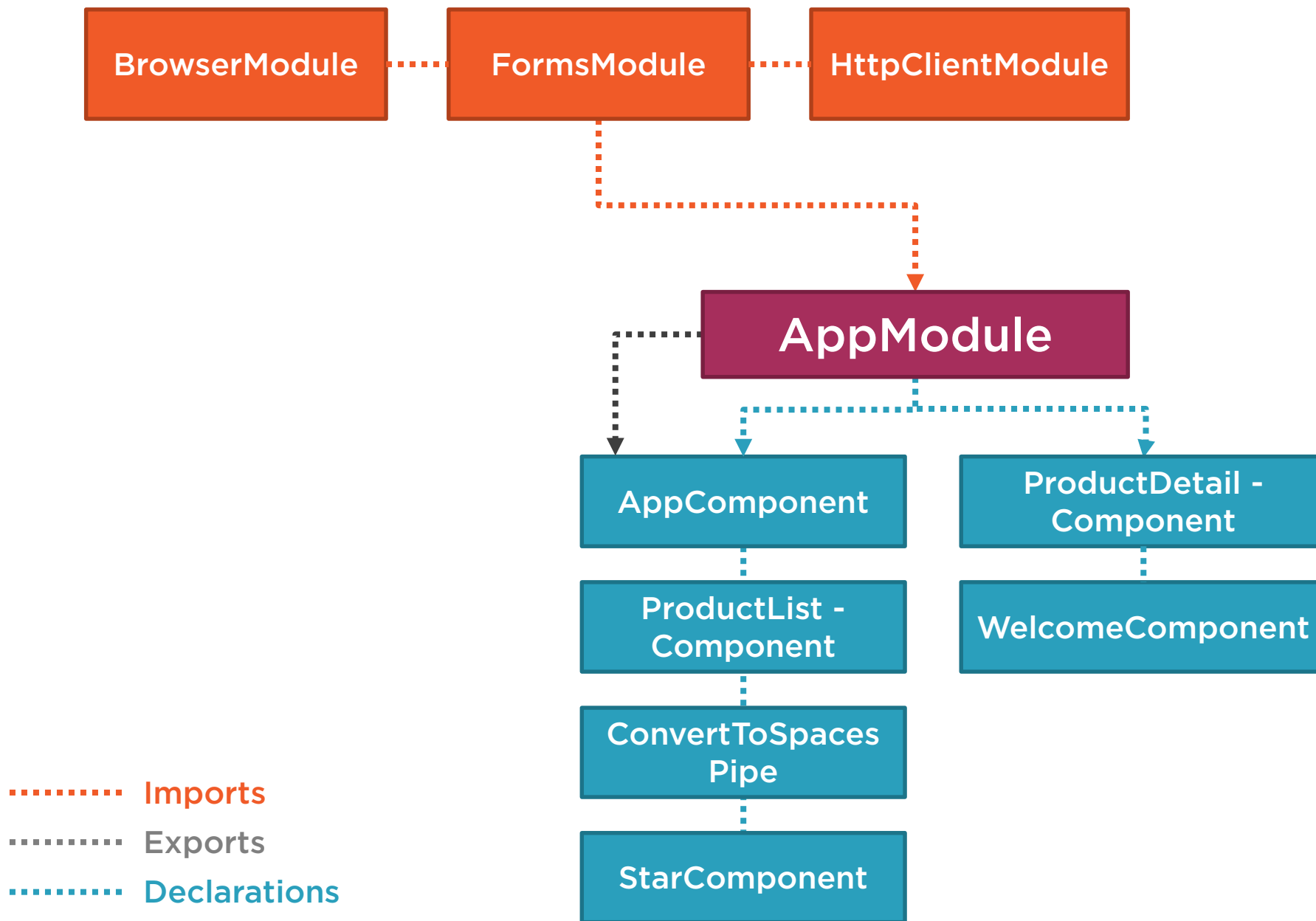
Application Architecture



Application Architecture







- Imports
- Exports
- Declarations
- Providers
- Bootstrap



How Routing Works

```
▼<pm-root _ngghost-c0 ng-version="6.0.5">
  ▶<nav _ngcontent-c0 class="navbar navbar-expand navbar-light bg-light">...</nav>
  ▼<div _ngcontent-c0 class="container">
    <router-outlet _ngcontent-c0></router-outlet>
    ▼<ng-component _ngghost-c1>
      ▼<div _ngcontent-c1 class="card">
        <div _ngcontent-c1 class="card-header">
          Product List </div>
        ▼<div _ngcontent-c1 class="card-body">
          ▶<div _ngcontent-c1 class="row">...</div>
          <!--bindings={
            "ng-reflect-ng-if": ""
          }-->
          ▼<div _ngcontent-c1 class="table-responsive">
            <!--bindings={
              "ng-reflect-ng-if": "5"
            }-->
            ▼<table _ngcontent-c1 class="table">
              ▶<thead _ngcontent-c1>...</thead>
              ▶<tbody _ngcontent-c1>...</tbody>
            </table>
          </div>
        </div>
      </div>
    </div>
    <!--bindings={
      "ng-reflect-ng-if": ""
    }-->
  </ng-component>
</div>
</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 }
```

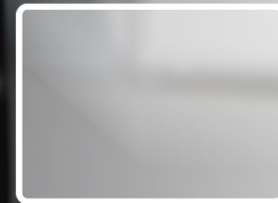
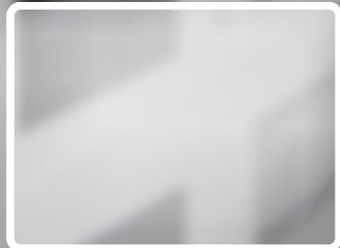
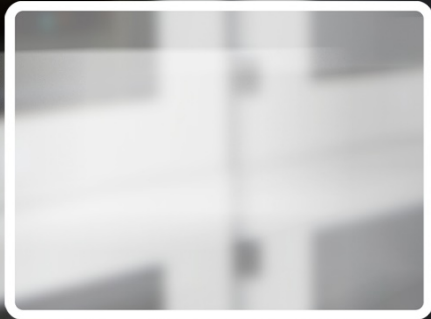
```
<router-outlet></router-outlet>
```

product-list.component.ts

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

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





'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

app.module.ts

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

@NgModule({
  imports: [
    BrowserModule,
    FormsModule,
    HttpClientModule,
    RouterModule.forRoot([], { useHash: true })
  ],
  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

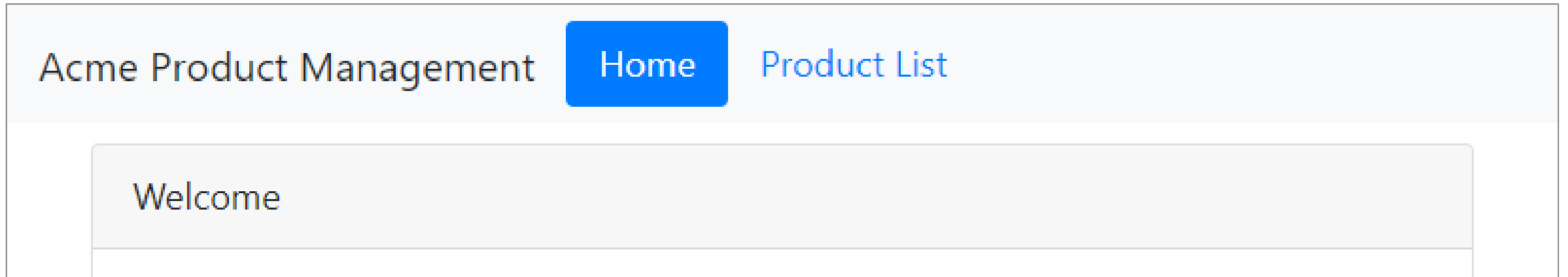


Menu option, link, image or button that activates a route

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>
  `
})
```



Placing the Views

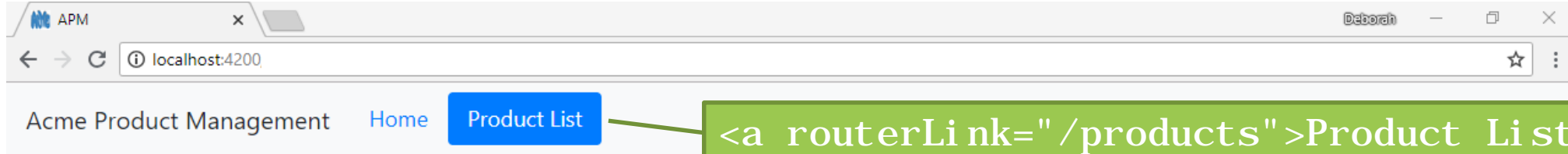
app.component.ts

```
...

@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 }
```

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

product-list.component.ts

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

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



Checklist: Displaying Components



Nest-able components

- Define a selector
- Nest 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



Routing 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



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

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

Summary



How Does Routing Work?

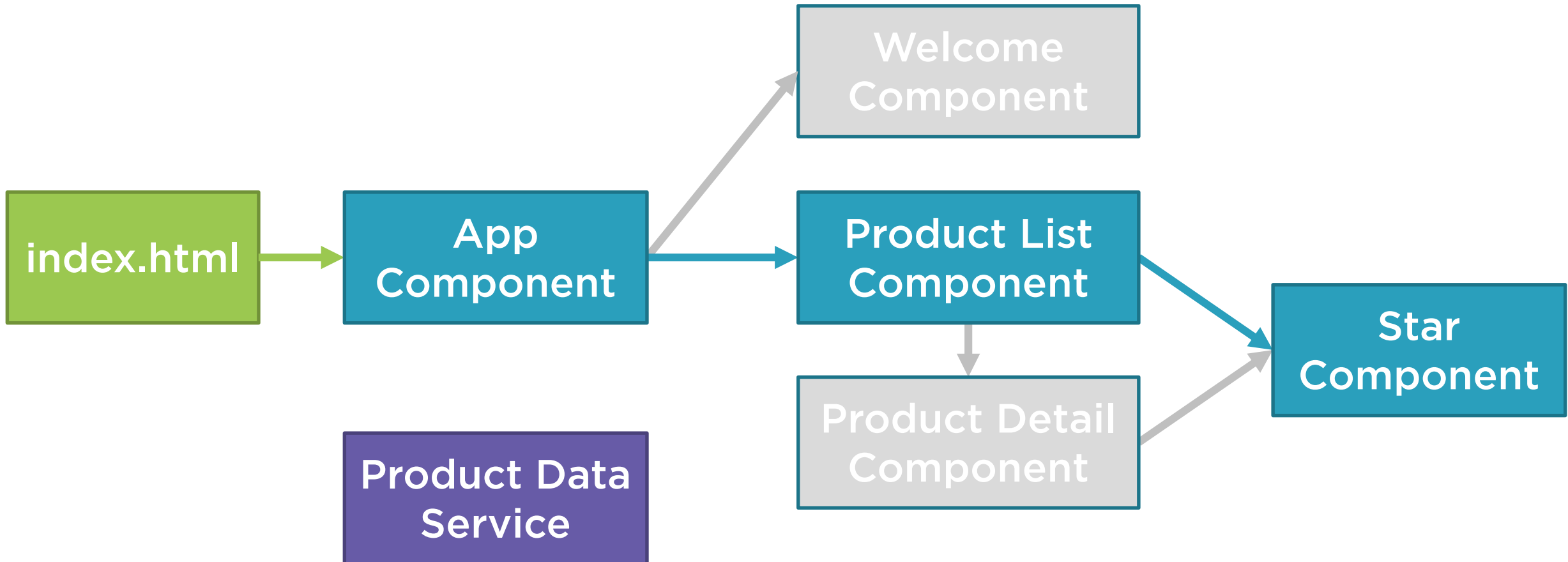
Configuring Routes

Tying Routes to Actions

Placing the Views



Application Architecture



Application Architecture

