

Angular 2.0

Interview Readiness Program
January, 2018

People matter, results count.



Objectives of Angular

■ Purpose:

- To have working knowledge (competency) in Angular 2

■ Product:

- Basic understanding of Angular 2
- To know Typescript features etc

■ Process:

- Theory Sessions along with hands on assignments
- A review at the end of the session and a Quiz.

Table of Contents

- Introduction to Angular
- Introduction to Components
- Templates , Interpolation and Directives
- Data Binding and Pipes
- More on Components
- Building Nested Components
- Services and Dependency Injection
- Retrieving Data using HTTP
- Navigation and Routing
- Angular Modules

Angular-Introduction

- A JavaScript framework
- For building client side Applications
- Using HTML,CSS and JavaScript

Why Angular



Expressive
HTML

Powerful
Data
Binding

Modular
By Design

Built-in
Back-End
Integration

What is Angular ?

Angular JS is an open source framework built over JavaScript.

It was built by the developers at Google.

Why new Angular



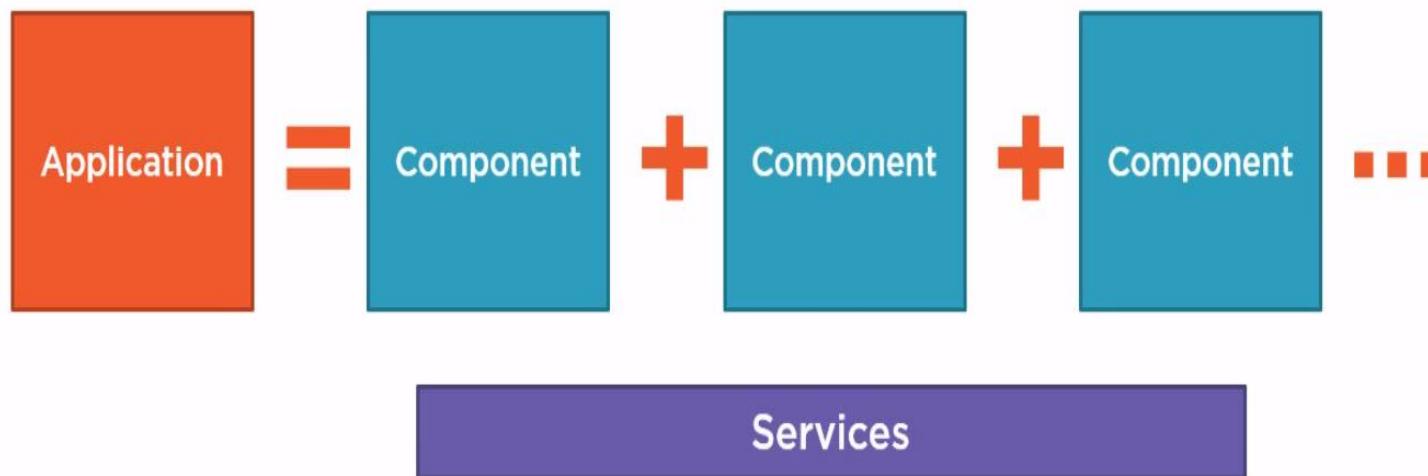
Built for
Speed

Modern

Simplified
API

Enhances
Productivity

Anatomy of a Angular Application



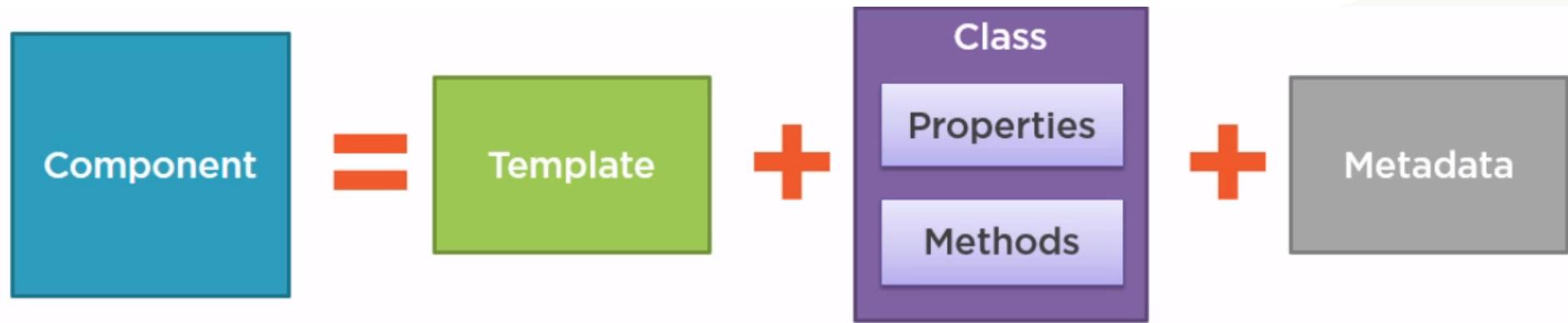
Features of Angular 2

- **Components** – Components help to build the applications into many modules. This helps in better maintaining the application over a period of time.
- **TypeScript** – The newer version of Angular is based on TypeScript. This is a superset of JavaScript and is maintained by Microsoft.
- **Services** – Services are a set of code that can be shared by different components of an application.
- **Example :**
- If you had a data component that picked data from a database, you could have it as a shared service that could be used across multiple applications

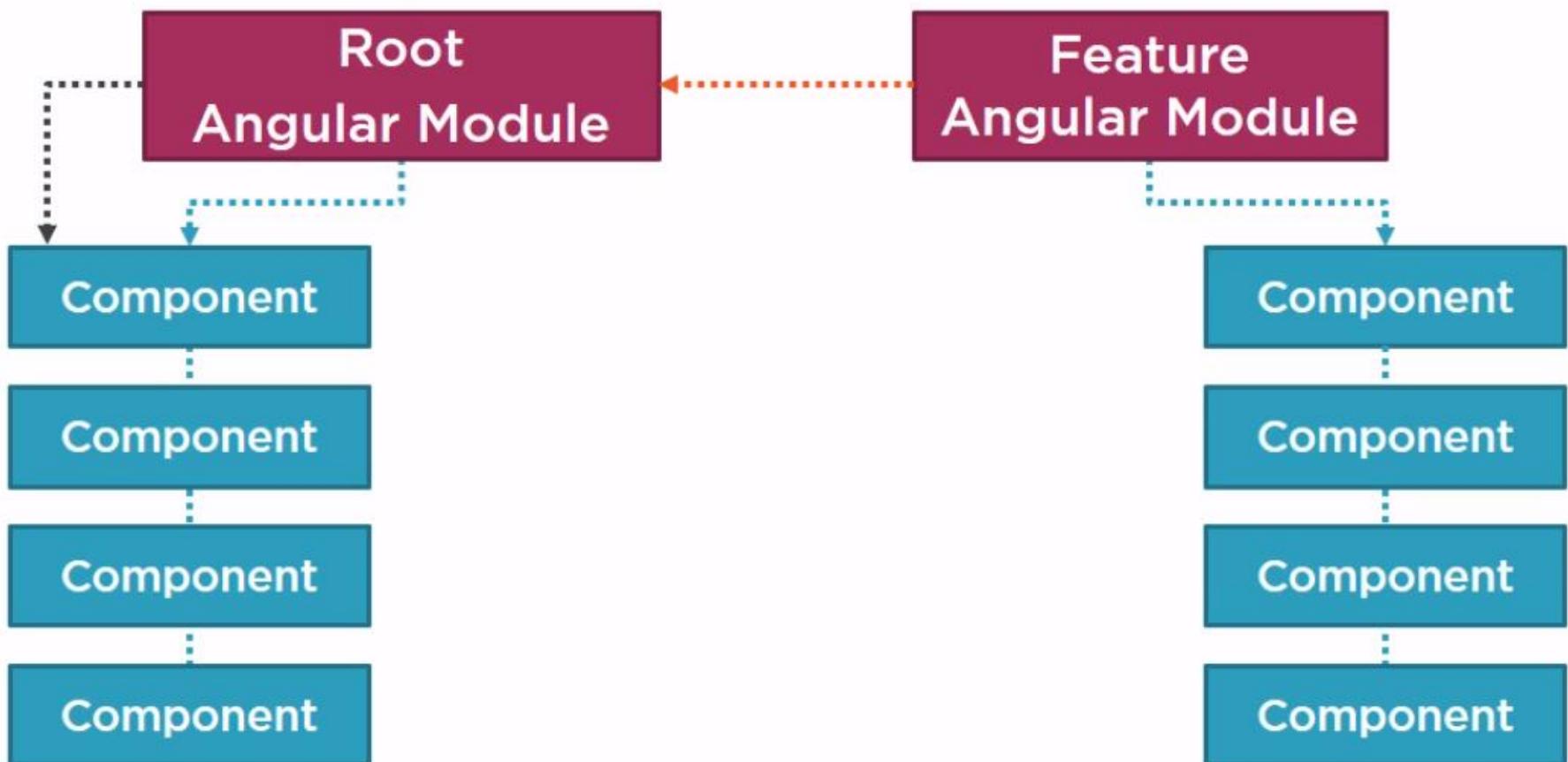
Components of Angular 2

- **Modules** – This is used to break up the application into logical pieces of code. Each piece of code or module is designed to perform a single task.
- **Component** – This can be used to bring the modules together.
- **Templates** – This is used to define the views of an Angular JS application.
- **Metadata** – This can be used to add more data to an Angular JS class.
- **Service** – This is used to create components which can be shared across the entire application

Component



Angular Modules

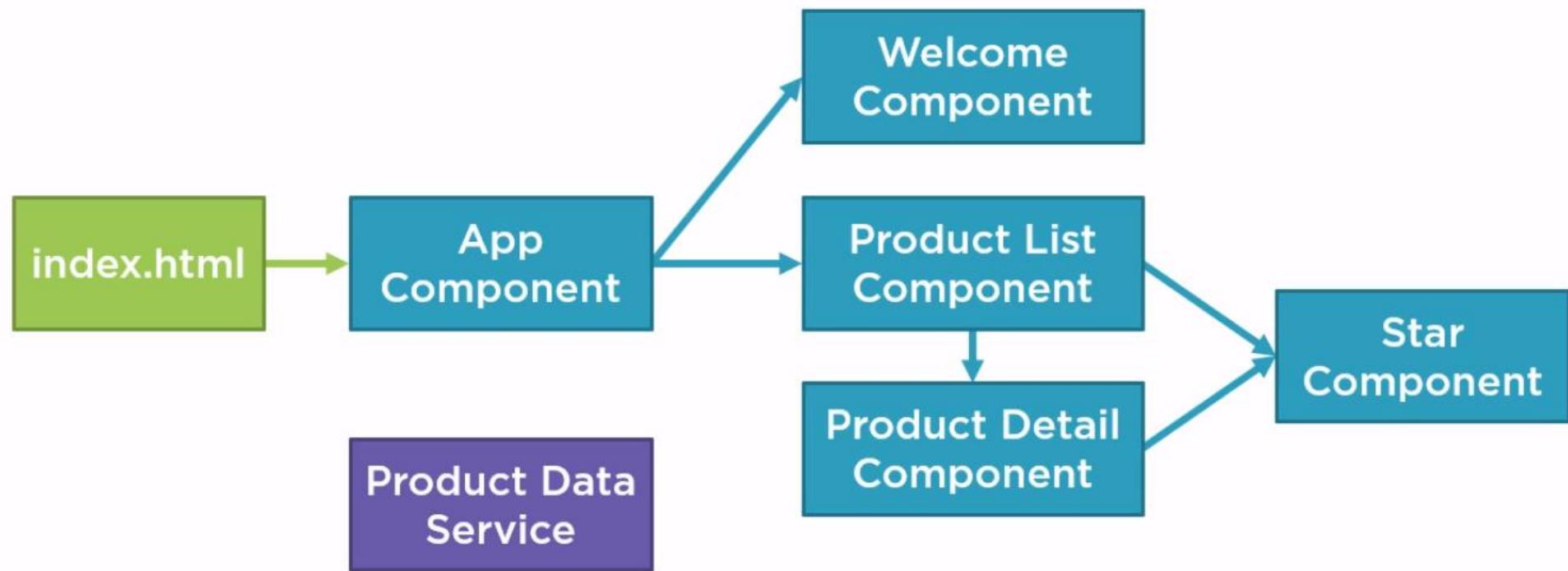


Sample Application

Acme Product Management Home Product List

Product List					
Filter by:					
Show Image	Product	Code	Available	Price	5 Star Rating
	Leaf Rake	GDN-0011	March 19, 2016	\$19.95	★★★
	Garden Cart	GDN-0023	March 18, 2016	\$32.99	★★★★
	Hammer	TBX-0048	May 21, 2016	\$8.9	★★★★★
	Saw	TBX-0022	May 15, 2016	\$11.55	★★★★
	Video Game Controller	GMG-0042	October 15, 2015	\$35.95	★★★★★

Sample Application Architecture



JavaScript Language Specification

- ECMAScript (ES)
- ES 3 - supported by all browsers
- ES 5 – supported by most modern browsers
- ES 2015 (formerly known as ES6)
 - most browsers do not support ES2015
 - code must be transpiled to ES5

Selecting a Language

ES 5

- Runs in the browser
- No compile required

ES 2015

- Lots of new features (classes, let, arrow, etc.)

TypeScript

- Superset of JavaScript
- Strong typing
- Great IDE tooling

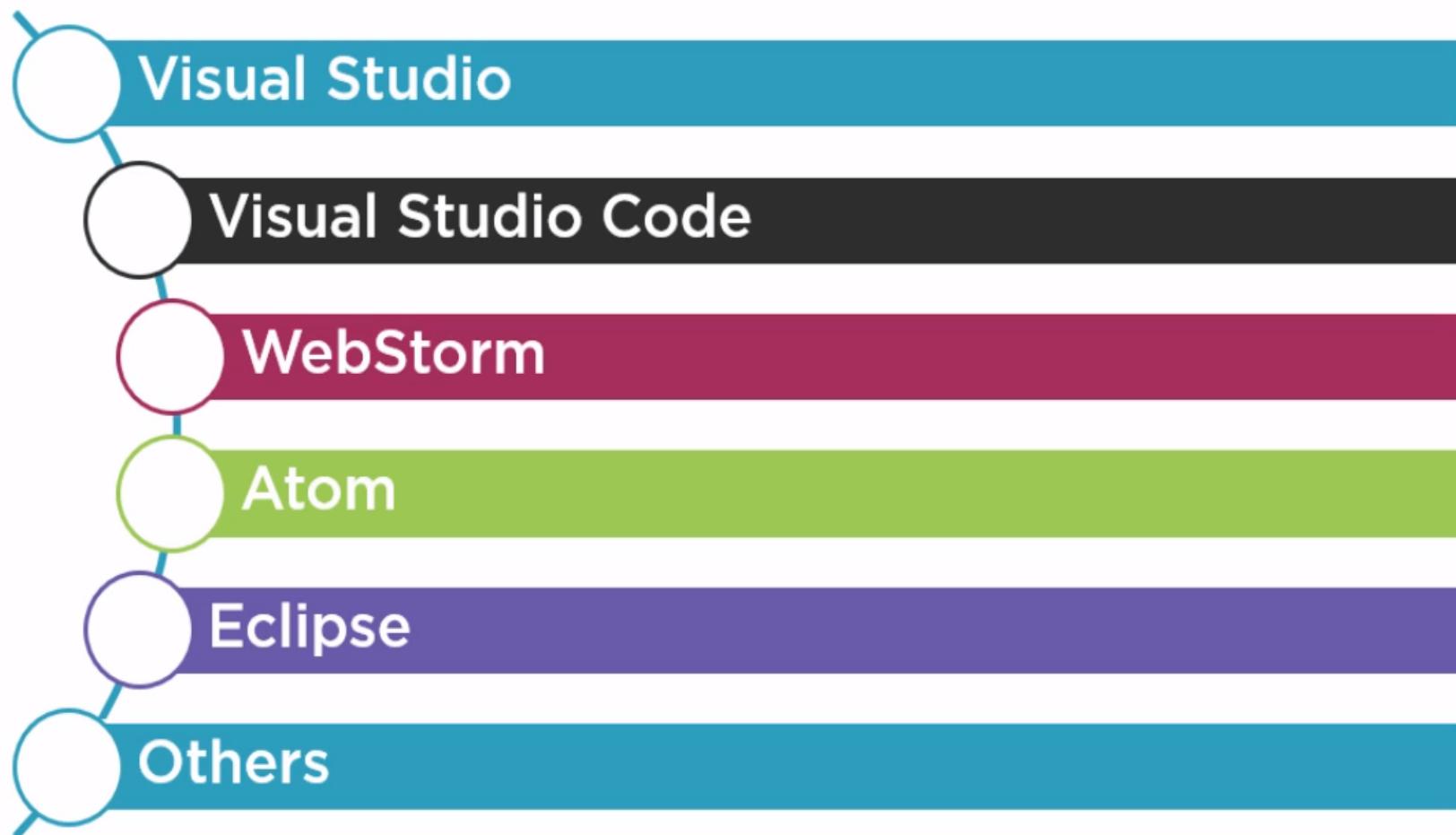
Dart

- No JavaScript

TypeScript

- Open Source Language
- Superset of JavaScript
- Transpiles to plain javascript
- Strongly typed
 - TypeScript type definition files (*.d.ts)
- TypeScript implements ES 2015 Class based Object-orientation

TypeScript Editors



Setting up Environment

- npm - Node package Manager
- Set up Angular Application

npm

- command line utility to interact with repository of Open Source projects
- package manager for javaScript
- Installs libraries , packages and applications
- <https://www.npmjs.com/>

Setting up an Angular Application

- Create an Application folder
- Add package definition and configuration files
- Install the packages
- Create the app's angular module
- Create main.ts file
- Create the host web page (index.html)

Setting up an Angular Application --- contd.,

- Manually perform each step
- Angular CLI
- Startup files

Different types of Files in Angular

- Source files
- Configuration files/ Set up files
- Dependency files (after executing npm install)

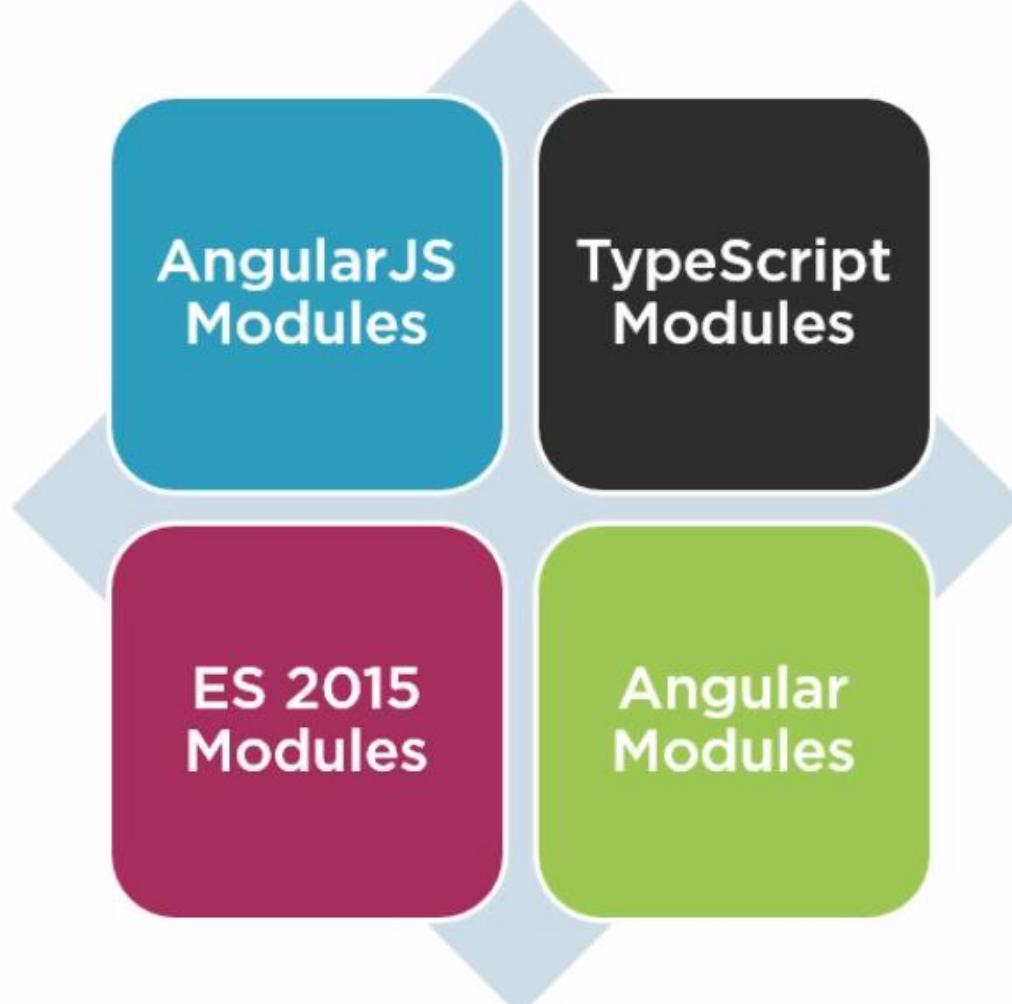
Running an Angular application

- npm install
- install the dependencies
- npm start
- start : ng –serve –o
 - ng → starts the cli
 - serve → build the application and starts the web server
 - -o → opens the url in our default browser

Modules

Namespaces

Code Organization



ES 2015 Modules

Export

product.ts

```
export class Product{  
}
```

product.js

```
function Product() {  
}
```

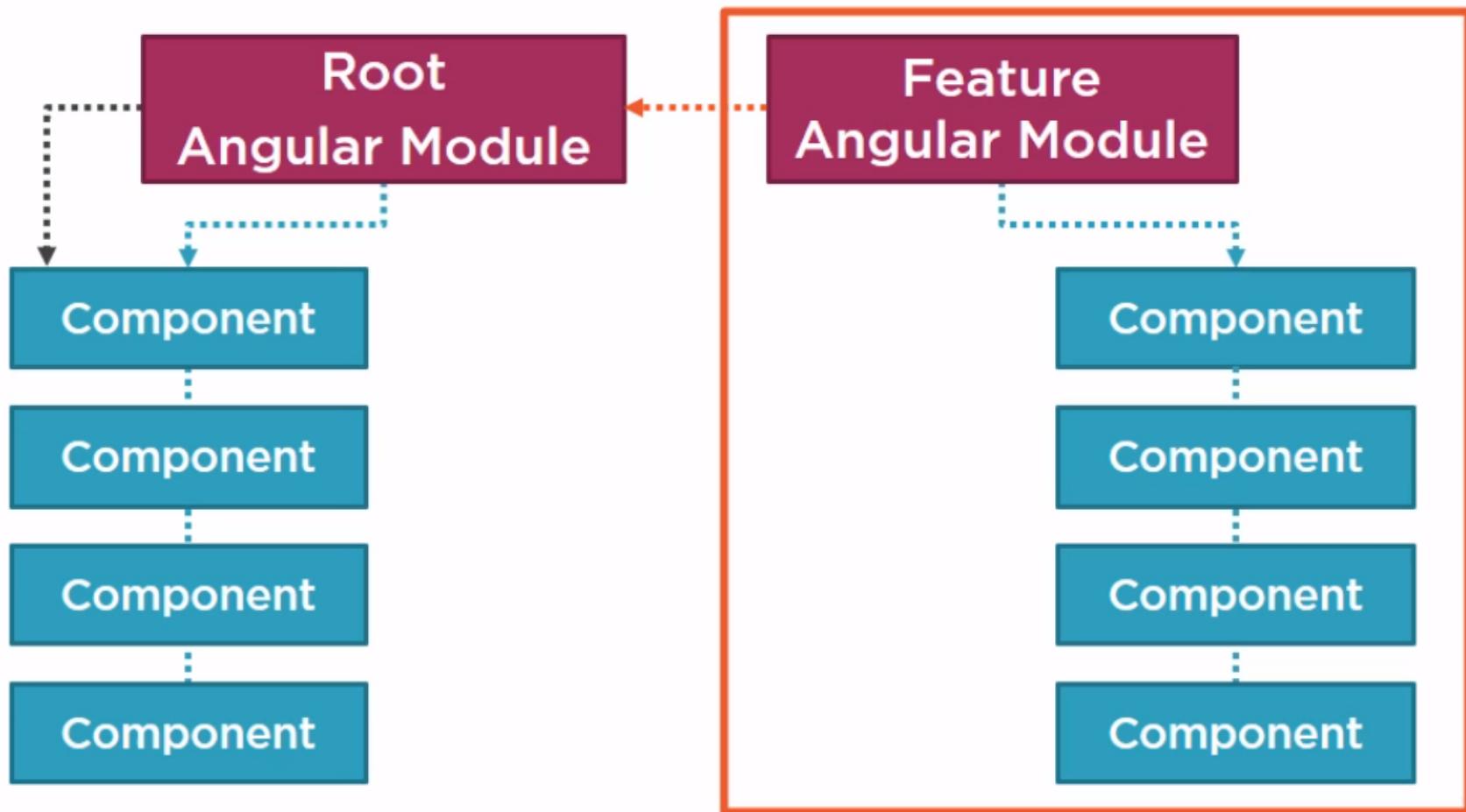
Import

product-list.ts

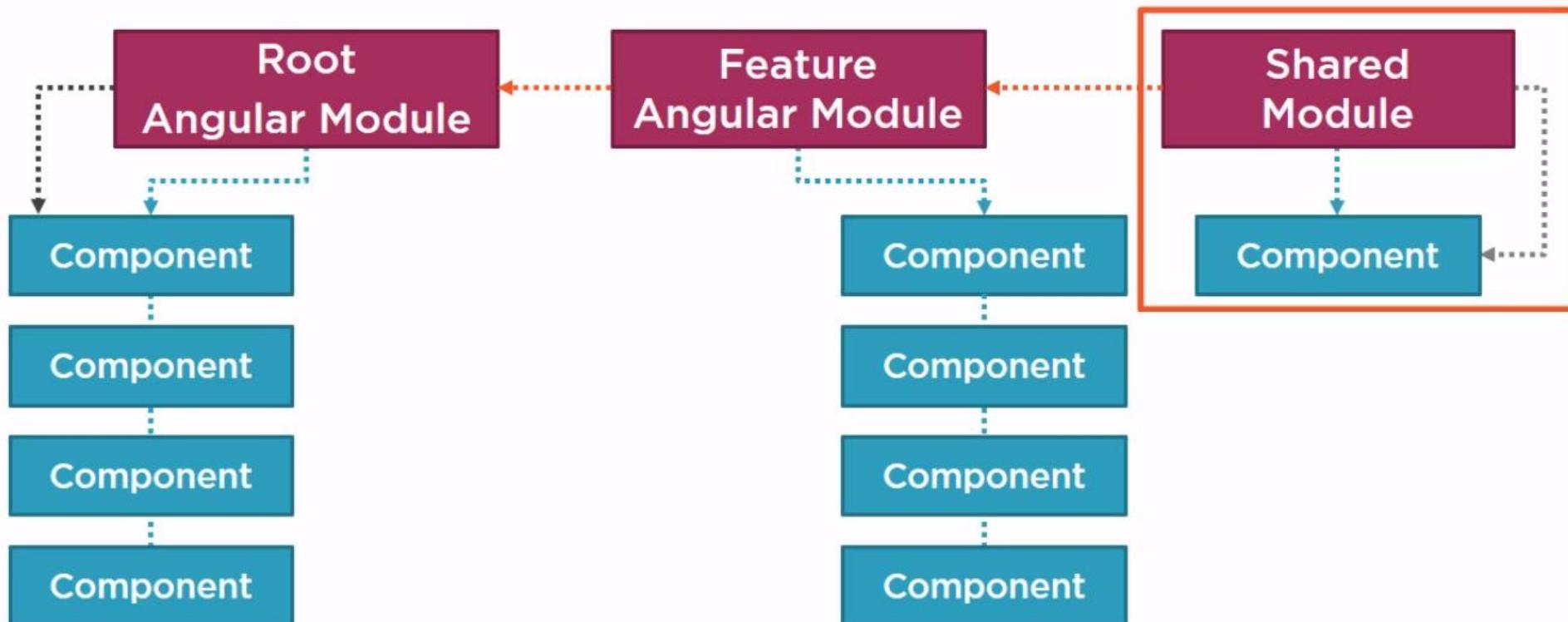
```
import { Product } from  
'./product'
```



Angular Modules



Angular Modules



Modules

ES Modules

Code files that import or export something

Organize our code files

Modularize our code

Promote code reuse

Angular Modules

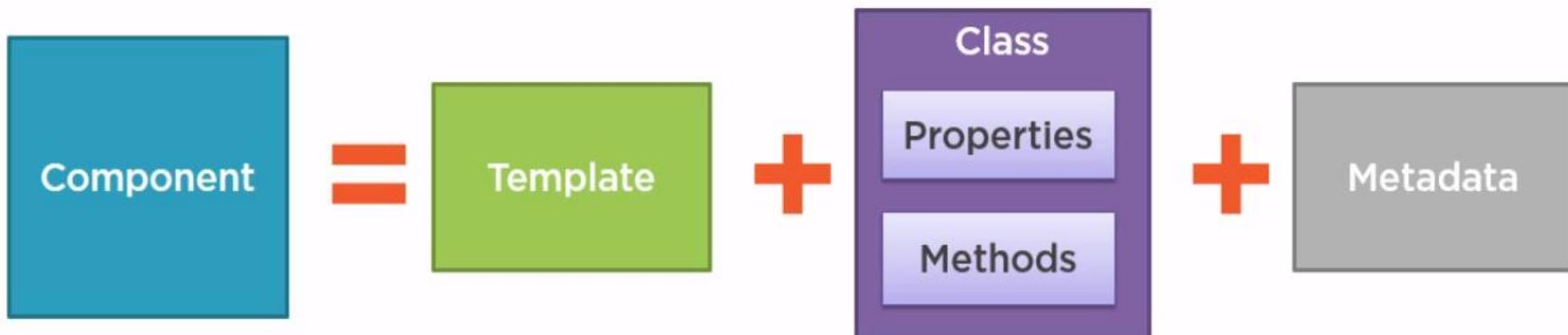
Code files that organize the application into cohesive blocks of functionality

Organize our application

Modularize our application

Promote application boundaries

What is a Component ?



- View layout
- Created with HTML
- Includes binding and directives
- Code supporting the view
- Created with TypeScript
- Properties: data
- Methods: logic
- Extra data for Angular
- Defined with a decorator

Component

app.component.ts

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

@Component({
  selector: 'pm-root',
  template:
    <div><h1>{{pageTitle}}</h1>
      <div>My First Component</div>
    </div>
}

export class AppComponent {
  pageTitle: string = 'Acme Product Management';
}
```

Import

Metadata &
Template

Class

Creating the Component Class

app.component.ts

```
export class AppComponent {  
  pageTitle: string = 'Acme Product Management';  
}
```

The diagram illustrates the components of the `AppComponent` class definition:

- class keyword**: Points to the `class` keyword in the code.
- export keyword**: Points to the `export` keyword in the code.
- Class Name**: Points to the identifier `AppComponent`.
- Component Name when used in code**: Points to the string value `'Acme Product Management'`.

Creating the Component Class

app.component.ts

```
export class AppComponent {  
  pageTitle: string = 'Acme Product Management';  
}
```

Property
Name

Data Type

Default
Value

Methods

Decorator

- A function that adds metadata to a class, its members , or its method arguments
- prefixed with @
- Angular has built in decorators
- `@Component()`

Defining the metadata

app.component.ts

```
@Component({  
  selector: 'pm-root',  
  template:  
    <div><h1>{{pageTitle}}</h1>  
      <div>My First Component</div>  
    </div>  
}  
)  
export class AppComponent {  
  pageTitle: string = 'Acme Product Management'  
}
```

Component
decorator

Directive Name
used in HTML

View Layout

Binding

Importing What We Need

- Before we use an external function or class, we define where to find it

Importing what we need



Before we use an external function or class,
we define where to find it

import statement

import allows us to use exported members
from external ES modules

**Import from a third-party library, our own
ES modules, or from Angular**

Importing what we need

@angular/
core

@angular/
animate

@angular/
http

@angular/
router

<https://www.npmjs.com/~angular>

Importing What we need

app.component.ts

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

@Component({
  selector: 'pm-root',
  template:
    <div><h1>{{pageTitle}}</h1>
      <div>My First Component</div>
    </div>
}

export class AppComponent {
  pageTitle: string = 'Acme Product Management';
}
```

import keyword

Angular library
module name

Member name

Completed Component

app.component.ts

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

@Component({
  selector: 'pm-root',
  template: `
    <div><h1>{{pageTitle}}</h1>
      <div>My First Component</div>
    </div>
  `

})
export class AppComponent {
  pageTitle: string = 'Acme Product Management';
}
```

Sample

```
@Component({})
```

```
  [ts] Cannot find name 'Component'.
```

```
}) any
```

management';

Contd.,

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

@Component({
  s
}) selector      (property) Directive.selector: string ⓘ
exp styles
  styleUrls
} set
  setTimeout
  switch
```

App.component.ts

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

@Component({
  selector: 'pm-root'
})
export class AppComponent {
  pageTitle: string = 'Acme Product Management';
}
```

App.component.ts

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

@Component({
  selector: 'pm-root',
  template: `
    <div><h1>{{pageTitle}}</h1>
      <div>My First Component</div>
    </div>
  `})
export class AppComponent {
  pageTitle: string = 'Acme Product Management';
}
```

Back Ticks
NOT Quotes

Bootstrapping the app Component

- Host the application with index.html
- Defining the angular module
- Index.html contains the main page for the application
- Called as Single Page Application

Hosting the Application

index.html

```
<body>
<pm-root></pm-root>
</body>
```

app.component.ts

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

@Component({
  selector: 'pm-root',
  template:
    <div><h1>{{pageTitle}}</h1>
      <div>My First Component</div>
    </div>

})
export class AppComponent {
  pageTitle: string = 'Acme Product Management';
}
```

Hosting the Application

index.html

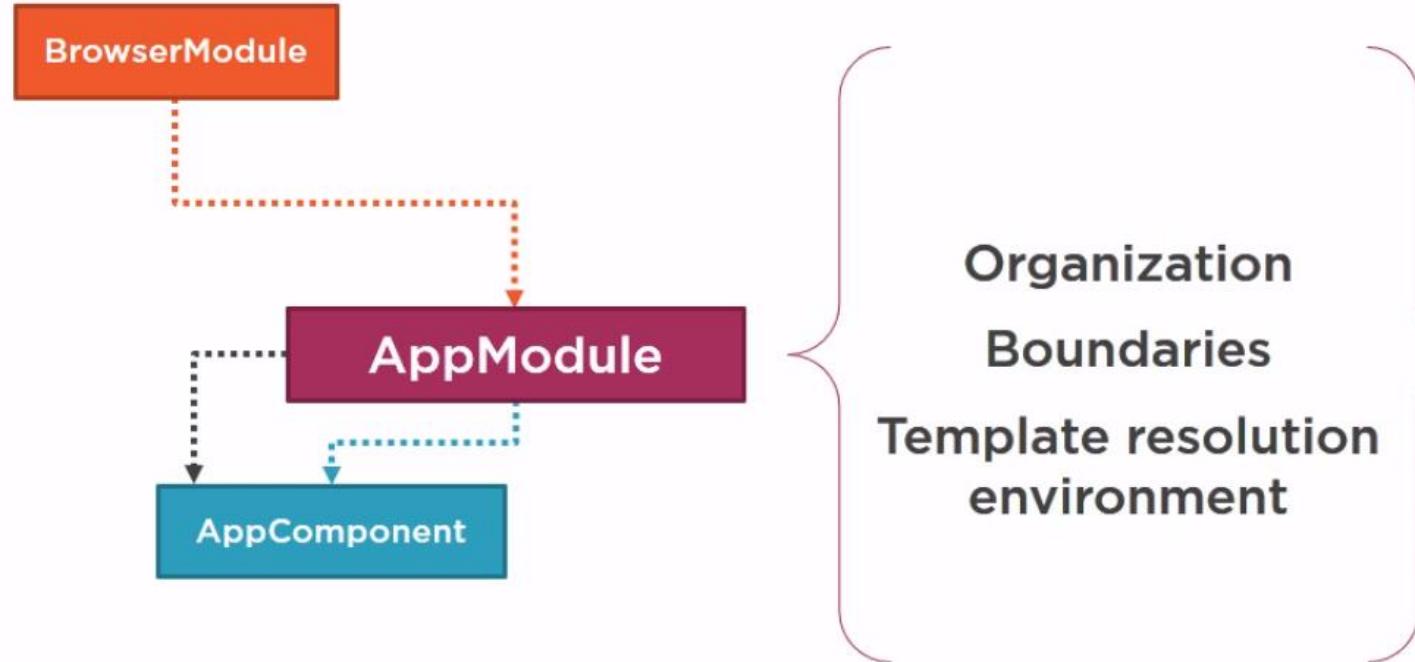
```
<body>
<pm-root></pm-root>
</body>
```

app.component.ts

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

@Component({
  selector: 'pm-root',
  template:
    <div><h1>{{pageTitle}}</h1>
      <div>My First Component</div>
    </div>
})
export class AppComponent {
  pageTitle: string = 'Acme Product Management';
}
```

Bootstrapping our App Component



- Imports
- Exports
- Declarations
- Providers
- Bootstrap

Defining the Angular Module

app.module.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppComponent } from './app.component';

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

Index.html

```
TS app.component.ts    <> index.html x
1  <!doctype html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8">
5      <title>Acme Product Management</title>
6      <base href="/">
7
8      <meta name="viewport" content="width=device-width, initial-scale=1">
9      <link rel="icon" type="image/x-icon" href="favicon.ico">
10     </head>
11     <body>
12         <pm-root></pm-root>
13     </body>
14 </html>
15
```

app.module.ts

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';

import { AppComponent } from './app.component';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

Bootstrapping the App

The screenshot shows a web browser window with the URL `localhost:4200` in the address bar. The main content area displays the title "Acme Product Management" and a sub-section titled "My First Component". Below the main content is the Angular developer tools toolbar, which includes tabs for Elements, Console, Sources, Network, Performance, Memory, Application, Security, and Audits. The "Elements" tab is currently selected. At the bottom of the browser window, a message in the console states: "Angular is running in the development mode. Call enableProdMode() to enable the production mode." with the file reference "vendor.bundle.js:46569".

Contd.,

Acme Product Management

My First Component

The screenshot shows the Chrome Developer Tools interface with the 'Console' tab selected. The console output area contains the following message:

```
Angular is running in the development mode. Call enableProdMode() to enable the production mode.
```

The entire message is highlighted with a red rectangular box.

Contd.,

Acme Product Management

My First Component

The screenshot shows the Chrome DevTools interface with the 'Elements' tab selected (highlighted with a red box). The left panel displays the DOM tree, starting with the <html> element. The body element is expanded, showing its children: <pm-root>, <script> tags for inline.bundle.js, polyfills.bundle.js, styles.bundle.js, vendor.bundle.js, and main.bundle.js, and finally </body>. The right panel shows the 'Styles' tab with the CSS rules applied to the body element. The rules include:

```
element.style {  
}  
body {  
    font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;  
    font-size: 14px;  
    line-height: 1.42857143;  
    color: #333;  
    background-color: #fff;  
}  
body {  
    margin: 0;  
}  
* {  
    box-sizing: border-box;  
}  
body {  
    user agent styles  
}
```

Contd.,

Acme Product Management

My First Component

The screenshot shows the Chrome DevTools interface with the 'Elements' tab selected. A red box highlights the DOM tree on the left, which displays the following HTML structure:

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body> == $0
    <pm-root ng-version="4.2.4">...</pm-root>
    <script type="text/javascript" src="inline.bundle.js"></script>
    <script type="text/javascript" src="polyfills.bundle.js"></script>
    <script type="text/javascript" src="styles.bundle.js"></script>
    <script type="text/javascript" src="vendor.bundle.js"></script>
    <script type="text/javascript" src="main.bundle.js"></script>
  </body>
</html>
```

The right panel shows the 'Styles' tab with the following CSS rules:

```
element.style { }
body {
  font-family: "Helvetica Neue", Helvetica, Arial;
  font-size: 14px;
  line-height: 1.42857143;
  color: #333;
  background-color: #fff;
}
body {
  margin: 0;
```

Contd.,

Acme Product Management

My First Component

The screenshot shows the browser's developer tools with the 'Elements' tab selected. The left panel displays the DOM tree for the 'Acme Product Management' page. A red box highlights the first two levels of the tree: the root node '' and its child '

' element, which contains the text 'Acme Product Management' and another '

' element with the text 'My First Component'. The right panel shows the CSS styles applied to the body element. The 'element.style' section includes rules for 'body' (font-family: "Helvetica Neue", Helvetica, Arial, sans-serif; font-size: 14px; line-height: 1.42857143; color: #333; background-color: #fff;) and 'body' (margin: 0;). The '*' section includes a rule for 'box-sizing: border-box;'. The 'body' section at the bottom is partially cut off.

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body> == $0
    <pm-root ng-version="4.2.4">
      <div>
        <h1>Acme Product Management</h1>
        <div>My First Component</div>
      </div>
    </pm-root>
    <script type="text/javascript" src="inline.bundle.js"></script>
    <script type="text/javascript" src="polyfills.bundle.js"></script>
    <script type="text/javascript" src="styles.bundle.js"></script>
    <script type="text/javascript" src="vendor.bundle.js"></script>
    <script type="text/javascript" src="main.bundle.js"></script>
  </body>
</html>
```

Styles Computed Event Listeners DOM Breakpoints Properties

Filter

```
element.style { }
body {
  font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;
  font-size: 14px;
  line-height: 1.42857143;
  color: #333;
  background-color: #fff;
}
body {
  margin: 0;
}
* {
  box-sizing: border-box;
}
body {
```

View and debug the code

Acme Product Management

My First Component

The screenshot shows the Chrome DevTools interface with the 'Sources' tab selected (highlighted with a red box). The left sidebar lists files under the 'localhost:4200' connection, including '(index)', 'inline.bundle.js', 'main.bundle.js', 'polyfills.bundle.js', 'styles.bundle.js', and 'vendor.bundle.js'. The '(index)' file is currently selected and its content is displayed in the main pane. The content includes meta tags for viewport and icon, followed by a script tag pointing to 'inline.bundle.js'. The right sidebar contains sections for 'Watch', 'Call Stack' (labeled 'Not Paused'), 'Scope' (labeled 'Not Paused'), 'Breakpoints' (with a single entry for 'app.component.ts:9'), 'XHR Breakpoints', 'DOM Breakpoints', 'Global Listeners', and 'Event Listener Breakpoints'.

```
7 <meta name="viewport" content="width=device-width, initial-scale=1">
8 <link rel="icon" type="image/x-icon" href="favicon.ico">
9
10 </head>
11 <body>
12 <pm-root></pm-root>
13 <script type="text/javascript" src="inline.bundle.js"></script><script type="text/javascript" src="polyfills.bundle.js"></script>
14 </body>
15 </html>
```

TypeScript files under Webpack

The screenshot shows the Chrome DevTools interface with the 'Sources' tab selected. On the left, the file tree displays a local host entry with several bundled files: inline.bundle.js, main.bundle.js, polyfills.bundle.js, styles.bundle.js, and vendor.bundle.js. Below this, there are entries for '(no domain)' and 'ng://'. A red box highlights the 'webpack://...' entry, which contains sub-folders for '(webpack)-dev-server', '(webpack)', and 'C:/Users/Deborah/Pluralsight/Angular Getting Started/APM'. The main pane shows the source code for the '(index)' file, which includes meta tags for viewport and icon, a pm-root placeholder, and a script tag pointing to inline.bundle.js.

```
7<meta name="viewport" content="width=device-width, initial-scale=1">
8<link rel="icon" type="image/x-icon" href="favicon.ico">
9
10</head>
11<body>
12<pm-root></pm-root>
13<script type="text/javascript" src="inline.bundle.js"></script><script type="text/javascript" src="polyfills.bundle.js"></script>
14</body>
15</html>
```

The right sidebar contains developer tools for 'Watch', 'Call Stack' (not paused), 'Scope' (not paused), 'Breakpoints' (with one entry for app.component.ts:9), 'XHR Breakpoints', 'DOM Breakpoints', 'Global Listeners', and 'Event Listener Breakpoints'.

Building a Template

app.component.ts

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

@Component({
  selector: 'pm-root',
  template: `
    <div><h1>{pageTitle}</h1>
      <div>My First Component</div>
    </div>
  `
})
export class AppComponent {
  pageTitle: string = 'Acme Product Management';
}
```

Defining a template in a Component

Inline Template

```
template:  
"<h1>{{pageTitle}}</h1>"
```

Inline Template

```
template:  
<div>  
  <h1>{{pageTitle}}</h1>  
  <div>  
    My First Component  
  </div>  
</div>
```

ES 2015
Back Ticks

Linked Template

```
templateUrl:  
'./product-list.component.html'
```

Product List View

Product List

Filter by:

 x

Filtered by: am

Show Image

Product	Code	Available	Price	5 Star Rating
Hammer	TBX-0048	May 21, 2016	\$8.99	★★★★★
Video Game Controller	GMG-0042	Oct 15, 2015	\$35.95	★★★★★

Product List View

Product List

Filter by:

Filtered by: am

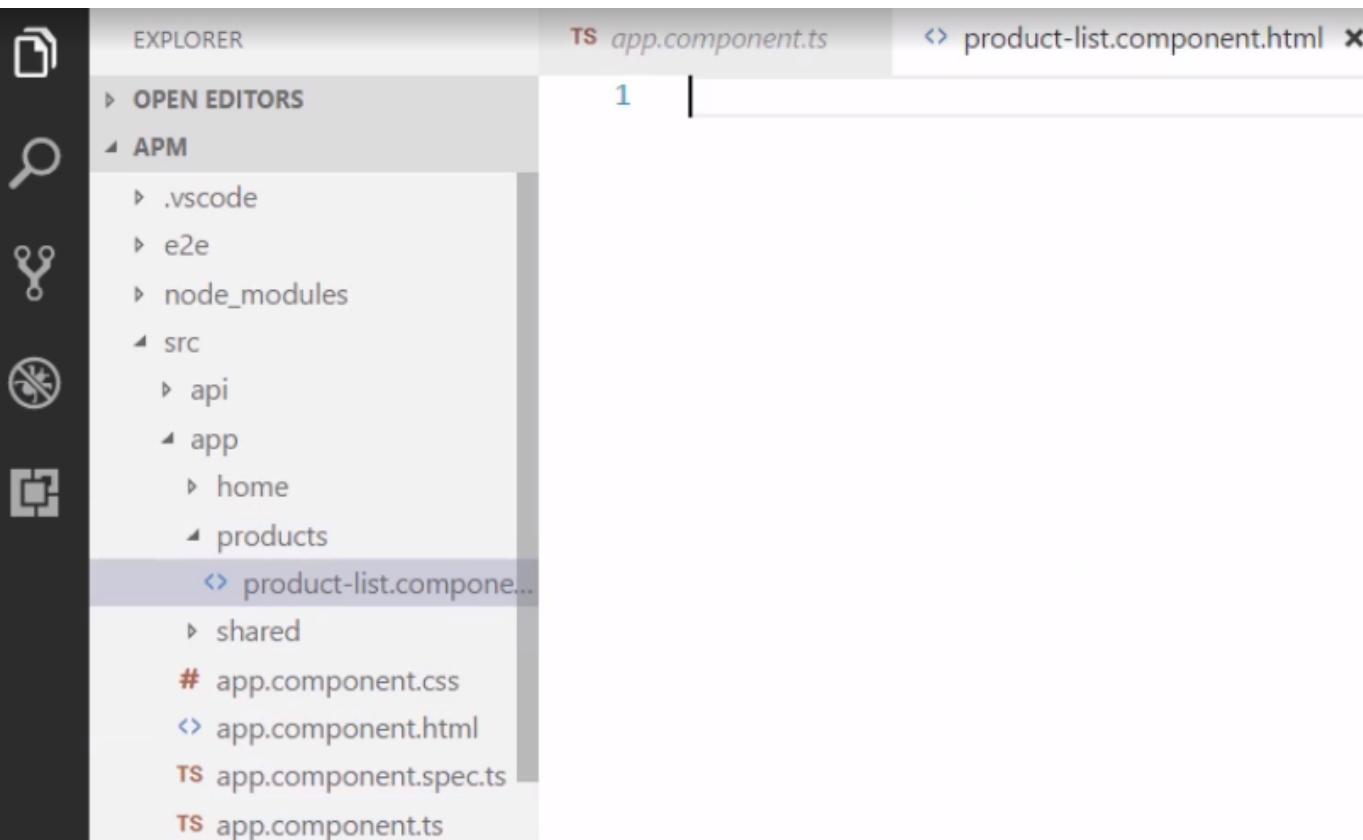
<input type="button" value="Hide Image"/>	Product	Code	Available	Price	5 Star Rating
	Hammer	TBX-0048	May 21, 2016	\$8.99	★★★★★
	Video Game Controller	GMG-0042	Oct 15, 2015	\$35.95	★★★★★

Adding External Link Template

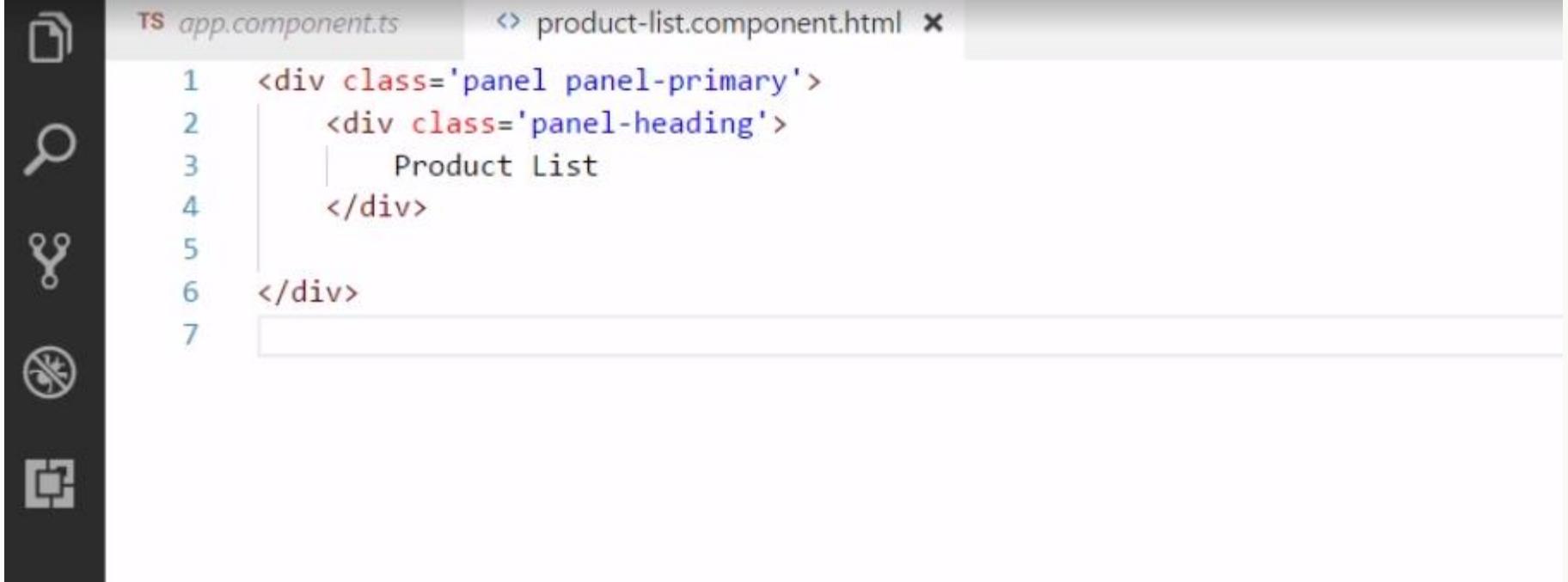
The screenshot shows the Visual Studio Code interface. On the left is the Explorer sidebar with icons for File, Open Editors, APM, .vscode, e2e, node_modules, and a src folder containing api and app. The app folder is selected and has a context menu open. The menu items are: New File, New Folder, Reveal in Explorer (Alt+Shift+R), Open in Command Prompt (Ctrl+Shift+C), Find in Folder, Copy (Ctrl+C), Paste (Ctrl+V), Copy Path (Alt+Shift+C), Rename (F2), Delete (Del), and index.html. Below the menu is a file named main.ts. The main.ts file contains the following code:

```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'pm-root',
5   template: `
6     <div><h1>{{pageTitle}}</h1>
7       <div>My First Component</div>
8     </div>
9   `
10})
11export class AppComponent {
12  pageTitle: string = 'Acme Product Management';
13}
```

Create a product-list.component.html



Product-list.component.html



The screenshot shows a code editor interface with a dark sidebar containing five icons: a file, a magnifying glass, a gear, a crossed-out document, and a square. The main area has two tabs: 'app.component.ts' (marked with a TS icon) and 'product-list.component.html'. The 'product-list.component.html' tab is active, displaying the following code:

```
1 <div class='panel panel-primary'>
2   <div class='panel-heading'>
3     |   Product List
4   </div>
5
6 </div>
7
```

Contd.,

TS app.component.ts <> product-list.component.html ✘

```
1  <div class='panel panel-primary'>
2    <div class='panel-heading'>
3      Product List
4    </div>
5    <div class='panel-body'>
6      <div class='row'>
7        <div class='col-md-2'>Filter by:</div>
8        <div class='col-md-4'>
9          <input type='text' />
10         </div>
11       </div>
12       <div class='row'>
13         <div class='col-md-6'>
14           <h3>Filtered by: </h3>
15         </div>
16       </div>
17     </div>
18   </div>
19 </div>
20
```

TS app.component.ts

product-list.component.html

```
1 <div class='panel panel-primary'>
2   <div class='panel-heading'>
3     Product List
4   </div>
5   <div class='panel-body'>
6     <div class='row'>
7       <div class='col-md-2'>Filter by:</div>
8       <div class='col-md-4'>
9         <input type='text' />
10      </div>
11    </div>
12    <div class='row'>
13      <div class='col-md-6'>
14        <h3>Filtered by: </h3>
15      </div>
16    </div>
17
18  </div>
19 </div>
20
```

Building the Component- Linked tribute

product-list.component.ts

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

@Component({
  selector: 'pm-products',
  templateUrl: './product-list.component.html'
})
export class ProductListComponent {
  pageTitle: string = 'Product List';
}
```

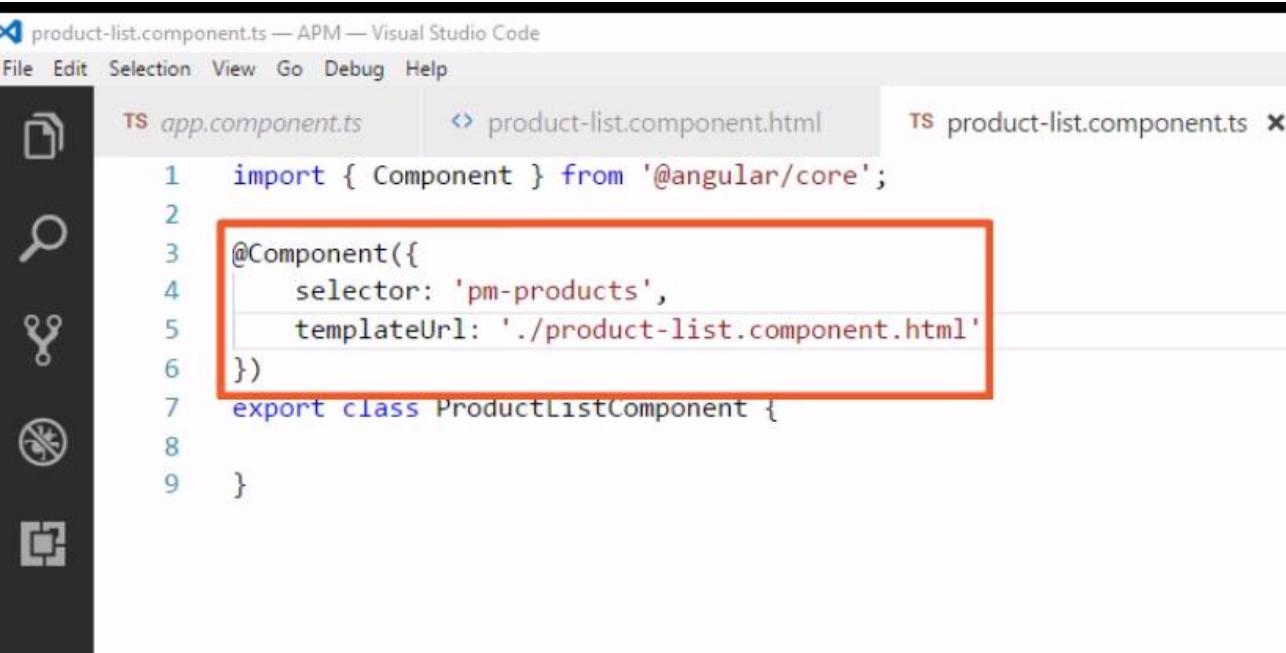
product-list.component.ts

The screenshot shows the VS Code interface with the following details:

- EXPLORER**: Shows the project structure:
 - OPEN EDITORS
 - APM
 - .vscode
 - e2e
 - node_modules
 - src
 - api
 - app
 - home
 - products
 - product-list.component.ts (selected)
 - product-list.component.html
 - shared
 - # app.component.css
 - app.component.html
 - app.component.spec.ts
 - app.component.ts
 - app.module.ts
 - assets
 - environments
 - favicon.ico
- TS app.component.ts**: Preview pane showing the component template.
- product-list.component.html**: Preview pane showing the component template.

```
13 <div class="col-md-6">
14 |   <h3>Filtered by: </h3>
15 |   </div>
16 </div>
17 <div class='table-responsive'>
18   <table class='table'>
19     <thead>
20       <tr>
21         <th>
22           <button class='btn btn-primary'>
23             Show Image
24           </button>
25         </th>
26         <th>Product</th>
27         <th>Code</th>
28         <th>Available</th>
29         <th>Price</th>
30         <th>5 Star Rating</th>
31       </tr>
32     </thead>
33     <tbody>
34
35   </tbody>
36 </table>
37 </div>
```

product-list.component.ts



```
product-list.component.ts — APM — Visual Studio Code
File Edit Selection View Go Debug Help
TS app.component.ts    product-list.component.html    TS product-list.component.ts ×
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'pm-products',
5   templateUrl: './product-list.component.html'
6 })
7 export class ProductListComponent {
8
9 }
```

Using a component as a directive

app.component.ts

```
@Component({  
  selector: 'pm-root',  
  template:  
    <div><h1>{{pageTitle}}</h1>  
      <div>My First Component</div>  
    </div>  
})  
export class AppComponent { }
```

product-list.component.ts

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

Contd.,

app.component.ts

```
@Component({  
  selector: 'pm-root',  
  template:  
    <div><h1>{{pageTitle}}</h1>  
      <pm-products></pm-products>  
    </div>  
})  
export class AppComponent { }
```

product-list.component.ts

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



Contd.,

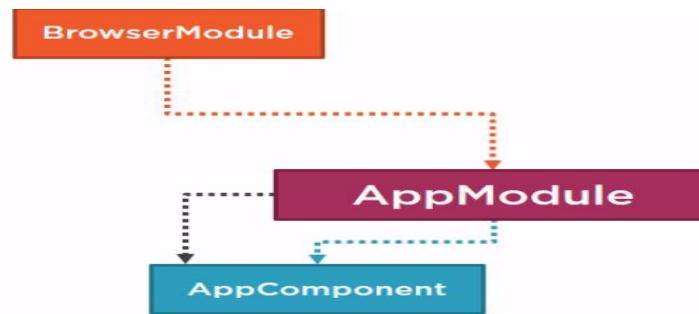
app.component.ts

```
1 @Component({  
  selector: 'pm-root',  
  template:  
    <div><h1>{{pageTitle}}</h1>  
    <pm-products></pm-products>  
  </div>  
})  
export class AppComponent { }
```

product-list.component.ts

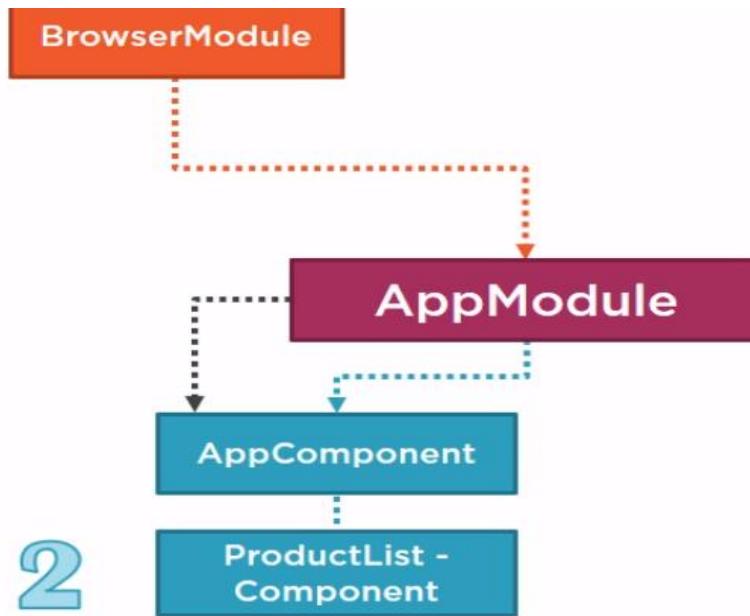
```
@Component({  
  selector: 'pm-products',  
  templateUrl:  
    './product-list.component.html'  
})  
export class ProductListComponent { }
```

Contd.,



- Imports
- Exports
- Declarations
- Providers
- Bootstrap

Contd.,



- Imports
- Exports
- Declarations
- Providers
- Bootstrap

File Edit Selection View Go Debug Help

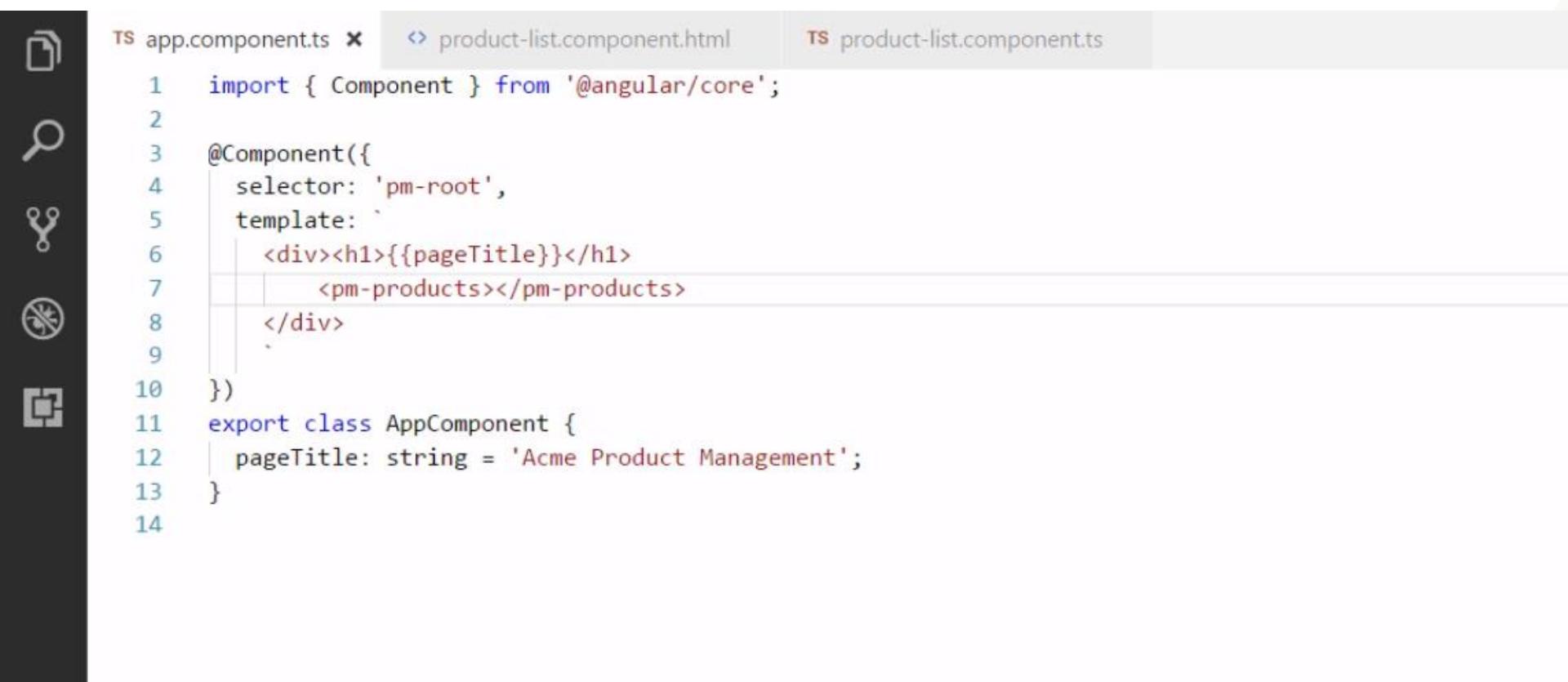
TS app.component.ts x

product-list.component.html

TS product-list.component.ts

```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'pm-root',
5   template: `
6     <div><h1>{{pageTitle}}</h1>
7     |   <div>My First Component</div>
8     </div>
9   `
10 })
11 export class AppComponent {
12   pageTitle: string = 'Acme Product Management';
13 }
14
```

Contd.,



The screenshot shows a code editor interface with two tabs open: `app.component.ts` and `product-list.component.html`. The left sidebar contains icons for file operations like new file, save, search, and refresh.

```
TS app.component.ts ✘  <> product-list.component.html  TS product-list.component.ts

1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'pm-root',
5   template: `
6     <div><h1>{{pageTitle}}</h1>
7       <pm-products></pm-products>
8     </div>
9   `
10 })
11 export class AppComponent {
12   pageTitle: string = 'Acme Product Management';
13 }
14
```

Contd.,

The screenshot shows a browser's developer tools open to the 'Console' tab. A red box highlights the first line of the error output, which reads: "'pm-products' is not a known element:'. Below this, a stack trace is provided:

```
'pm-products' is not a known element:
1. If 'pm-products' is an Angular component, then verify that it is part of this module.
2. If 'pm-products' is a Web Component then add 'CUSTOM_ELEMENTS_SCHEMA' to the '@NgModule.schemas' of this component to suppress this message. ("<div><h1>{pageTitle}</h1>
<[ERROR -]><pm-products></pm-products>
</div>
");
ng:///AppComponent.html@2:8
at syntaxError (http://localhost:4200/vendor.bundle.js:17606:34)
at TemplateParser.webpackJsonp.../.../compiler/@angular/compiler.es5.js.TemplateParser.parse (http://localhost:4200/vendor.bundle.js:28718:19)
at JitCompiler.webpackJsonp.../.../compiler/@angular/compiler.es5.js.JitCompiler._compileTemplate (http://localhost:4200/vendor.bundle.js:42763:39)
at http://localhost:4200/vendor.bundle.js:42683:62
at Set.forEach (native)
at JitCompiler.webpackJsonp.../.../compiler/@angular/compiler.es5.js.JitCompiler._compileComponents (http://localhost:4200/vendor.bundle.js:42683:19)
at http://localhost:4200/vendor.bundle.js:42570:19
at Object.then (http://localhost:4200/vendor.bundle.js:17596:156)
at JitCompiler.webpackJsonp.../.../compiler/@angular/compiler.es5.js.JitCompiler._compileModuleAndComponents (http://localhost:4200/vendor.bundle.js:42569:26)
at JitCompiler.webpackJsonp.../.../compiler/@angular/compiler.es5.js.JitCompiler.compileModuleAsync (http://localhost:4200/vendor.bundle.js:42498:37)
```

Adding Product-list component



```
TS app.component.ts      TS app.module.ts ✘  product-list.component.html      TS product-list.component.ts
1 import { BrowserModule } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3
4 import { AppComponent } from './app.component';
5
6 @NgModule({
7   declarations: [
8     AppComponent,
9     ProductListComponent
10  ],
11  imports: [
12    BrowserModule
13  ],
14  providers: [],
15  bootstrap: [AppComponent]
16 })
17 export class AppModule { }
```

Import Product-list Component

```
TS app.component.ts      TS app.module.ts ✘  ↵ product-list.component.html      TS product-list.component.ts

1 import { BrowserModule } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3
4 import { AppComponent } from './app.component';
5
6 @NgModule({
7   declarations: [
8     AppComponent,
9     ProductListComponent
10
11   ],
12   imports: [
13     BrowserModule
14   ],
15   providers: [],
16   bootstrap: [AppComponent]
17 })
18 export class AppModule { }
```

Import ProductListComponent from "app/products/product-list.component".

App.module.ts

TS app.component.ts TS app.module.ts ✘ ⏺ product-list.component.html TS product-list.component.ts

```
1 import { BrowserModule } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3
4 import { AppComponent } from './app.component';
5 import { ProductListComponent } from "app/products/product-list.component";
6
7 @NgModule({
8   declarations: [
9     AppComponent,
10    ProductListComponent
11   ],
12   imports: [
13     BrowserModule
14   ],
15   providers: [],
16   bootstrap: [AppComponent]
17 })
18 export class AppModule { }
```

App.module.ts

```
TS app.component.ts    TS app.module.ts ✘  ↗ product-list.component.html    TS product-list.component.ts

1 import { BrowserModule } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3
4 import { AppComponent } from './app.component';
5 import { ProductListComponent } from 'app/products/product-list.component';
6
7 @NgModule({
8   declarations: [
9     AppComponent,
10    ProductListComponent
11   ],
12   imports: [
13     BrowserModule
14   ],
15   providers: [],
16   bootstrap: [AppComponent]
17 })
18 export class AppModule { }
```

Acme Product Management X Deborah

localhost:4200

Acme Product Management

Product List

Filter by:

Filtered by:

Show Image

Product

Code

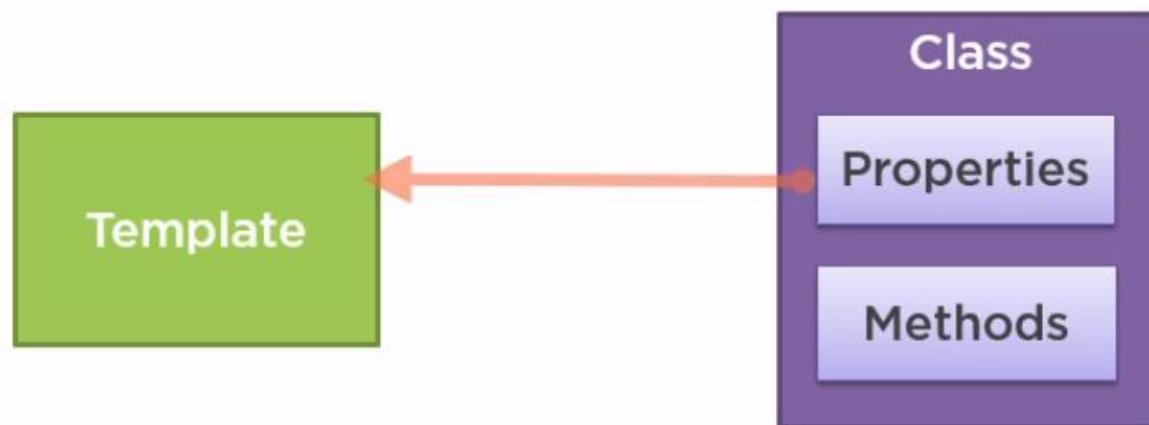
Available

Price

5 Star Rating

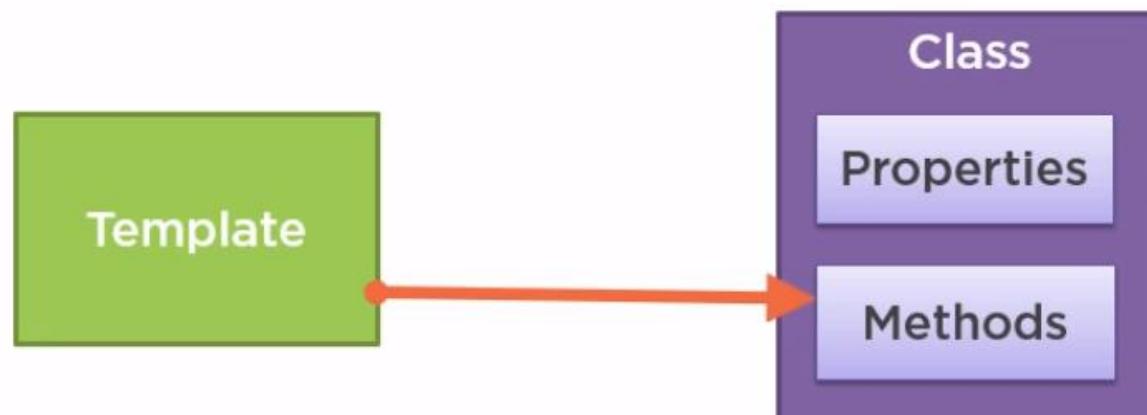
Binding

Coordinates communication between the component's class and its template and often involves passing data.



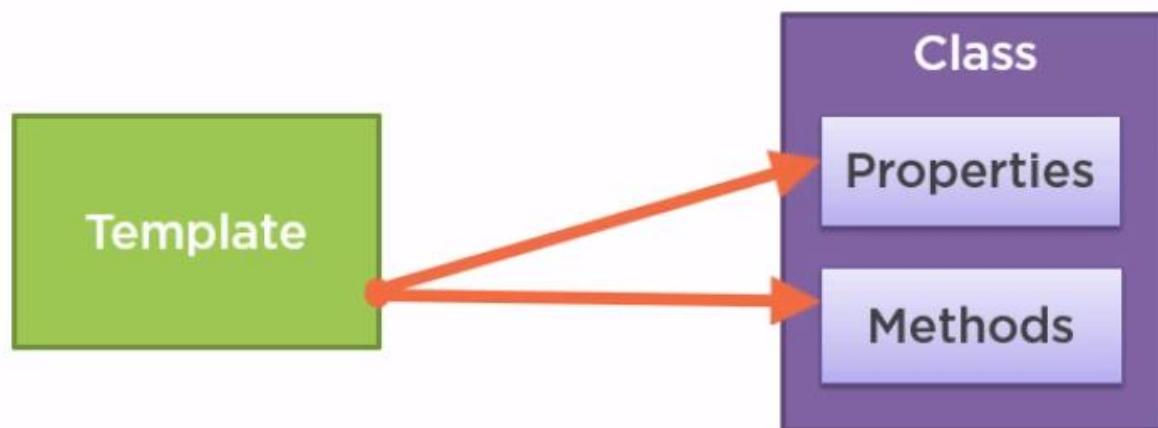
Contd.,

Coordinates communication between the component's class and its template and often involves passing data.



Contd.,

Coordinates communication between the component's class and its template and often involves passing data.



Interpolation

Template

```
<h1>{{pageTitle}}</h1>
```

Class

```
export class AppComponent {  
  pageTitle: string =  
    'Acme Product Management';  
}
```

Interpolation

Template

```
<h1>{{pageTitle}}</h1>
{{'Title: ' + pageTitle}}
{{2*20+1}}

{{'Title: ' + getTitle()}}
<h1 innerText="{{pageTitle}}></h1>
```

Class

```
export class AppComponent {
  pageTitle: string =
    'Acme Product Management';
  getTitle(): string {...};
}
```

Contd.,

Template

```
<h1>{{pageTitle}}</h1>
{{'T' + pageTitle}}
{{2*age+1}}
{{'Title: ' + getTitle()}}
<h1 innerText={{pageTitle}}></h1>
```

Template Expression

Class

```
export class AppComponent {
  pageTitle: string =
    'Acme Product Management';
  getTitle(): string {...};
}
```

Contd.,

Template

```
<h1>{{pageTitle}}</h1>  
{{'Title: ' + pageTitle}}  
{{2*20+1}}  
  
&gt;<h1 innerText={{pageTitle}}></h1>
```

Class

```
export class AppComponent {  
  pageTitle: string =  
    'Acme Product Management';  
  getTitle(): string {...};  
}
```

Contd.,

Template

```
<h1>{{pageTitle}}</h1>  
{{'Title: ' + pageTitle}}  
{2*20+1}  
  
<h1 innerText={{pageTitle}}></h1>
```

Class

```
export class AppComponent {  
  pageTitle: string =  
    'Acme Product Management';  
  getTitle(): string {...};  
}
```

Product-list.component.html

EXPLORER

OPEN EDITORS

APM

.vscode

e2e

node_modules

src

api

app

home

products

product-list.component.html

product-list.component.ts

shared

app.component.css

app.component.html

app.component.spec.ts

app.component.ts

app.module.ts

assets

environments

favicon.ico

index.html

main.ts

```
<> product-list.component.html ✘
```

```
1  <div class='panel panel-primary'>
2    <div class='panel-heading'>
3      Product List
4    </div>
5    <div class='panel-body'>
6      <div class='row'>
7        <div class='col-md-2'>Filter by:</div>
8        <div class='col-md-4'>
9          <input type='text' />
10         </div>
11       </div>
12       <div class='row'>
13         <div class='col-md-6'>
14           <h3>Filtered by: </h3>
15         </div>
16       </div>
17       <div class='table-responsive'>
18         <table class='table'>
19           <thead>
20             <tr>
21               <th>
22                 <button class='btn btn-primary'>
23                   Show Image
24                 </button>
25               </th>
26               <th>Product</th>
27             </tr>
```

```
product-list.component.html
```

```
1 <div class='panel panel-primary'>
2   <div class='panel-heading'>
3     {{pageTitle}}
4   </div>
5   <div class='panel-body'>
6     <div class='row'>
7       <div class='col-md-2'>Filter by:</div>
8       <div class='col-md-4'>
9         <input type='text' />
10      </div>
11    </div>
12    <div class='row'>
13      <div class='col-md-6'>
14        <h3>Filtered by: </h3>
15      </div>
16    </div>
17    <div class='table-responsive'>
18      <table class='table'>
19        <thead>
20          <tr>
21            <th>
22              <button class='btn btn-primary'>
23                Show Image
24              </button>
25            </th>
26            <th>Product</th>
```

```
product-list.component.ts
```

```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'pm-products',
5   templateUrl: './product-list.component.html'
6 })
7 export class ProductListComponent {
8   pageTitle: string = 'Product List';
9 }
```

Acme Product Management X Deborah

localhost:4200

Acme Product Management

Product List

Filter by:

Filtered by:

Show Image	Product	Code	Available	Price	5 Star Rating
------------	---------	------	-----------	-------	---------------

The image shows a developer's environment with a code editor and a web browser side-by-side.

Code Editor (product-list.component.ts):

```
TS product-list.component.ts x
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'pm-products',
5   templateUrl: './product-list.component.html'
6 })
7 export class ProductListComponent {
8   pageTitle: string = 'Product List!';
9 }
```

Browser (localhost:4200):

The browser displays the "Acme Product Management" application. The title bar says "localhost:4200". The main content area shows "Product List!" above a search bar labeled "Filter by:". Below the search bar, the text "Filtered by:" is followed by a row of buttons: "Show Image" (highlighted in blue), "Product", "Code", "Available", "Price", and "5 Star Rating". A red arrow points from the line of code `pageTitle: string = 'Product List!';` in the code editor to the "Product List!" text in the browser.

Directive

- Custom HTML element or attribute used to power up and extend our HTML

- Custom
- Built – In

Custom Directives

app.component.ts

```
@Component({  
  selector: 'pm-root',  
  template:  
    <div><h1>{{pageTitle}}</h1>  
      <pm-products></pm-products>  
    </div>  
  }  
}  
export class AppComponent { }
```

product-list.component.ts

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

Angular Build in Directives

Structural Directives



***ngIf: If logic**

***ngFor: For loops**

*ngif Built –in Directive

```
<div class='table-responsive'>
  <table class='table' *ngIf='products && products.length'>
    <thead> ...
    </thead>
    <tbody> ...
    </tbody>
  </table>
</div>
```

*ngIf Built-in Directive

```
<div class='table-responsive'>
  <table class='table' *ngIf='products && products.length'>
    <thead> . .
    </thead>
    <tbody> ...
    </tbody>
  </table>
</div>
```

```
...<!DOCTYPE html>
<html>
  <head>...
  </head>
  <body>
    <script id="__bs_script__" src="/browser-sync/browser-sync-client.2.11.2.js"></script>
    <pm-app>
      <div>
        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>
  </body>
</html>
```

Contd.,

```
<div class='table-responsive'>
<table class='table' *ngIf='products && products.length'>
<thead> ...
</thead>
<tbody> ...
</tbody>
</table>
</div>
```

```
...<!DOCTYPE html>
<html>
  <head>...
  </head>
  <body>
    <script id="__bs_script__" ...></script>
    <script async src="/browser-sync/browser-sync-client.2.11.2.js"></script>
    <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={}-->
                </div>
              ::after
            </div>
          </div>
        </pm-products>
      </div>
    </pm-app>
  </body>
</html>
```

product-list.component.html

```
1 <div class='panel panel-primary'>
2   <div class='panel-heading'>
3     {{pageTitle}}
4   </div>
5   <div class='panel-body'>
6     <div class='row'>
7       <div class='col-md-2'>Filter by:</div>
8       <div class='col-md-4'>
9         <input type='text' />
10      </div>
11    </div>
12    <div class='row'>
13      <div class='col-md-6'>
14        <h3>Filtered by: </h3>
15      </div>
16    </div>
17    <div class='table-responsive'>
18      <table class='table'>
19        <thead>
20          <tr>
21            <th>
22              <button class='btn btn-primary'>
23                Show Image
24              </button>
25            </th>
26            <th>Product</th>
27            <th>Code</th>
```

...

product-list.component.ts

```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'pm-products',
5   templateUrl: './product-list.component.html'
6 })
7 export class ProductListComponent {
8   pageTitle: string = 'Product List';
9   products: any[] = []
10 }
```

Ln 9, Col 20 Spaces: 4 UTF-8 CRLF TypeScript

EXPLORER

OPEN EDITORS

- APM
- .vscode
- e2e
- node_modules
- src**
- api**
- products
- { products.json }
- app
- home
- products
- product-list.com..
- product-list.com..
- shared
- # app.component.css
- app.component.h...
- app.components...
- app.component.ts
- app.module.ts
- assets

product-list.component.html

```

1 <div class='panel panel-primary'>
2   <div class='panel-heading'>
3     {{pageTitle}}
4   </div>
5   <div class='panel-body'>
6     <div class='row'>
7       <div class='col-md-2'>Filter by:</div>
8       <div class='col-md-4'>
9         <input type='text' />
10      </div>
11    </div>
12    <div class='row'>
13      <div class='col-md-6'>
14        <h3>Filtered by: </h3>
15      </div>
16    </div>
17    <div class='table-responsive'>
18      <table class='table'>
19        <thead>
20          <tr>
21            <th>
22              <button class='btn'>
23                Show Image
24              </button>
25            </th>
26          </tr>
27        </thead>
28      </table>
29    </div>
30  </div>

```

product-list.component.ts

```

5   })
6 }
7 export class ProductListComponent {
8   pageTitle: string = 'Product List';
9   products: any[] = [
10     {
11       "productId": 2,
12       "productName": "Garden Cart",
13       "productCode": "GDN-0023",
14       "releaseDate": "March 18, 2016",
15       "description": "15 gallon capacity",
16       "price": 32.99,
17       "starRating": 4.2,
18       "imageUrl": "http://openclipart.org"
19     },
20     {
21       "productId": 5,
22       "productName": "Hammer",
23       "productCode": "TBX-0048",
24       "releaseDate": "May 21, 2016",
25       "description": "Curved claw steel",
26       "price": 8.9,
27       "starRating": 4.8,
28       "imageUrl": "http://openclipart.org"
29     }
30   ]
31 }

```

```
< product-list.component.html >
```

```
16      </div>
17      <div class='table-responsive'>
18          <table class='table'>
19              <thead>
20                  <tr>
21                      <th>
22                          <button class='btn btn-primary'>
23                              Show Image
24                          </button>
25                      </th>
26                      <th>Product</th>
27                      <th>Code</th>
28                      <th>Available</th>
29                      <th>Price</th>
30                      <th>5 Star Rating</th>
31                  </tr>
32              </thead>
33              <tbody>
34
35                  </tbody>
36          </table>
37      </div>
38  </div>
39 </div>
```

```
... TS product-list.component.ts >
```

```
6      })
7  export class ProductListComponent {
8      pageTitle: string = 'Product List';
9      products: any[] = [
10      {
11          "productId": 2,
12          "productName": "Garden Cart",
13          "productCode": "GDN-0023",
14          "releaseDate": "March 18, 2016",
15          "description": "15 gallon capacity rolling g",
16          "price": 32.99,
17          "starRating": 4.2,
18          "imageUrl": "http://openclipart.org/image/300",
19      },
20      {
21          "productId": 5,
22          "productName": "Hammer",
23          "productCode": "TBX-0048",
24          "releaseDate": "May 21, 2016",
25          "description": "Curved claw steel hammer",
26          "price": 8.9,
27          "starRating": 4.8,
28          "imageUrl": "http://openclipart.org/image/300",
29      }
30  ]
```

The image shows a code editor with two tabs open: `product-list.component.html` and `product-list.component.ts`.

product-list.component.html

```
15      </div>
16    </div>
17  <div class='table-responsive'>
18    <table class='table'
19      *ngIf='products && products.length'>
20      <thead>
21        <tr>
22          <th>
23            <button class='btn btn-primary'>
24              Show Image
25            </button>
26          </th>
27          <th>Product</th>
28          <th>Code</th>
29          <th>Available</th>
30          <th>Price</th>
31          <th>5 Star Rating</th>
32        </tr>
33      </thead>
34      <tbody>
```

product-list.component.ts

```
6  })
7  export class ProductListComponent {
8    pageTitle: string = 'Product List';
9    products: any[] = [
10      {
11        "productId": 2,
12        "productName": "Garden Cart",
13        "productCode": "GDN-0023",
14        "releaseDate": "March 18, 2016",
15        "description": "15 gallon capacity rolling",
16        "price": 32.99,
17        "starRating": 4.2,
18        "imageUrl": "http://openclipart.org/image/240px_color_800x800px_garden%20cart.svg"
19      },
20      {
21        "productId": 5,
22        "productName": "Hammer",
23        "productCode": "TBX-0048",
24        "releaseDate": "May 21, 2016",
25        "description": "Curved claw steel hammer",
26        "price": 14.99,
27        "starRating": 4.8,
28        "imageUrl": "http://openclipart.org/image/240px_color_800x800px_hammer.svg"
29      }
30    ]
31  }
```

Acme Product Management

Product List

Filter by:

Filtered by:

Show Image	Product	Code	Available	Price	5 Star Rating

```
product-list.component.html x ... TS product-list.component.ts x
```

```
15      </div>
16    </div>
17    <div class='table-responsive'>
18      <table class='table'
19        *ngIf='products && products.length'>
20        <thead>
21          <tr>
22            <th>
23              <button class='btn btn-primary'>
24                Show Image
25              </button>
26            </th>
27            <th>Product</th>
28            <th>Code</th>
29            <th>Available</th>
30            <th>Price</th>
31            <th>5 Star Rating</th>
32          </tr>
33        </thead>
34        <tbody>
35
36          </tbody>
37        </table>
38      </div>
39    </div>
40  </div>
```

```
b   })
7   export class ProductListComponent {
8     pageTitle: string = 'Product List';
9     // products: any[] = [
10    // {
11      // "productId": 2,
12      // "productName": "Garden Cart",
13      // "productCode": "GDN-0023",
14      // "releaseDate": "March 18, 2016",
15      // "description": "15 gallon capacity rolling",
16      // "price": 32.99,
17      // "starRating": 4.2,
18      // "imageUrl": "http://openclipart.org/image"
19    },
20    // {
21      // "productId": 5,
22      // "productName": "Hammer",
23      // "productCode": "TBX-0048",
24      // "releaseDate": "May 21, 2016",
25      // "description": "Curved claw steel hammer",
26      // "price": 8.9,
27      // "starRating": 4.8,
28      // "imageUrl": "http://openclipart.org/image"
29    }
30  ];
31 }
```

Acme Product Management

Product List

Filter by:

Filtered by:

*ng for built in directive

```
<tr *ngFor='let product of products'>
  <td></td>
  <td>{{ product.productName }}</td>
  <td>{{ product.productCode }}</td>
  <td>{{ product.releaseDate }}</td>
  <td>{{ product.price }}</td>
  <td>{{ product.starRating }}</td>
</tr>
```

Table row creation

```
21          <tr>
22              <th>
23                  <button class='btn btn-primary'>
24                      Show Image
25                  </button>
26              </th>
27              <th>Product</th>
28              <th>Code</th>
29              <th>Available</th>
30              <th>Price</th>
31              <th>5 Star Rating</th>
32          </tr>
33      </thead>
34      <tbody>
35          </tbody>
36      </table>
37  </div>
38 </div>
39 ...
```

```
11      "productId": 2,
12      "productName": "Garden Cart",
13      "productCode": "GDN-0023",
14      "releaseDate": "March 18, 2016",
15      "description": "15 gallon capacity rolling g
16      "price": 32.99,
17      "starRating": 4.2,
18      "imageUrl": "http://openclipart.org/image/30
19  },
20  {
21      "productId": 5,
22      "productName": "Hammer",
23      "productCode": "TBX-0048",
24      "releaseDate": "May 21, 2016",
25      "description": "Curved claw steel hammer",
26      "price": 8.9,
27      "starRating": 4.8,
28      "imageUrl": "http://openclipart.org/image/30
29  }
30  ];
```

```
product-list.component.html x product-list.component.ts x
16      </div>
17      <div class='table-responsive'>
18          <table class='table'>
19              *ngIf='products && products.length'
20              <thead>
21                  <tr>
22                      <th>
23                          <button class='btn btn-primary'>
24                              Show Image
25                          </button>
26                      </th>
27                      <th>Product</th>
28                      <th>Code</th>
29                      <th>Available</th>
30                      <th>Price</th>
31                      <th>5 Star Rating</th>
32                  </tr>
33              </thead>
34              <tbody>
35                  <tr *ngFor='let product of products'>
36                      <td>[redacted]</td>
37                  </tr>
38              </tbody>
39          </table>
40      </div>
```

```
6      }
7      export class ProductListComponent {
8          pageTitle: string = 'Product List';
9          products: any[] = [
10              {
11                  "productId": 2,
12                  "productName": "Garden Cart",
13                  "productCode": "GDN-0023",
14                  "releaseDate": "March 18, 2016",
15                  "description": "15 gallon capacity rolling",
16                  "price": 32.99,
17                  "starRating": 4.2,
18                  "imageUrl": "http://openclipart.org/image/3"
19              },
20              {
21                  "productId": 5,
22                  "productName": "Hammer",
23                  "productCode": "TBX-0048",
24                  "releaseDate": "May 21, 2016",
25                  "description": "Curved claw steel hammer",
26                  "price": 8.9,
27                  "starRating": 4.8,
28                  "imageUrl": "http://openclipart.org/image/3"
29              }
30          ];
31      }
```

```
product-list.component.html x product-list.component.ts x ...
```

```
product-list.component.html
```

```
16     </div>
17     <div class='table-responsive'>
18         <table class='table'
19             *ngIf='products && products.length'>
20             <thead>
21                 <tr>
22                     <th>
23                         <button class='btn btn-primary'>
24                             Show Image
25                         </button>
26                     </th>
27                     <th>Product</th>
28                     <th>Code</th>
29                     <th>Available</th>
30                     <th>Price</th>
31                     <th>5 Star Rating</th>
32                 </tr>
33             </thead>
34             <tbody>
35                 <tr *ngFor='let product of products'>
36                     <td></td>
37                     <td>{{ product.productName }}</td>
38                 </tr>
39             </tbody>
40         </table>
```

```
product-list.component.ts
```

```
6     })
7     export class ProductListComponent {
8         pageTitle: string = 'Product List';
9         products: any[] = [
10             {
11                 "productId": 2,
12                 "productName": "Garden Cart",
13                 "productCode": "GDN-0023",
14                 "releaseDate": "March 18, 2016",
15                 "description": "15 gallon capacity rolling garden cart",
16                 "price": 32.99,
17                 "starRating": 4.2,
18                 "imageUrl": "http://openclipart.org/image/300px/svg_to_png/140/garden-cart.png"
19             },
20             {
21                 "productId": 5,
22                 "productName": "Hammer",
23                 "productCode": "TBX-0048",
24                 "releaseDate": "May 21, 2016",
25                 "description": "Curved claw steel hammer",
26                 "price": 8.9,
27                 "starRating": 4.8,
28                 "imageUrl": "http://openclipart.org/image/300px/svg_to_png/140/hammer-curved-claw-hammer.png"
29             }
30         ];
31     }
```

product-list.component.html

```
18     <table class='table'  
19         *ngIf='products && products.length'>  
20         <thead>  
21             <tr>  
22                 <th>  
23                     <button class='btn btn-primary'>  
24                         Show Image  
25                     </button>  
26                 </th>  
27                 <th>Product</th>  
28                 <th>Code</th>  
29                 <th>Available</th>  
30                 <th>Price</th>  
31                 <th>5 Star Rating</th>  
32             </tr>  
33         </thead>  
34         <tbody>  
35             <tr *ngFor='let product of products'>  
36                 <td></td>  
37                 <td>{{ product.productName }}</td>  
38                 <td>{{ product.productCode }}</td>  
39                 <td>{{ product.releaseDate }}</td>  
40                 <td>{{ product.price }}</td>  
41                 <td>{{ product.starRating }}</td>  
42             </tr>
```

product-list.component.ts

```
6     })  
7     export class ProductListComponent {  
8         pageTitle: string = 'Product List';  
9         products: any[] = [  
10             {  
11                 "productId": 2,  
12                 "productName": "Garden Cart",  
13                 "productCode": "GDN-0023",  
14                 "releaseDate": "March 18, 2016",  
15                 "description": "15 gallon capacity rolling",  
16                 "price": 32.99,  
17                 "starRating": 4.2,  
18                 "imageUrl": "http://openclipart.org/i",  
19             },  
20             {  
21                 "productId": 5,  
22                 "productName": "Hammer",  
23                 "productCode": "TBX-0048",  
24                 "releaseDate": "May 21, 2016",  
25                 "description": "Curved claw steel hammer",  
26                 "price": 8.9,  
27                 "starRating": 4.8,  
28                 "imageUrl": "http://openclipart.org/i",  
29             }  
30         ];  
31     }
```

Acme Product Management

Product List

Filter by:

Filtered by:

Show Image	Product	Code	Available	Price	5 Star Rating
	Garden Cart	GDN-0023	March 18, 2016	32.99	4.2
	Hammer	TBX-0048	May 21, 2016	8.9	4.8

Acme Product Management

Product List

Filter by:

Filtered by:

Show Image	Product	Code	Available	Price	5 Star Rating
	Garden Cart	GDN-0023	March 18, 2016	32.99	4.2
	Hammer	TBX-0048	May 21, 2016	8.9	4.8

for ...of vs fr ... in

for...of

- Iterates over iterable objects, such as an array.
- Result: di, boo, punkeye

```
let nicknames= ['di', 'boo', 'punkeye'];

for (let nickname of nicknames) {
  console.log(nickname);
}
```

for...in

- Iterates over the properties of an object.
- Result: 0, 1, 2

```
let nicknames= ['di', 'boo', 'punkeye'];

for (let nickname in nicknames) {
  console.log(nickname);
}
```

Component as a Directive



Inline template

- For short templates
- Specify the **template** property
- Use the ES 2015 back ticks for multiple lines
- Watch syntax

Linked template

- For longer templates
- Specify the **templateUrl** property
- Define the path to the HTML file

product-list.component.ts

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

app.component.ts

```
1 @Component({  
  selector: 'pm-root',  
  template:  
    <div><h1>{{pageTitle}}</h1>  
      <pm-products></pm-products>  
    </div>  
})  
export class AppComponent { }
```

app.module.ts

```
2 @NgModule({  
  imports: [ BrowserModule ],  
  declarations: [  
    AppComponent,  
    ProductListComponent ],  
  bootstrap: [ AppComponent ]  
})  
export class AppModule { }
```

Interpolation



One way binding

- From component class property to an element property.

Defined with double curly braces

- Contains a template expression
- No quote needed

Structural Directives



*ngIf and *ngFor

- Prefix with an asterisk
- Assign to a quoted string expression

*ngIf

- Expression is evaluated as a true or false value

*ngFor

- Define the local variable with let
- Specify 'of': 'let product of products'

Data Binding

Data Binding

- Property Binding
- Handling events with Event Binding
- Handling input with two way Binding
- Transforming data with pipes

Property Binding

```
<img [src]='product.imageUrl'>
```

Element
Property

Template
Expression

```
<img [src]='product.imageUrl'>
```

[]
Binding Target

''
Binding Source

Contd.,

```
<img [src]='product.imageUrl'>
```

```
<img src={{product.imageUrl}}>
```

```
<img src='http://openclipart.org/{{product.imageUrl}}'>
```

Contd.,

```
</th>
<th>Product</th>
<th>Code</th>
<th>Available</th>
<th>Price</th>
<th>5 Star Rating</th>
</tr>
</thead>
<tbody>
  <tr *ngFor='let product of products'>
    <td>
      <img [src]='product.imageUrl'>
    </td>
    <td>
      {{ product.productName }}  

      {{ product.price | currency }}  

      {{ product.starRating }}  

      {{ product.description }}  

      {{ product.releaseDate }}  

      {{ product.productId }}  

      {{ product.productCode }}  

      {{ product.imageUrl }}  

    </td>
  </tr>
</tbody>
```

The code shows an Angular template for a product table. It includes an `thead` section with six columns: Product, Code, Available, Price, and 5 Star Rating. The `tbody` section uses an `ngFor` loop to iterate over an array of `products`. Each row contains two `td` elements. The first `td` displays the product's name, price, star rating, and description. The second `td` displays the product's ID, code, and URL. An `img` tag with `[src]` bound to `product.imageUrl` is also present in the first `td`.

```
<td>
  <img [src]='product.imageUrl'
       [title]='product.productName'>
</td>
```

Product List

Filter by:

Filtered by:

Show Image	Product	Code	Available	Price	5 Star Rating
 A grey wheelbarrow icon with a black wheel and a handle.	Garden Cart	GDN-0023	March 18, 2016	32.99	4.2
 A blue and silver hammer icon.	Hammer	TBX-0048	May 21, 2016	8.9	4.8

```
        Show Image
    </button>
</th>
<th>Product</th>
<th>Code</th>
<th>Available</th>
<th>Price</th>
<th>5 Star Rating</th>
</tr>
</thead>
<tbody>
    <tr *ngFor='let product of products'>
        <td>
            <img [src]='product.imageUrl'
                  [title]='product.productName'
                  [style.width.px]='imageWidth'>
        </td>
```

```
2
3     @Component({
4         selector: 'pm-products',
5         templateUrl: './product-list.component.html'
6     })
7     export class ProductListComponent {
8         pageTitle: string = 'Product List';
9         imageWidth: number = 50;
10        imageMargin: number = 2;
11        products: any[] = [
12            {
13                "productId": 2,
14                "productName": "Garden Cart",
15                "productCode": "GDN-0023",
16                "releaseDate": "March 18, 2016",
17                "description": "15 gallon capacity rol",
18                "price": 32.99,
```

```

        Show Image
    </button>
</th>
<th>Product</th>
<th>Code</th>
<th>Available</th>
<th>Price</th>
<th>5 Star Rating</th>
</tr>
</thead>
<tbody>
    <tr *ngFor='let product of products'>
        <td>
            <img [src]='product.imageUrl'
                [title]='product.productName'
                [style.width.px]='imageWidth'
                [style.margin.px]='imageMargin'>
        </td>

```

```

2
3     @Component({
4         selector: 'pm-products',
5         templateUrl: './product-list.component.ht
6     })
7     export class ProductListComponent {
8         pageTitle: string = 'Product List';
9         imageWidth: number = 50;
10        imageMargin: number = 2;
11        products: any[] = [
12        {
13            "productId": 2,
14            "productName": "Garden Cart",
15            "productCode": "GDN-0023",
16            "releaseDate": "March 18, 2016",
17            "description": "15 gallon capacit
18            "price": 32.99,
19            "starRating": 4.2,

```

Acme Product Management

Product List

Filter by:

Filtered by:

Show Image	Product	Code	Available	Price	5 Star Rating
	Garden Cart	GDN-0023	March 18, 2016	32.99	4.2
	Hammer	TBX-0048	May 21, 2016	8.9	4.8

Template

```
<h1>{{pageTitle}}</h1>
<img [src]='product.imageUrl'>
```

Class

```
export classListComponent {
  pageTitle: string = 'Product List';
  products: any[] = [...];
}
```

Event Binding

Template

Class

```
<h1>{{pageTitle}}</h1>  
<img [src]='product.imageUrl'>  
<button (click)='toggleImage()'>
```

```
export classListComponent {  
  pageTitle: string = 'Product List';  
  products: any[] = [...];
```

()

Target Event

"

Template Statement

Contd.,

Template

```
<h1>{{pageTitle}}</h1>
<img [src]='product.imageUrl'>
<button (click)='toggleImage()'>
```

Class

```
export classListComponent {
  pageTitle: string = 'Product List';
  products: any[] = [...];
  toggleImage(): void {...}
}
```



```
<tr>
<th>
  <button class='btn btn-primary'>
    Show Image
  </button>
</th>
<th>Product</th>
<th>Code</th>
<th>Available</th>
<th>Price</th>
<th>5 Star Rating</th>
</tr>
</thead>
```

```
2
3 @Component({
4   selector: 'pm-products',
5   templateUrl: './product-list.component.html'
6 })
7 export class ProductListComponent {
8   pageTitle: string = 'Product List';
9   imageWidth: number = 50;
10  imageMargin: number = 2;
11  products: any[] = [
12    {
13      "productId": 2,
14      "productName": "Garden Cart",
```

```

    ...
export class ProductListComponent {
  pageTitle: string = 'Product List';
  imageWidth: number = 50;
  imageMargin: number = 2;
  showImage: boolean = false;
  products: any[] = [
    {

<tr>
  <th>
    <button class='btn btn-primary'
      (click)='toggleImage()'
      Show Image
    </button>
  </th>
  <th>Product</th>
  <th>Code</th>
  <th>Available</th>
  <th>Price</th>
  <th>5 Star Rating</th>
</tr>
thead>
body>
  <tr *ngFor='let product of products'>
    <td>
      22
      23
      },
      24
      {
      "productId": 5,
      "productName": "Hammer",
      "productCode": "TBX-0048",
      "releaseDate": "May 21, 2016",
      "description": "Curved claw steel hammer",
      "price": 8.9,
      "starRating": 4.8,
      "imageUrl": "http://openclipart.org/image/"
      25
      26
      27
      28
      29
      30
      31
      32
      }
      33
      ];
      34
      35
      toggleImage(): void {
      36
        this.showImage = !this.showImage;
      37
      }
      38
      39
    
```

```

<th>
  <button class='btn btn-primary'
    (click)='toggleImage()'
    Show Image
  </button>
</th>
<th>Product</th>
<th>Code</th>
<th>Available</th>
<th>Price</th>
<th>5 Star Rating</th>
</tr>
</thead>
<tbody>
  <tr *ngFor='let product of products'>
    <td>
      <img *ngIf='showImage'
        [src]='product.imageUrl'
        [title]='product.productName'
        [style.width.px]='imageWidth'
        [style.margin.px]='imageMargin'>

```

```

23           {
24             "productId": 5,
25             "productName": "Hammer",
26             "productCode": "TBX-0048",
27             "releaseDate": "May 21, 2016",
28             "description": "Curved claw steel hammer",
29             "price": 8.9,
30             "starRating": 4.8,
31             "imageUrl": "http://openclipart.org/image/3
32           }
33         ];
34
35       toggleImage(): void {
36         this.showImage = !this.showImage;
37     }
38   }
39

```

Acme Product Management

Product List

Filter by:

Filtered by:

Show Image	Product	Code	Available	Price	5 Star Rating
	Garden Cart	GDN-0023	March 18, 2016	32.99	4.2
	Hammer	TBX-0048	May 21, 2016	8.9	4.8

Adding image text

```
<th>
  <button class='btn btn-primary'
    (click)='toggleImage()'
    {{}} Show Image
  </button>
</th>
<th>Product</th>
<th>Code</th>
<th>Available</th>
<th>Price</th>
<th>5 Star Rating</th>
/tr>
ad>
y>
tr *ngFor='let product of products'>
  ...
```

```
23   {
24     "productId": 5,
25     "productName": "Hammer",
26     "productCode": "TBX-0048",
27     "releaseDate": "May 21, 2016",
28     "description": "Curved claw steel",
29     "price": 8.9,
30     "starRating": 4.8,
31     "imageUrl": "http://openclipart.or
32   }
33 ];
34
35   toggleImage(): void {
36     this.showImage = !this.showImage;
37   }
38 }
```

```
<thead>
  <tr>
    <th>
      <button class='btn btn-primary'
        (click)='toggleImage()'
        {{showImage ? 'Hide'}} Show Image
      </button>
    </th>
    <th>Product</th>
    <th>Code</th>
    <th>Available</th>
    <th>Price</th>
    <th>5 Star Rating</th>
  </tr>
</thead>
<tbody>
  <tr *ngFor='let product of products'>
    <td>
```

```
21           imageUrl: "http://openclipart.org
22       },
23   {
24     "productId": 5,
25     "productName": "Hammer",
26     "productCode": "TBX-0048",
27     "releaseDate": "May 21, 2016",
28     "description": "Curved claw steel ha
29     "price": 8.9,
30     "starRating": 4.8,
31     "imageUrl": "http://openclipart.org
32   }
33 ];
34
35   toggleImage(): void {
36     this.showImage = !this.showImage;
37   }
38 }
```

```
<th>
  <button class='btn btn-primary'
    (click)='toggleImage()'
    {{showImage ? 'Hide' : 'Show'}} Image
  </button>
</th>
<th>Product</th>
<th>Code</th>
<th>Available</th>
<th>Price</th>
<th>5 Star Rating</th>
</tr>
```

Filtered by:

Hide Image	Product	Code	Available	Price	5 Star Rating
	Garden Cart	GDN-0023	March 18, 2016	32.99	4.2
	Hammer	TBX-0048	May 21, 2016	8.9	4.8

Two way Binding – Filter by

Template

```
<input [(ngModel)]='listFilter'>
```

Class

```
export class ListComponent {  
  listFilter: string = 'cart';  
}
```

Template

```
<input [(ngModel)]='listFilter'>
```

Class

```
export class ListComponent {  
  listFilter: string = 'cart';  
}
```

Two way Binding

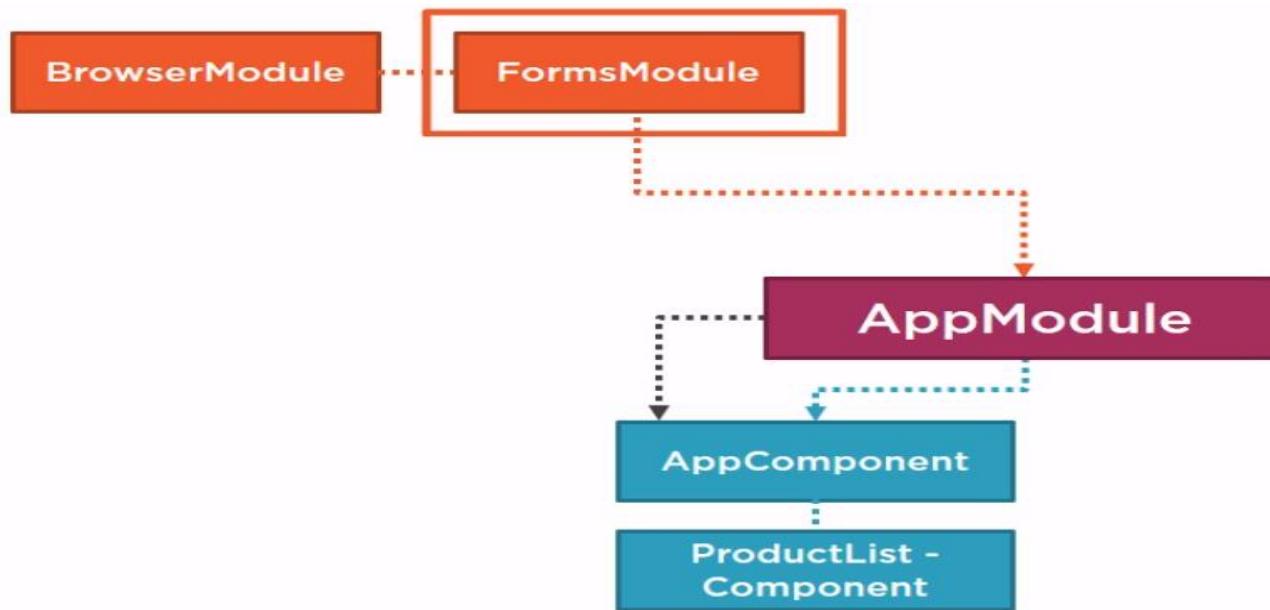
Template

```
<input [(ngModel)]='listFilter'>
```

Class

```
export class ListComponent {  
  listFilter: string = 'cart';  
}
```

[0]
Banana in a Box



- Imports
- Exports
- Declarations
- Providers
- Bootstrap

```
x product-list.component.ts app\products
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'pm-products',
5   templateUrl: './product-list.component.html'
6
7 })
8 export class ProductListComponent {
9   pageTitle: string = 'Product List';
10  imageWidth: number = 50;
11  imageMargin: number = 2;
12  showImage: boolean = false;
13  listFilter: string = 'cart'; listFilter: string = 'cart';
14  products: any[] = [
15    {
16      "productId": 2,
17      "productName": "Garden Cart",
18      "productCode": "GDN-0023",
```

&& produc

```
<div class='panel panel-primary'>
  <div class='panel-heading'>
    {{pageTitle}}
  </div>
  <div class='panel-body'>
    <div class='row'>
      <div class='col-md-2'>Filter by:</div>
      <div class='col-md-4'>
        <input type='text'
          [(ngModel)]='listFilter' />
      </div>
    </div>
    <div class='row'>
      <div class='col-md-6'>
        <h3>Filtered by: {{listFilter}} </h3>
      </div>
    </div>
  </div>
```

```
1   import { Component } from '@angular/core';
2
3   @Component({
4     selector: 'pm-products',
5     templateUrl: './product-list.component.ht
6
7   })
8   export class ProductListComponent {
9     pageTitle: string = 'Product List';
10    imageWidth: number = 50;
11    imageMargin: number = 2;
12    showImage: boolean = false;
13    listFilter: string = 'cart';
14    products: any[] = [
15      {
16        "productId": 2,
17        "productName": "Garden Cart",
18        "imagePath": "http://www.angularindepth.com/images/garden-cart.jpg"
19      }
20    ];
21
22    constructor() { }
23
24    ngOnInit() {
25      this.filteredProducts = this.products;
26    }
27
28    applyFilter(filterValue: string) {
29      filterValue = filterValue.trim();
30      filterValue = filterValue.toLowerCase();
31      this.filteredProducts = this.products.filter(product =>
32        product.productName.toLowerCase().indexOf(filterValue) === 0);
33    }
34  }
35
```

✖ top ▼ Preserve log

```
Can't bind to 'ngModel' since it isn't a known property of 'input'. ("'col-md-2'>Filter by:</div>
<div class='col-md-4'>
    <input type='text' [ERROR ->][(ngModel)]=listFilter' />
</div>
</div>
"): ProductListComponent@8:35 ; Zone: <root> ; Task: Promise.then ; Value: Error: Template parse error
Can't bind to 'ngModel' since it isn't a known property of 'input'. ("'col-md-2'>Filter by:</div>
<div class='col-md-4'>
    <input type='text' [ERROR ->][(ngModel)]=listFilter' />
</div>
</div>
```

```
1 import { BrowserModule } from '@angular/platform-browser'
2 import { NgModule } from '@angular/core';
3 import { FormsModule } from '@angular/forms';
4
5 import { AppComponent } from './app.component';
6 import { ProductListComponent } from './products/product-
7
8 @NgModule({
9   declarations: [
10     AppComponent,
11     ProductListComponent
12   ],
13   imports: [
14     BrowserModule,
15     FormsModule
16   ],
17   providers: [],
18   bootstrap: [AppComponent]
19 })
```

Two way Binding

Product List

Filter by:

Filtered by: cart

Show Image

Product

Code

Available

Price

5 Star Rating

Garden Cart

GDN-0023

March 18, 2016

32.99

4.2

Hammer

TBX-0048

May 21, 2016

8.9

4.8

Product List

Filter by:

Filtered by: carts

Show Image

Product

Code

Available

Price

5 Star Rating

Garden Cart

GDN-0023

March 18, 2016

32.99

4.2

Hammer

TBX-0048

May 21, 2016

8.9

4.8

Filter by:

Filtered by: carts

Show Image	Product	Code	Available	Price	5 Star Rating
	Garden Cart	GDN-0023	March 18, 2016	32.99	4.2
	Hammer	TBX-0048	May 21, 2016	8.9	4.8

Transforming Data with Pipes

Transform
bound
properties
before
display

Built-in pipes

- date
- number, decimal, percent, currency
- json, slice
- etc

Custom
pipes

Pipe Examples

```
 {{ product.productCode | lowercase }}  
  
<img [src]='product.imageUrl'  
      [title]='product.productName | uppercase'>  
  
 {{ product.price | currency | lowercase }}  
  
 {{ product.price | currency:'USD':true:'1.2-2' }}
```

```
<th>Code</th>
<th>Available</th>
<th>Price</th>
<th>5 Star Rating</th>
</tr>
</thead>
<tbody>
<tr *ngFor='let product of products'>
<td>
    <img *ngIf='showImage'
        [src]='product.imageUrl'
        [title]='product.productName'
        [style.width.px]='imageWidth'
        [style.margin.px]='imageMargin'>
</td>
<td>{{ product.productName }}</td>
<td>{{ product.productCode | lowercase }}</td>
<td>{{ product.releaseDate }}</td>
<td>{{ product.price | currency:'USD':true:'1.2-2' }}</td>
<td>{{ product.starRating }}</td>
</tr>
</tbody>
</table>
</div>
</div>
</div>
```

Filter by:

Filtered by: cart

Show Image	Product	Code	Available	Price	5 Star Rating
	Garden Cart	gdn-0023	March 18, 2016	\$32.99	4.2
	Hammer	tbx-0048	May 21, 2016	\$8.90	4.8

Data Binding

DOM

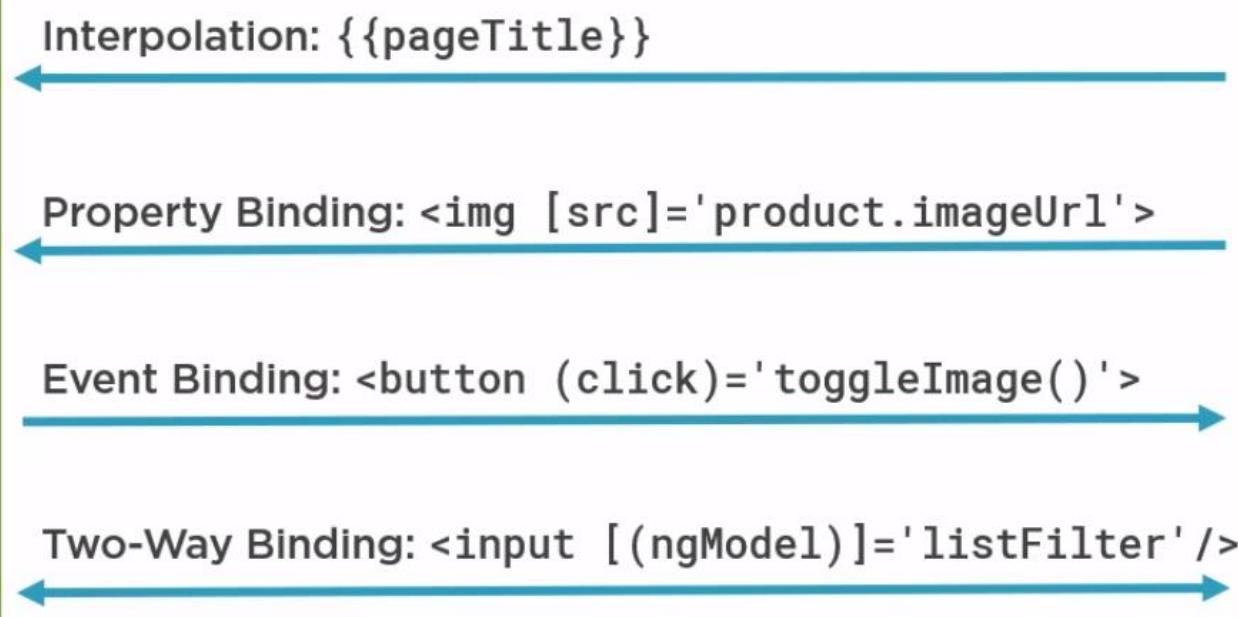
```
▼<pm-app>
  ▼<div>
    <h1>Acme Product Management</h1>
  ▼<pm-products>
    ▼<div class="panel panel-primary">
      <div class="panel-heading">
        Product List
      </div>
      <div class="panel-body">
        ::before
      ▶<div class="row">...</div>
      ▶<div class="row">...</div>
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

product-list.component.ts

```
@Component({
  selector: 'pm-products',
  templateUrl: './product-list.component.html'
})
export class ProductListComponent {
  pageTitle: string = 'Product List';
  listFilter: string = 'cart';
  products: any[] = [...];
  toggleImage(): void {...}
}
```



Data Binding



DOM

Component

Ngmodel directive

product-list.component.html

```
<div class='col-md-4'>
  <input type='text'
    [(ngModel)]='listFilter' />
</div>
```

app.module.ts

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

Pipes



Pipe character |

Pipe name

Pipe parameters

- Separated with colons

Example

- {{ product.price | currency:'USD':true:'1.2-2' }}

More on Components

- Strong typing & Interfaces
- Encapsulating Styles
- Lifecycle hooks
- Custom pipes
- Nested Components

More on Components

- Defining an Interface
- Encapsulating Component Styles
- Using lifecycle hooks
- Building a custom pipe

Strong typing

```
export class ProductListComponent {
  pageTitle: string = 'Product List';
  showImage: boolean = false;
  listFilter: string = 'cart';
  message: string;

  products: any[] = [...];

  toggleImage(): void {
    this.showImage = !this.showImage;
  }

  onRatingClicked(message: string): void {
    this.message = message;
  }
}
```

Interface

A **specification** identifying a related set of properties and methods.

A class commits to supporting the specification by **implementing** the interface.

Use the interface as a **data type**.

Development time only!

Interface Is a Specification

```
export interface IProduct {  
    productId: number;  
    productName: string;  
    productCode: string;  
    releaseDate: Date;  
    price: number;  
    description: string;  
    starRating: number;  
    imageUrl: string;  
    calculateDiscount(percent: number): number;  
}
```

export
keyword

Interface
Name

interface
keyword

Using Interface as a Datatype

```
import { IProduct } from './product';

export class ProductListComponent {
  pageTitle: string = 'Product List';
  showImage: boolean = false;
  listFilter: string = 'cart';

  products: IProduct[] = [...];

  toggleImage(): void {
    this.showImage = !this.showImage;
  }
}
```

TS product-list.component.ts ✘

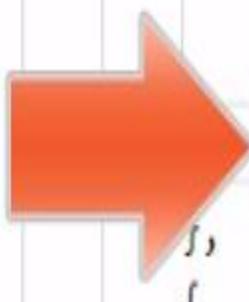
```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'pm-products',
5   templateUrl: './product-list.component.html'
6 })
7 export class ProductListComponent {
8   pageTitle: string = 'Product List';
9   imageWidth: number = 50;
10  imageMargin: number = 2;
11  showImage: boolean = false;
12  listFilter: string = 'cart';
13  products: any[] = [
14    {
15      "productId": 2,
16      "productName": "Garden Cart",
17      "productCode": "GDN-0023",
18      "releaseDate": "March 18, 2016",
19      "description": "15 gallon capacity rolling garden cart",
20      "price": 32.99,
```

Product.ts

```
product-list.component.ts          product.ts      ×  
1  export interface IProduct {  
2      productId: number;  
3      productName: string;  
4      productCode: string;  
5      releaseDate: string;  
6      price: number;  
7      description: string;  
8      starRating: number;  
9      imageUrl: string;  
10 }
```

Introducing Topographical error

```
15      "productId": 2,  
16      "productName": "Garden Cart",  
17      "productCode": "GDN-0023",  
18      "releaseDate": "March 18, 2016",  
19      "description": "15 gallon capacity rolling garden cart",  
20      "price": 32.99,  
21      "starRating": 4.2,  
22      "imageUrl": "http://openclipart.org/image/300px/svg\_to\_png/58"  
23      ,  
24      {
```



Filtered by: cart

Hide Image	Product	Code	Available	Price	5 Star Rating
	Garden Cart	gdn-0023	March 18, 2016	\$32.99	4.2
	Hammer	tbx-0048	May 21, 2016	\$8.90	4.8

```
8   pageTitle: string = 'Product List';
9   imageWidth: number = 50;
10  imageMargin: number = 2;
11  showImage: boolean = false;
12  listFilter: string = 'cart';
13  products: IProduct[] = [
  Import IProduct from "app/products/product".
    14  [
    15    {
    16      "productName": "Garden Cart",
    17      "productCode": "GDN-0023",

```

```
    })
8  export class ProductListComponent {
9    pageTitle: string = 'Product List';
10   imageWidth: number = 50;
11   imageMargin: number = 2;
12   showImage: boolean = false;
13   listFilter: string = 'cart';
14   products: IProduct[] = [
15     {
16       "productId": 2,
17       "productName": "Garden Cart",
18       "productCode": "GDN-0023",
19       "releaseDate": "March 18, 2016",
20       "description": "15 gallon capacity rolling garden cart",
21       "price": 32.99,
22       "starRating": 4.2,
23       "imageUrls": "http://openclipart.org/image/300px/svg\_to\_png/58
```



```
--  
13     listFilter: string = 'cart';  
14     products: IProduct[] = [  
15         {  
16             "productId": 2,  
17             "productName": "Garden Cart",  
18             "productCode": "GDN-0023",  
19             "releaseDate": "March 18, 2016",  
20             "description": "15 gallon capacity rolling garden cart",  
21             "price": 32.99,  
22             "starRating": 4.2,  
23             "imageUrl": "http://openclipart.org/image/300px/svg_to_png/584  
24         },        
25     {
```

```
3     productName: string;
4     productCode: string;
5     releaseDate: string;
6     price: number;
7     description: string;
8     starRating: number;
9     imageUrl: string;
10    }
11
12   export class Product implements IProduct {
13
14     constructor(public productId: number,
15                 public productName: string,
16                 public productCode: string,
17                 public releaseDate: string,
18                 public price: number,
19                 public description: string,
20                 public starRating: number,
21                 public imageUrl: string) {
22     }
23
24     calculateDiscount(percent: number): number {
25       return this.price - (this.price * percent / 100);
26     }
27   }
28 }
```

product.ts

x

```
1  export interface IProduct {  
2      productId: number;  
3      productName: string;  
4      productCode: string;  
5      releaseDate: string;  
6      price: number;  
7      description: string;  
8      starRating: number;  
9      imageUrl: string;  
10 }
```

Handling Unique Component Styles



Templates sometimes require unique styles

We can inline the styles directly into the HTML

We can build an external stylesheet and link it in index.html

There is a better way!

Encapsulating Component Styles

styles

```
@Component({  
  selector: 'pm-products',  
  templateUrl: './product-list.component.html',  
  styles: ['thead {color: #337AB7;}'])})
```

styleUrls

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

Filtered by: cart

Show Image

Product	Code	Available	Price	5 Star Rating
Garden Cart	gdn-0023	March 18, 2016	\$32.99	4.2
Hammer	tbx-0048	May 21, 2016	\$8.90	4.8

Product-list.component.css

```
▶ app
  ▶ home
  ▶ products
    # product-list.component.css
    <> product-list.component.html
    TS product-list.component.ts
    TS product.ts
  ▶ shared
  # app.component.css
  <> app.component.html
```

The screenshot shows a code editor interface with two tabs: 'product-list.component.ts' and '# product-list.component.css'. The 'product-list.component.ts' tab is active, displaying the following TypeScript code:

```
1  thead {
2    color: #337AB7
3  }
```

Product-list.component.ts

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

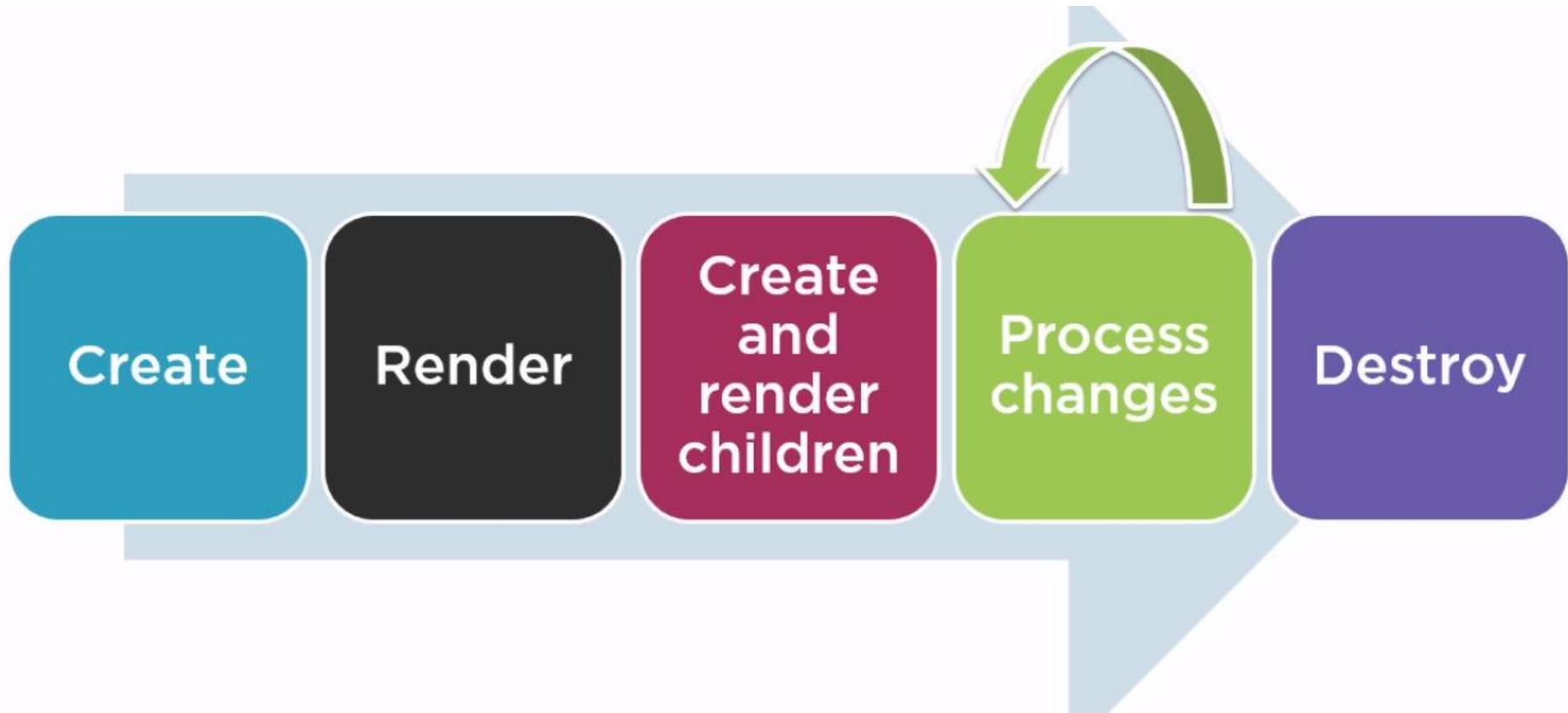
@Component({
  selector: 'pm-products',
  templateUrl: './product-list.component.html',
  styleUrls: ['./product-list.component.css']
})
export class ProductListComponent {
  pageTitle: string = 'Product List';
  imageWidth: number = 50;
  imageMargin: number = 2;
  showImage: boolean = false;
  listFilter: string = 'cart';
  products: IProduct[] = [
```

Filtered by: cart

Show Image

Product	Code	Available	Price	5 Star Rating
Garden Cart	gdn-0023	March 18, 2016	\$32.99	4.2
Hammer	tbx-0048	May 21, 2016	\$8.90	4.8

Component Life cycle Hook



Component Life cycle Hook



OnInit: Perform component initialization, retrieve data

OnChanges: Perform action after change to input properties

OnDestroy: Perform cleanup

Using a Lifecycle Hook

```
2 import { Component, OnInit } from '@angular/core';
1 export class ProductListComponent
    implements OnInit {
    pageTitle: string = 'Product List';
    showImage: boolean = false;
    listFilter: string = 'cart';
    products: IProduct[] = [...];
3
3     ngOnInit(): void {
        console.log('In OnInit');
    }
}
```

TS product-list.component.ts ✘

```
1 import { Component } from '@angular/core';
2 import { IProduct } from './product';
3
4 @Component({
5   selector: 'pm-products',
6   templateUrl: './product-list.component.html',
7   styleUrls: ['./product-list.component.css']
8 })
9 export class ProductListComponent implements OnInit {
10   title = 'Product List';
11   [redacted]
12   imageMargin: number = 2;
13   showImage: boolean = false;
14   listFilter: string = 'cart';
15   products: IProduct[] = [
16     {
17       "productId": 2,
18       "productName": "Garden Cart",
19       "productCode": "GDN-0023",
20       "releaseDate": "March 18, 2016",
21       "description": "15 gallon capacity rolling garden cart",
22       "price": 32.99,
23       "starRating": 4.2,
24       "imageUrl": "http://openclipart.org/image/300px/svg_to_png/58471/garden_cart.png"
25     },
26   ]
```

ts product-list.component.ts ✘

```
1 import { Component, OnInit } from '@angular/core';
2 import { IProduct } from './product';
3
4 @Component({
5   selector: 'pm-products',
6   templateUrl: './product-list.component.html',
7   styleUrls: ['./product-list.component.css']
8 })
9 export class ProductListComponent implements OnInit {
10   pageTitle: string = 'Product List';
11   imageWidth: number = 50;
12   imageMargin: number = 2;
13   showImage: boolean = false;
14   listFilter: string = 'cart';
15   products: IProduct[] = [
16     {
17       "productId": 2,
18       "productName": "Garden Cart",
19       "productCode": "GDN-0023",
20       "releaseDate": "March 18, 2016",
21       "description": "15 gallon capacity rolling garden cart",
22       "price": 32.99,
23       "starRating": 4.2,
24       "imageUrl": "http://openclipart.org/image/300px/svg_to_png/58471/garden_cart.png"
25     },
26   ]
```

```
import { Component, OnInit } from '@angular/core';
import { IProduct } from './product';
[ts]
@Component({
  selector: 'app-product-list',
  template: `
    <ul>
      <li *ngFor="let product of products" >{{ product.name }} - {{ product.price }}</li>
    </ul>
  `,
  styleUrls: ['./product-list.component.css']
})
class ProductListComponent implements OnInit {
  pageTitle: string = 'Product List';
  imageWidth: number = 50;
  imageMargin: number = 2;
  showImage: boolean = false;
  listFilter: string = 'cart';
}

export class ProductListComponent implements OnInit {
```

```
toggleImage(): void {
    this.showImage = !this.showImage;
}

ngOnInit(): void {
    console.log('In OnInit');
}
}
```

Filter by:

cart

Filtered by: cart

Show Image	Product	Code	Available	Price	5 Star I
	Garden Cart	ndn-0023	March 18, 2016	\$32.99	4.2

Elements Console Sources Network Performance Memory Application Security Audits

Filter Info ▾

In OnInit
Angular is running in the development mode. Call enableProdMode() to enable the production mode.

>



Filtered by: cart

Show Image	Product	Code	Available	Price	5 Star Rating
	Garden Cart	gdn-0023	March 18, 2016	\$32.99	4.2
	Hammer	tbx-0048	May 21, 2016	\$8.90	4.8

Building a Custom Pipe

```
import { Pipe, PipeTransform } from '@angular/core';

@Pipe({
  name: 'convertToSpaces'
})
export class ConvertToSpacesPipe
  implements PipeTransform {

  transform(value: string,
    character: string): string{
  }
}
```

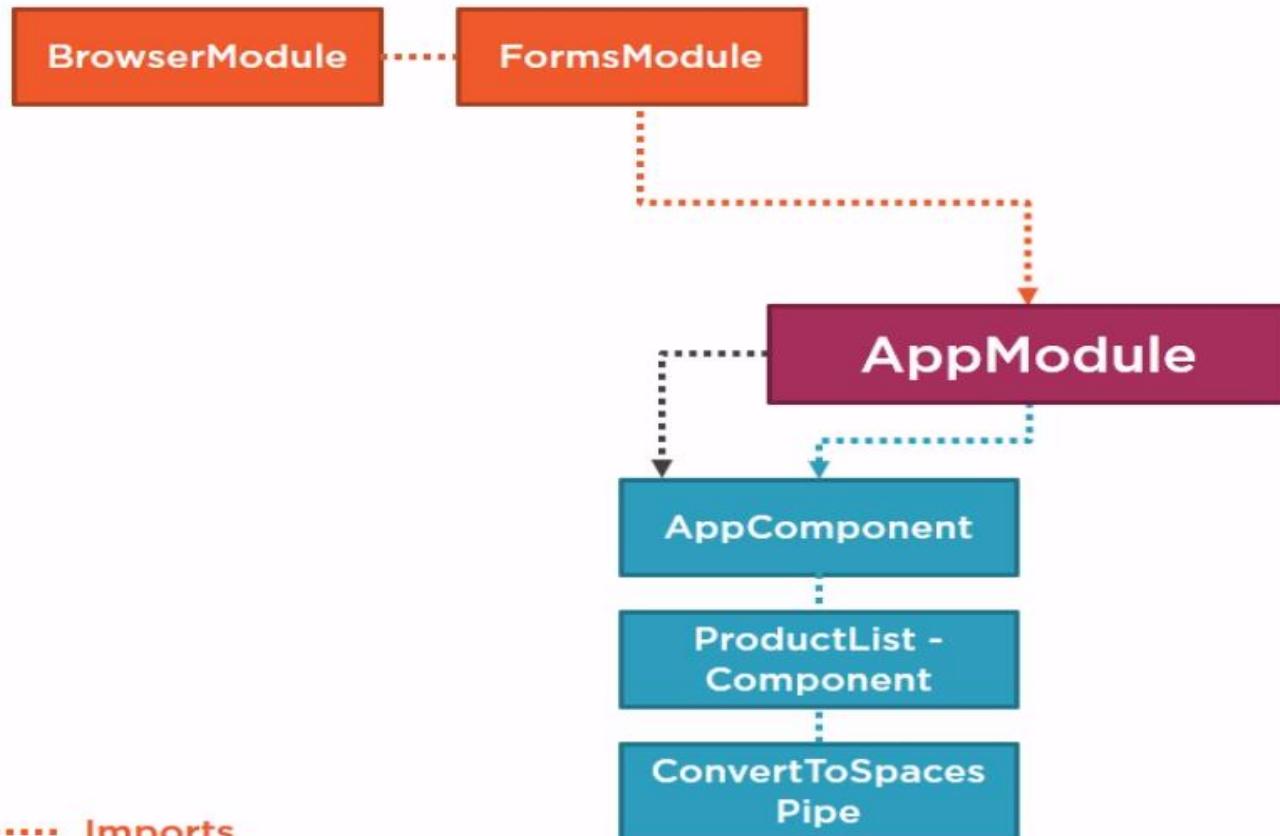
Using a Custom Pipe

Template

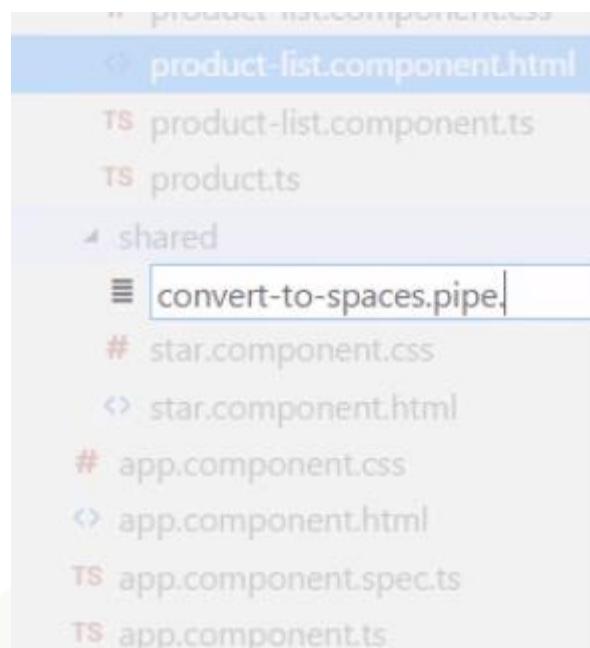
```
<td>{{ product.productCode | convertToSpaces:'-' }}</td>
```

Pipe

```
transform(value: string, character: string): string {  
}  
}
```



- Imports
- Exports
- Declarations
- Providers
- Bootstrap



```
39   [src]
40   [titl
41   [styl
42   [styl
43   </td>
44   <td>{{ produc
45   <td>{{ produc
46   <td>{{ produc
47   <td>{{ produc
48   <td>{{ produc
49   </tr>
50   </tbody>
51   </table>
```

product-list.component.html

TS convert-to-spaces.pipe.ts X

```
1 import { Pipe, PipeTransform } from '@angular/core';
2
3 @Pipe({
4   name: 'convertToSpaces'
5 })
6 export class ConvertToSpacesPipe implements PipeTransform {
7
8 }
```

```
<> product-list.component.html      TS convert-to-spaces.pipe.ts ×
1   import { Pipe, PipeTransform } from '@angular/core';
2
3   @Pipe({
4     name: 'convertToSpaces'
5   })
6   export class ConvertToSpacesPipe implements PipeTransform {
7
8     transform(value: string, character: string): string {
9       return value.replace(character, ' ');
10    }
11  }
12 }
```

```
</td>
<td>{{ product.productName }}</td>
<td>{{ product.productCode | lowercase | convertToSpaces: '-' }}</td>
<td>{{ product.releaseDate }}</td>
<td>{{ product.price | currency:'USD':true:'1.2-2' }}</td>
<td>{{ product.starRating }}</td>
<tr>
```

✖ Uncaught Error: Template parse errors:

```
The pipe 'convertToSpaces' could not be found ("  </td>
  <td>{{ product.productName }}</td>
  <td>{{ [ERROR ->]product.productCode | lowercase | convertToSpaces }}</td>
  <td>{{ product.rel }}: ng:/// AppModule/ProductListComponent.html@44:31
at SyntaxError (http://localhost:4200/vendor.bundle.js:10052:34)
at TemplateParser.webpackJsonp.../.../.../compiler/@angular/compiler.es5.js.TemplateParser.parse (http://localhost:4200/vendor.bundle.js:10052:34)
at JitCompiler.webpackJsonp.../.../.../compiler/@angular/compiler.es5.js.JitCompiler._compileTemplate (http://localhost:4200/vendor.bundle.js:10052:34)
```

product-list.component.html

app.module.ts

convert-to-spaces.pipe.ts

```
1 import { BrowserModule } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3 import { FormsModule } from '@angular/forms';
4
5 import { AppComponent } from './app.component';
6 import { ProductListComponent } from './products/product-list.component';
7 import { ConvertToSpacesPipe } from './shared/convert-to-spaces.pipe';
8
9 @NgModule({
10   declarations: [
11     AppComponent,
12     ProductListComponent,
13     ConvertToSpacesPipe
14   ],
15   imports: [
16     BrowserModule,
17     FormsModule
18   ],
19   providers: [],
20   bootstrap: [AppComponent]
21 })
22 export class AppModule { }
```

Filter by:

Filtered by: cart

Show Image	Product	Code	Available	Price	5 Star Rating
	Garden Cart	gdn 0023	March 18, 2016	\$32.99	4.2
	Hammer	tbx 0048	May 21, 2016	\$8.90	4.8

Acme Product Management

Product List

Filter by:

Filtered by: cart

Show Image	Product	Code	Available	Price	5 Star Rating
	Garden Cart	gdn 0023	March 18, 2016	\$32.99	4.2
	Hammer	tbx 0048	May 21, 2016	\$8.90	4.8

Filtering Data

```
<tr *ngFor='let product of products | productFilter: listFilter'>
```

```
export class ProductListComponent implements OnInit {
  pageTitle: string = 'Product List';
  imageWidth: number = 50;
  imageMargin: number = 2;
  showImage: boolean = false;
  listFilter: string = 'cart';

  filteredProducts: IProduct[];
  products: IProduct[] = [
    {

export class ProductListComponent implements OnInit {
  pageTitle: string = 'Product List';
  imageWidth: number = 50;
  imageMargin: number = 2;
  showImage: boolean = false;

  listFilter: string;
  get listFilter(): string {
    return this._listFilter;
  }
  set listFilter(value:string) {
    this._listFilter = value;
  }

}
```

```
_listFilter: string;
get listFilter(): string {
|   return this._listFilter;
}
set listFilter(value:string) {
|   this._listFilter = value;
}

filteredProducts: IProduct[];
products: IProduct[] = [
|   {

export class ProductListComponent implements OnInit {
pageTitle: string = 'Product List';
imageWidth: number = 50;
imageMargin: number = 2;
showImage: boolean = false;

_listFilter: string;
get listFilter(): string {
|   return this._listFilter;
}
set listFilter(value:string) {
|   this._listFilter = value;
|   this.filteredProducts= this.listFilter ? this.performFilter(this.listFilter) : this.products;
}
```

```
_listFilter: string;
get listFilter(): string {
    return this._listFilter;
}
set listFilter(value:string) {
    this._listFilter = value;
    this.filteredProducts= this.listFilter ? this.performFilter(this.listFilter) : this.products;
}
```

```
performFilter(filterBy: string): IProduct[] {
    filterBy = filterBy.toLocaleLowerCase();
    return this.products.filter((product: IProduct) =>
        product.productName.toLocaleLowerCase().indexOf(filterBy) !== -1);
}
```

```
toggleImage(): void {
    this.showImage = !this.showImage;
}
```

```
ngOnInit(): void {
    console.log('In OnInit');
}
```

```
product-list.component.html      ts product-list.component.ts ×
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
```

```
<tr *ngFor='let product of filteredProducts'>
  <td>
    <img *ngIf='showImage'
        [src]='product.imageUrl'
        [title]='product.productName'
        [style.width.px]='imageWidth'
        [style.margin.px]='imageMargin'>
  </td>
```

Product List

Filter by:

Filtered by: cart

Show Image

Product

Code

Available

Price

5 Star Rating

Garden Cart

gdn 0023

March 18, 2016

\$32.99

4.2



Product List

Filter by:

Filtered by: a

Show Image

Product

Code

Available

Price

5 Star Rating

Garden Cart

gdn 0023

March 18, 2016

\$32.99

4.2

Hammer

tbx 0048

May 21, 2016

\$8.90

4.8



Checklist : Interfaces



Defines custom types

Creating interfaces:

- **interface** keyword
- export it

Implementing interfaces:

- **implements** keyword & interface name
- Write code for each property & method

Using LifeCycle Hooks



styles property

- Specify an array of style strings

styleUrls property

- Specify an array of stylesheet paths

Contd.,



- Import the lifecycle hook interface**
- Implement the lifecycle hook interface**
- Write code for the hook method**

Building a Custom Pipe



Import Pipe and PipeTransform

**Create a class that implements
PipeTransform**

- **export** the class

Write code for the Transform method

Decorate the class with the Pipe decorator

Using a Custom Pipe



Import the custom pipe

Add the pipe to the declarations array of an Angular module

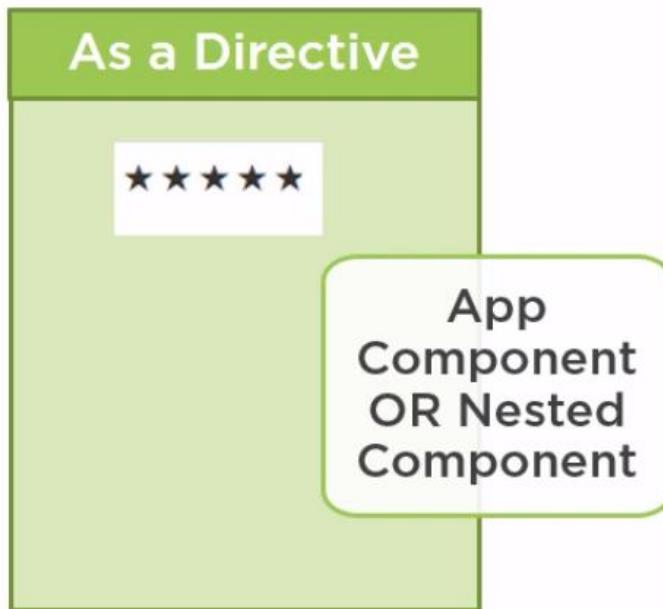
Any template associated with a component that is also declared in that Angular module can use that pipe

Use the Pipe in the template

- Pipe character
- Pipe name
- Pipe arguments (separated with colons)

Nested Components

Using a Component



As a Routing target

Product List

Show Image	Product	Code	Available	Price	5 Star Rating
	Leaf Rake	gdn-0011	March 19, 2016	\$19.95	★★★
	Garden Cart	gdn-0023	March 18, 2016	\$32.99	★★★★
	Hammer	tbx-0048	May 21, 2016	\$8.90	★★★★★
	Saw	tbx-0022	May 15, 2016	\$11.55	★★★★
	Video Game Controller	gmg-0042	October 15, 2015	\$35.95	★★★

Full page style view

```
<body>
  <pm-root></pm-root>
</body>
```

What makes a component nestable ?

Its template only manages a fragment of a larger view

It has a selector

It optionally communicates with its container

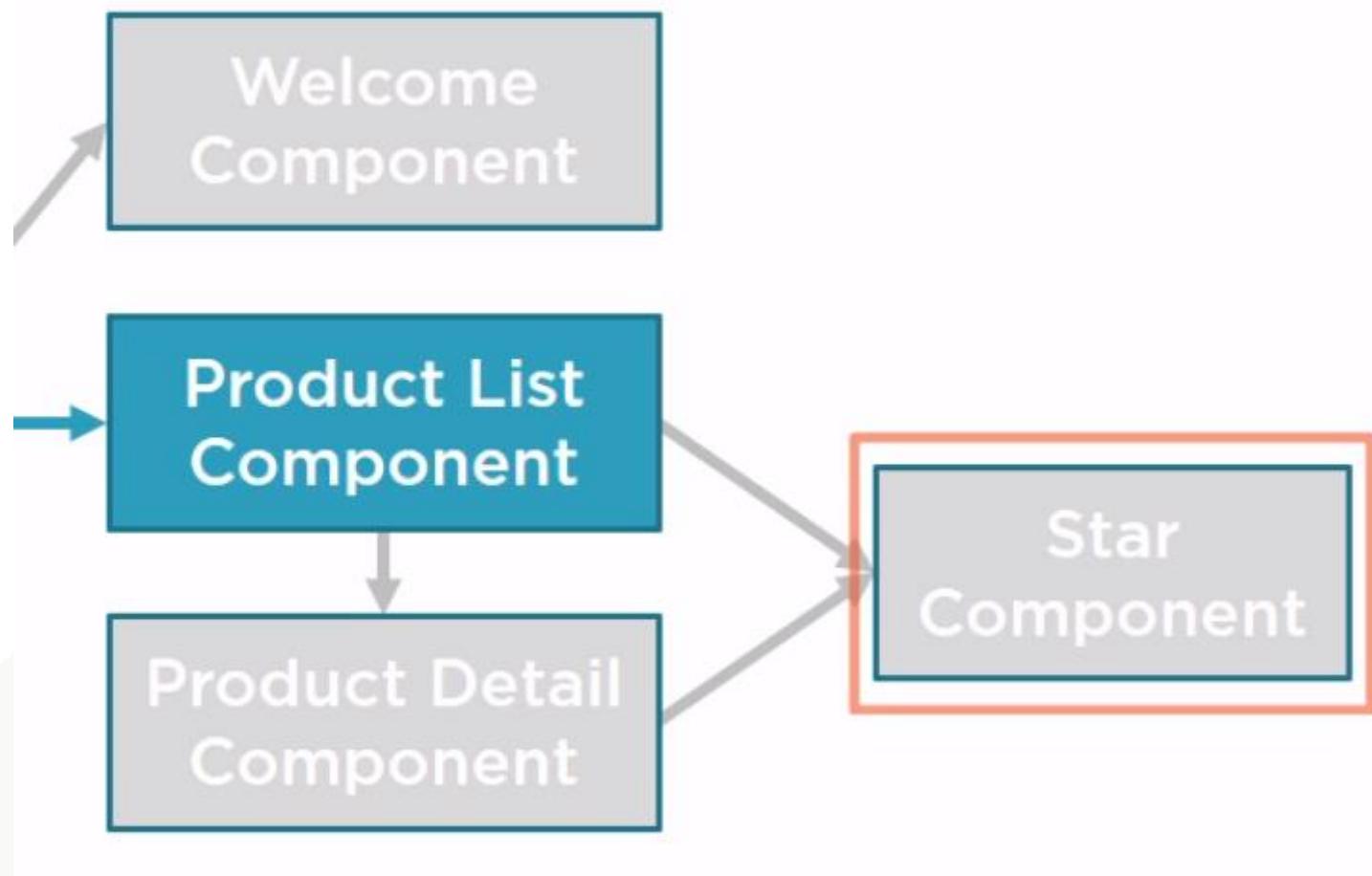
Building a Nested Component

Using a Nested Component

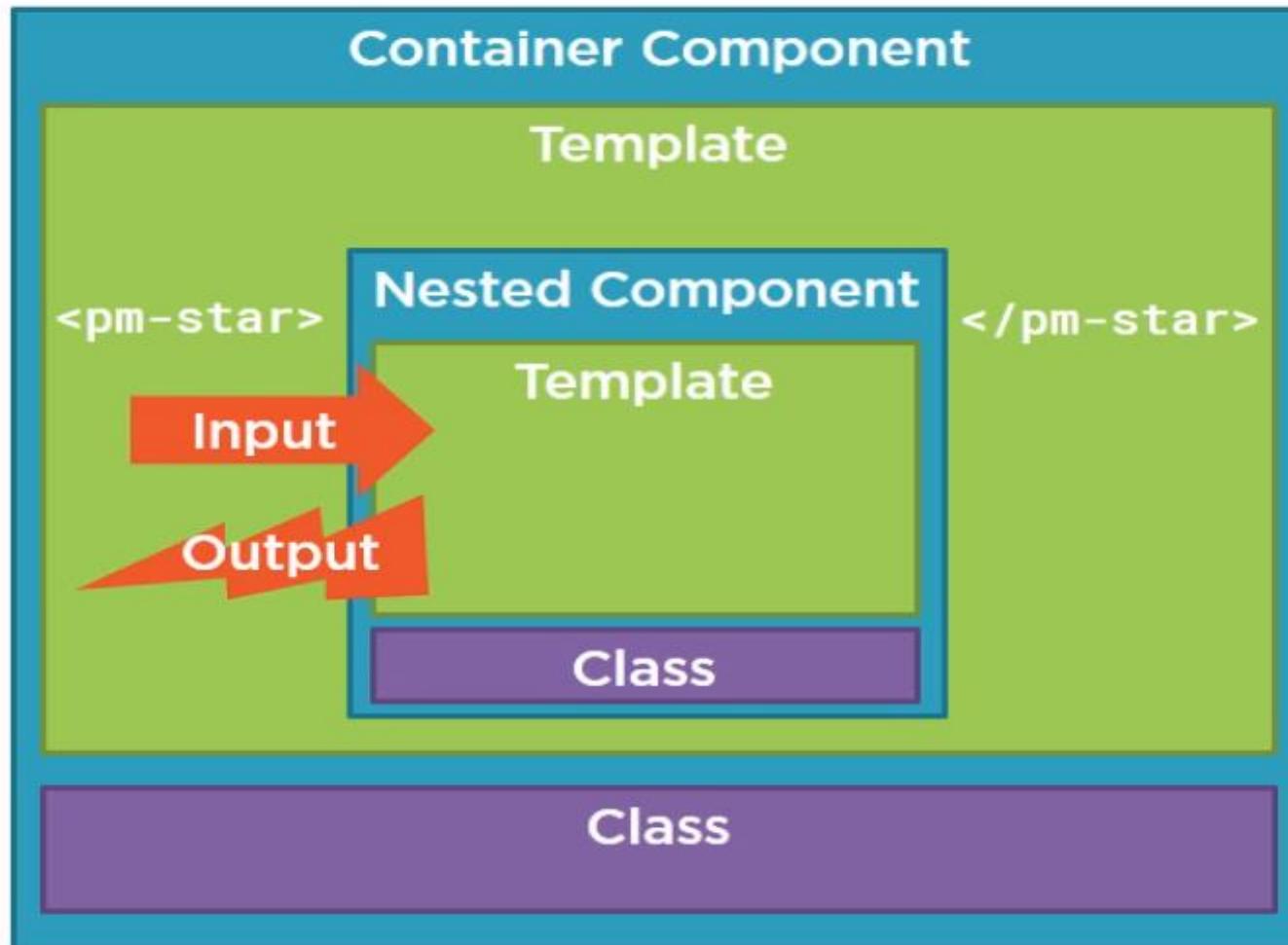
**Passing Data to a Nested Component
Using @Input**

**Raising an Event from a Nested
Component Using @Output**

Application Architecture



Building a Nested Component



Available	Price	5 Star Rating
March 19, 2016	\$19.95	3.2
March 18, 2016	\$32.99	4.2
May 21, 2016	\$8.9	4.8
May 15, 2016	\$11.55	3.7
October 15, 2015	\$35.95	4.6

Code	Available	Price	5 Star Rating
GDN-0011	Mar 19, 2016	\$19.95	★★★
GDN-0023	Mar 18, 2016	\$32.99	★★★★
TBX-0048	May 21, 2016	\$8.99	★★★★★
TBX-0022	May 15, 2016	\$11.55	★★★★
GMG-0042	Oct 15, 2015	\$35.95	★★★★★

```
<td>{{ product.productName }}</td>
<td>{{ product.productCode | lowercase | convertToSpaces: '-' }}</td>
<td>{{ product.releaseDate }}</td>
<td>{{ product.price | currency:'USD':true:'1.2-2' }}</td>
<td>{{ product.starRating }}</td>
</tr>
```

* shared

- TS convert-to-spaces.pipe.ts
- # star.component.css
- star.component.html
- # app.component.css
- app.component.html
- TS app.component.spec.ts

```
46 <td>{{ product.releaseDate }}</td>
47 <td>{{ product.price | currency:'USD':true:'1.2-2' }}</td>
48 <td>{{ product.starRating }}</td>
49 </tr>
50 </tbody>
51 </table>
52 </div>
53 </div>
...  
...
```

```
product.ts  
  TS product.ts  
    ▷ shared  
      TS star.component.ts  
      TS convert-to-spaces.pipe.ts  
      # star.component.css  
41  
42  
43  
44  
45  
46  
47
```

```
product-list.component.html      TS star.component.ts ✘  
1 import { Component } from '@angular/core';  
2  
3 @Component({  
4   selector: 'pm-star',  
5   templateUrl: './star.component.html',  
6   styleUrls: ['./star.component.css']  
7 })  
8 export class StarComponent {  
9  
10 }
```

```
product-list.component.html
```

```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'pm-star',
5   templateUrl: './star.component.html',
6   styleUrls: ['./star.component.css']
7 })
8 export class StarComponent {
9
10 }
```

```
TS star.component.ts ✘
```

```
...
```

```
star.component.html ✘
```

```
1 <div class="crop"
2   [style.width.px]="starWidth"
3   [title]="rating">
4     <div style="width: 86px">
5       <span class="glyphicon glyphicon-star"></span>
6       <span class="glyphicon glyphicon-star"></span>
7       <span class="glyphicon glyphicon-star"></span>
8       <span class="glyphicon glyphicon-star"></span>
9       <span class="glyphicon glyphicon-star"></span>
10      </div>
11    </div>
```

```
product-list.component.html      TS star.component.ts x   ...      star.component.html x   ...
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'pm-star',
5   templateUrl: './star.component.html',
6   styleUrls: ['./star.component.css']
7 })
8 export class StarComponent {
9   rating: number = 4;
10  starWidth: number;
11 }
```

```
1 <div class="crop"
2   [style.width.px]="starWidth"
3   [title]="rating">
4     <div style="width: 86px">
5       <span class="glyphicon glyphicon-star"></span>
6       <span class="glyphicon glyphicon-star"></span>
7       <span class="glyphicon glyphicon-star"></span>
8       <span class="glyphicon glyphicon-star"></span>
9       <span class="glyphicon glyphicon-star"></span>
10    </div>
11  </div>
```

```
product-list.component.html      star.component.ts      star.component.html
1 import { Component, OnChanges } from '@angular/core';
2
3 @Component({
4   selector: 'pm-star',
5   templateUrl: './star.component.html',
6   styleUrls: ['./star.component.css']
7 })
8 export class StarComponent implements OnChanges {
9   rating: number = 4;
10  starWidth: number;
11
12  ngOnChanges(): void {
13    this.starWidth = this.rating * 86/5;
14  }
15}
```

```
1 <div class="crop"
2   [style.width.px]="starWidth"
3   [title]="rating">
4     <div style="width: 86px">
5       <span class="glyphicon glyphicon-star"></span>
6       <span class="glyphicon glyphicon-star"></span>
7       <span class="glyphicon glyphicon-star"></span>
8       <span class="glyphicon glyphicon-star"></span>
9       <span class="glyphicon glyphicon-star"></span>
10    </div>
11  </div>
```

Using Nested Component as a directive

product-list.component.ts

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

product-list.component.html

```
<td>
  {{ product.starRating | number }}
</td>
```

star.component.ts

```
@Component({
  selector: 'pm-star',
  templateUrl: './star.component.html'
})
export class StarComponent {
  rating: number;
  starWidth: number;
}
```

product-list.component.ts

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

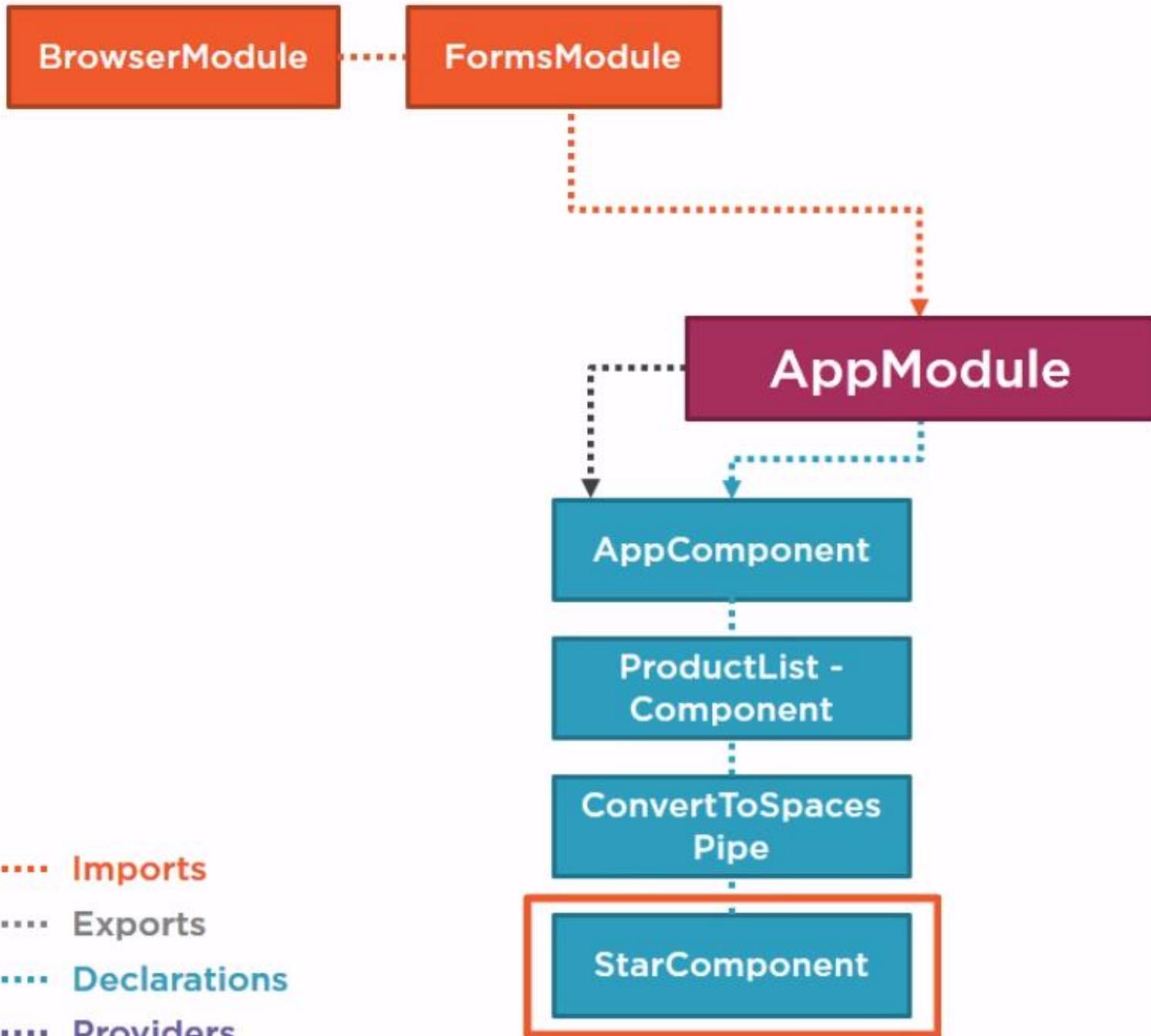
star.component.ts

```
@Component({  
  selector: 'pm-star',  
  templateUrl: './star.component.html'  
})  
export class StarComponent {  
  rating: number;  
  starWidth: number;  
}
```

product-list.component.html

```
<td>  
  <pm-star></pm-star>  
</td>
```





Telling angular about star component

app.module.ts

```
...
import { StarComponent } from './shared/star.component';

@NgModule({
  imports: [
    BrowserModule,
    FormsModule ],
  declarations: [
    AppComponent,
    ProductListComponent,
    ConvertToSpacesPipe,
    StarComponent ],
  bootstrap: [ AppComponent ]
})
export class AppModule { }
```

```
        |           style.marginRight = '10px' >
    </td>
    <td>{{ product.productName }}</td>
    <td>{{ product.productCode | lowercase | conv
    <td>{{ product.releaseDate }}</td>
    <td>{{ product.price | currency:'USD':true:'1
    <td>{{ product.starRating }}</td>
</tr>
</tbody>
</table>
.v>
```

```
    <td>{{ product.productName }}</td>
    <td>{{ product.productCode | lowercase | conv
    <td>{{ product.releaseDate }}</td>
    <td>{{ product.price | currency:'USD':true:'1
        <td><pm-star></pm-star></td>
    </tr>
</tbody>
</table>
.v>
```

```
9  @NgModule({
10  declarations: [
11      AppComponent,
12      ProductListComponent,
13      ConvertToSpacesPipe,
14      StarComponent
15  ],
16  imports: [
17      BrowserModule,
18      FormsModule
19  ],
20  providers: [],
21  bootstrap: [AppComponent]
22 })
23 export class AppModule { }
```

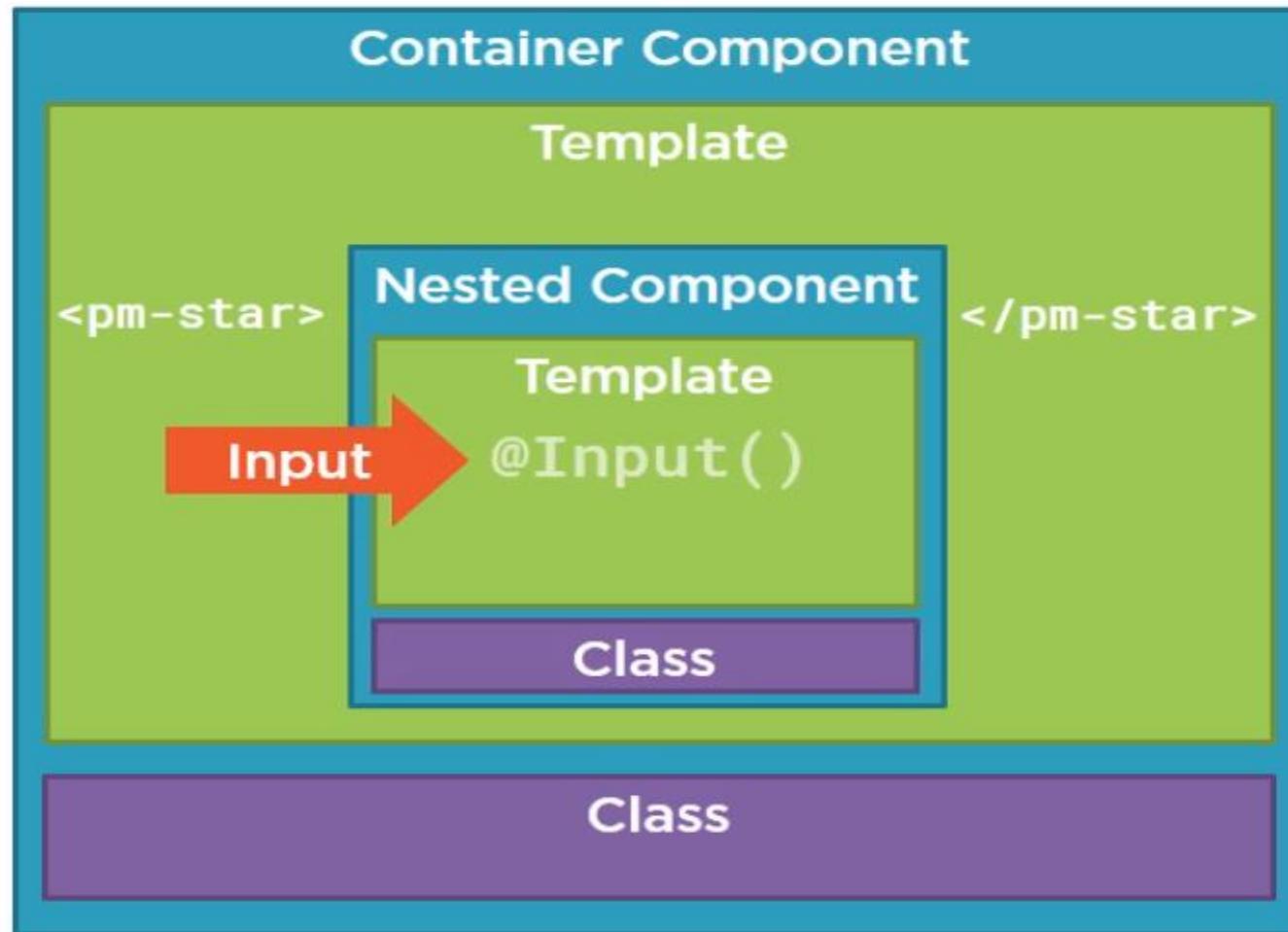
Available	Price	5 Star Rating
March 18, 2016	\$32.99	★★★★★



```
export class StarComponent implements OnChanges {
    rating: number = 4;
    starWidth: number;

    ngOnChanges(): void {
        this.starWidth = this.rating * 86/5;
    }
}
```

Passing data to a nested Component



Passing Data to a nested Component

product-list.component.ts

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

product-list.component.html

```
<td>  
  <pm-star></pm-star>  
</td>
```

star.component.ts

```
@Component({  
  selector: 'pm-star',  
  templateUrl: './star.component.html'  
})  
export class StarComponent {  
  @Input() rating: number;  
  starWidth: number;  
}
```

Passing Data to a Nested Component(@Input)

product-list.component.ts

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

product-list.component.html

```
<td>  
  <pm-star [rating]='product.starRating'>  
  </pm-star>  
</td>
```

star.component.ts

```
@Component({  
  selector: 'pm-star',  
  templateURL: './star.component.html'  
})  
export class StarComponent {  
  @Input() rating: number;  
  starWidth: number;  
}
```

Contd.,

product-list.component.ts

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

product-list.component.html

```
<td>
  <pm-star [rating]='product.starRating'>
  </pm-star>
</td>
```

star.component.ts

```
@Component({
  selector: 'pm-star',
  templateUrl: './star.component.html'
})
export class StarComponent {
  @Input() rating: number;
  starWidth: number;
}
```

```
8  export class StarComponent implements OnChanges {  
9      @Input() rating: number;  
10     starWidth: number;  
11  
12     ngOnChanges(): void {  
13         this.starWidth = this.rating * 86/5;  
14     }  
15 }
```

The screenshot shows an IDE with two files open:

- product-list.component.html**: An Angular template file containing a table with various columns for products. One column contains a component call to `<pm-star [rating]='product.starRating'></pm-star>`.
- star.component.ts**: A TypeScript class definition for the `StarComponent`. It includes an `@Input()` decorator for the `rating` property.

```
product-list.component.html
35  <tbody>
36    <tr *ngFor='let product of filteredProducts'>
37      <td>
38        <img *ngIf='showImage'
39          [src]="product.imageUrl"
40          [title]="product.productName"
41          [style.width.px]="imageWidth"
42          [style.margin.px]="imageMargin">
43      </td>
44      <td>{{ product.productName }}</td>
45      <td>{{ product.productCode | lowercase | slice:0:10 }}</td>
46      <td>{{ product.releaseDate }}</td>
47      <td>{{ product.price | currency:'USD':true:'1.2-2' }}</td>
48      <td>
49        <pm-star [rating]='product.starRating'></pm-star>
50      </td>
51    </tr>
52  </tbody>
53 </table>
54 '>
55
56
```

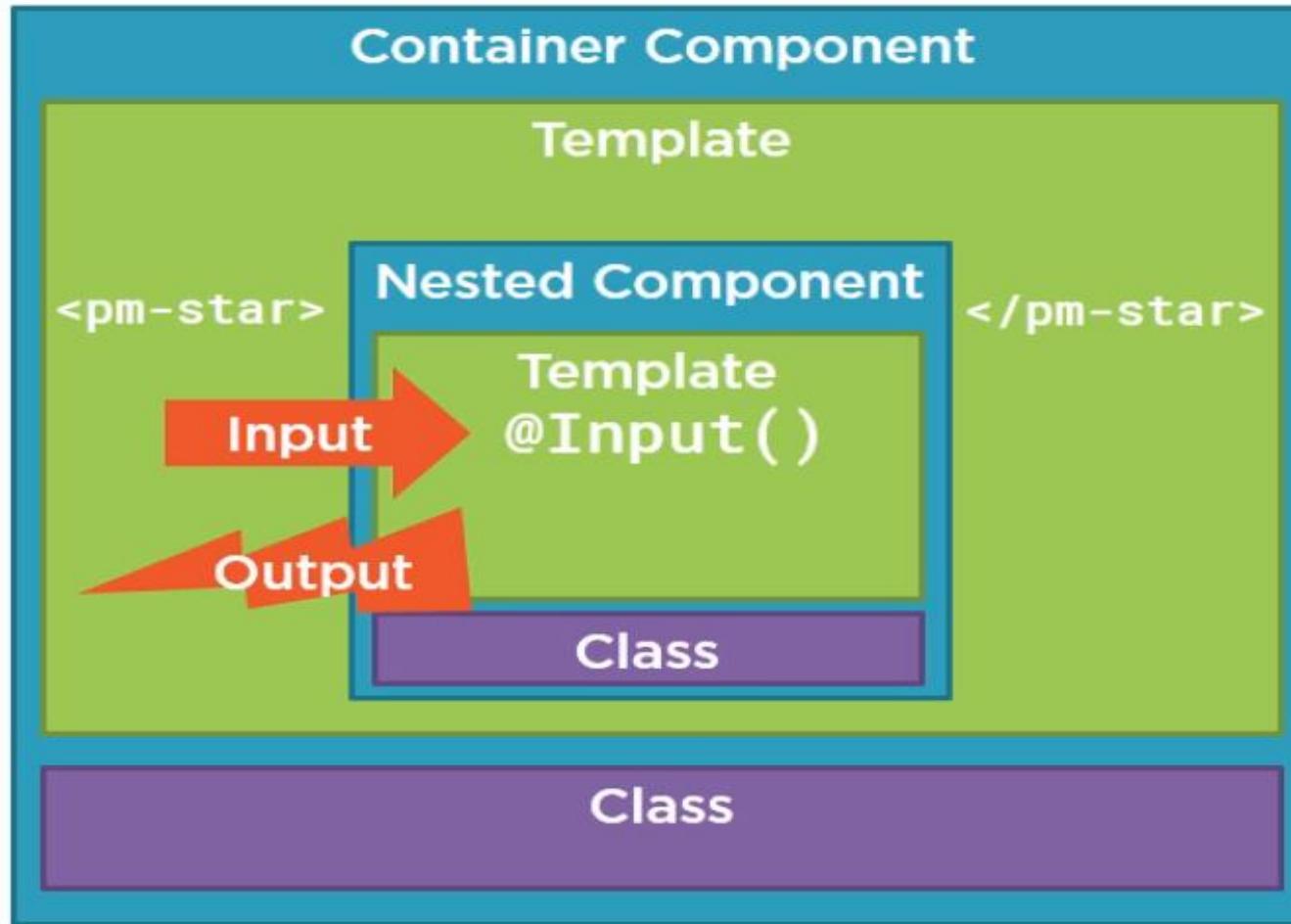
```
star.component.ts
1  import { Component, OnChanges, Input } from '@angular/core';
2
3  @Component({
4    selector: 'pm-star',
5    templateUrl: './star.component.html',
6    styleUrls: ['./star.component.css']
7  })
8  export class StarComponent implements OnChanges {
9    @Input() rating: number;
10   starWidth: number;
11
12   ngOnChanges(): void {
13     this.starWidth = this.rating * 86/5;
14   }
15 }
16
```

```
'  '
8  export class StarComponent implements OnChanges {
9    @Input() rating: number;
10   starWidth: number;
11
12  ngOnChanges(): void {
13    this.starWidth = this.rating * 86/5;
14  }
15}
16
```

Filtered by: cart

Show Image	Product	Code	Available	Price	5 Star Rating
	Garden Cart	gdn-0023	March 18, 2016	\$32.99	★★★★★

Passing the data from Component to the container



Contd.,

product-list.component.ts

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

star.component.ts

```
@Component({  
  selector: 'pm-star',  
  templateUrl: './star.component.html'  
})  
export class StarComponent {  
  @Input() rating: number;  
  starWidth: number;  
  @Output() notify: EventEmitter<string> =  
    new EventEmitter<string>();  
}
```

product-list.component.html

```
<td>  
  <pm-star [rating]='product.starRating'>  
  </pm-star>  
</td>
```

Contd.,

product-list.component.ts

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

product-list.component.html

```
<td>
  <pm-star [rating]='product.starRating'>
  </pm-star>
</td>
```

star.component.ts

```
@Component({
  selector: 'pm-star',
  templateUrl: './star.component.html'
})
export class StarComponent {
  @Input() rating: number;
  starWidth: number;
  @Output() notify: EventEmitter<string> =
    new EventEmitter<string>();
```

```
onClick() {
  this.notify.emit('clicked!');
}
}
```

star.component.html

```
<div (click)='onClick()'>
  ... stars ...
</div>
```



Contd.,

```
@Component({
  selector: 'pm-star',
  templateURL: './star.component.html'
})
export class StarComponent {
  @Input() rating: number;
  starWidth: number;
  @Output() notify: EventEmitter<string> = new EventEmitter<string>

  onClick() {
    this.notify.emit('clicked!');
  }
}

star.component.html
<div (click)='onClick()'>
  ... stars ...
</div>
```

star.component.ts

```
@Component({
  selector: 'pm-star',
  templateUrl: './star.component.html'
})
export class StarComponent {
  @Input() rating: number;
  starWidth: number;
  @Output() notify: EventEmitter<string> =
    new EventEmitter<string>();

  onClick() {
    this.notify.emit('clicked!');
  }
}
```

product-list.component.html

```
<td>
  <pm-star [rating]='product.starRating'>
  </pm-star>
</td>
```

```
  starRating: number,
  @Output() notify: EventEmitter<string>
    new EventEmitter<string>()

  onClick() {
    this.notify.emit('clicked!');
  }
}
```

star.component.html

```
<div (click)='onClick()'>
  ... stars ...
</div>
```

```
export class ProductListComponent {
```

product-list.component.html

```
<td>
  <pm-star [rating]='product.starRating'
            (notify)='onNotify($event)'>
  </pm-star>
</td>
```

```
export class StarComponent {
  @Input() rating: number;
  starWidth: number;
  @Output() notify: EventEmitter<string> = new EventEmitter<string>();
```

```
onClick() {
  this.notify.emit('clicked!');
}
```

star.component.html

```
<div (click)='onClick()'>
  ... stars ...
</div>
```

product-list.component.ts

```
@Component({
  selector: 'pm-products',
  templateURL: './product-list.component.html'
})
export class ProductListComponent {
  onNotify(message: string): void { }
}
```

product-list.component.html

```
<td>
  <pm-star [rating]='product.starRating'
            (notify)='onNotify($event)'>
    </pm-star>
  </td>
```

product-list.component.ts

```
@Component({  
  selector: 'pm-products',  
  templateUrl: './product-list.component.html'  
})  
export class ProductListComponent {  
  onNotify(message: string): void {}  
}
```

product-list.component.html

```
<td>  
  <pm-star [rating]='product.starRating'  
           (notify)='onNotify($event)'>  
  </pm-star>  
</td>
```

star.component.ts

```
@Component({  
  selector: 'pm-star',  
  templateUrl: './star.component.html'  
})  
export class StarComponent {  
  @Input() rating: number;  
  starWidth: number;  
  @Output() notify: EventEmitter<string>  
    = new EventEmitter<string>()  
  onClick() {  
    this.notify.emit('clicked!');  
  }  
}
```

star.component.html

```
<div (click)='onClick()'>  
  ... stars ...  
</div>
```

Product List

Filter by:

Filtered by: cart

Show Image

Product

Code

Available

Price

5 Star Rating

Garden Cart

gdn 0023

March 18, 2016

\$32.99

★★★

```
[style.width.px]="starWidth"
[title]="rating"
(click)="onClick()">
<div style="width: 86px">
  <span class="glyphicon glyphicon-star"></span>
  <span class="glyphicon glyphicon-star"></span>
  <span class="glyphicon glyphicon-star"></span>
  <span class="glyphicon glyphicon-star"></span>
  <span class="glyphicon glyphicon-star"></span>
</div>
```

iv>

```
2
3   @Component({
4     selector: 'pm-star',
5     templateUrl: './star.component.html',
6     styleUrls: ['./star.component.css']
7   })
8   export class StarComponent implements OnChanges {
9     @Input() rating: number;
10    starWidth: number;
11
12    ngOnChanges(): void {
13      this.starWidth = this.rating * 86/5;
14    }
15
16    onClick(): void {
17    }
18  }
19 }
```

The screenshot shows a code editor with two files open:

- product-list.component.html**: An Angular template file containing a star rating component. It includes a div with a crop class, a width of starWidth, and a title of rating. A click event is bound to the onClick() method. Inside, there's a div of width 86px containing five spans, each with a glyphicon glyphicon-star class.
- star.component.ts**: An Angular component file. It imports Component, OnChanges, and Input from '@angular/core'. The component has a selector of pm-star, a template URL of ./star.component.html, and a style URL of ./star.component.css'. It implements OnChanges and has a rating property of type number. The constructor takes a starWidth parameter. The ngOnChanges() method calculates the starWidth based on the rating. The onClick() method logs a message to the console: "The rating \${this.rating} was clicked!"

```
<div class="crop"
  [style.width.px]="starWidth"
  [title]="rating"
  (click)="onClick()">
  <div style="width: 86px">
    <span class="glyphicon glyphicon-star"></span>
    <span class="glyphicon glyphicon-star"></span>
    <span class="glyphicon glyphicon-star"></span>
    <span class="glyphicon glyphicon-star"></span>
    <span class="glyphicon glyphicon-star"></span>
  </div>
</div>
```

```
1   { Component, OnChanges, Input } from '@angular/core';
2
3   ent({
4     ector: 'pm-star',
5     templateUrl: './star.component.html',
6     styleUrls: ['./star.component.css']
7
8   class StarComponent implements OnChanges {
9     put() rating: number;
10    rWidth: number;
11
12   ngOnChanges(): void {
13     this.starWidth = this.rating * 86/5;
14
15   onClick(): void {
16     console.log(`The rating ${this.rating} was clicked!`)
```

Filtered by: cart

Show Image	Product	Code	Available	Price	5 Star Rating
	Garden Cart	gdn 0023	March 18, 2016	\$32.99	

Elements Console Sources Network Performance Memory Application Security Audits

Filter Info ▾

In OnInit

Angular is running in the development mode. Call enableProdMode() to enable the production mode.

The rating 4.2 was clicked!

```
product-list.component.html star.component.html ... star.component.ts product-list.component.ts
```

```
1 <div class="crop"
2   [style.width.px]="starWidth"
3   [title]="rating"
4   (click)="onClick()"
5   <div style="width: 86px">
6     <span class="glyphicon glyphicon-star"></span>
7     <span class="glyphicon glyphicon-star"></span>
8     <span class="glyphicon glyphicon-star"></span>
9     <span class="glyphicon glyphicon-star"></span>
10    <span class="glyphicon glyphicon-star"></span>
11  </div>
12</div>
```

```
1 import { Component, OnChanges, Input } from '@angular/core'
2
3 @Component({
4   selector: 'pm-star',
5   templateUrl: './star.component.html',
6   styleUrls: ['./star.component.css']
7 })
8 export class StarComponent implements OnChanges {
9   @Input() rating: number;
10  starWidth: number;
11
12  ngOnChanges(): void {
13    this.starWidth = this.rating * 86/5;
14  }
15
16  onClick(): void {
17    console.log(`The rating ${this.rating} was clicked!`);
18  }
19}
20
```

```
TS star.component.ts ✘  TS product-list.component.ts
1 import { Component, OnChanges, Input, EventEmitter }
2
3 @Component({
4   selector: 'pm-star',
5   templateUrl: './star.component.html',
6   styleUrls: ['./star.component.css']
7 })
8 export class StarComponent implements OnChanges {
9   @Input() rating: number;
10  starWidth: number;
11  ratingClicked: EventEmitter<number> // Line 11
12
13  ngOnChanges(): void {
14    this.starWidth = this.rating * 86/5;
15  }
16
17  onClick(): void {
18    console.log(`The rating ${this.rating} was clicked!`);
19  }
20}
21
```

```
TS star.component.ts x  TS product-list.component.ts
  1 import { Component, OnChanges, Input, EventEmitter } from
  2
  3 @Component({
  4   selector: 'pm-star',
  5   templateUrl: './star.component.html',
  6   styleUrls: ['./star.component.css']
  7 })
  8 export class StarComponent implements OnChanges {
  9   @Input() rating: number;
 10   starWidth: number;
 11   ratingClicked: EventEmitter<string> =
 12     new EventEmitter<string>();
 13
 14   ngOnChanges(): void {
 15     this.starWidth = this.rating * 86/5;
 16   }
 17
 18   onClick(): void {
 19     console.log(`The rating ${this.rating} was clicked!`);
 20   }
 21 }
 22
```

```
1 import { Component, OnChanges, Input, EventEmitter, Output } from '@angular/core';
2
3 @Component({
4   selector: 'pm-star',
5   templateUrl: './star.component.html',
6   styleUrls: ['./star.component.css']
7 })
8 export class StarComponent implements OnChanges {
9   @Input() rating: number;
10  starWidth: number;
11  @Output() ratingClicked: EventEmitter<string> = new EventEmitter<string>();
12
13
14  ngOnChanges(): void {
15    this.starWidth = this.rating * 86/5;
16  }
17
18  onClick(): void {
19    console.log(`The rating ${this.rating} was clicked!`);
20  }
21}
```



```
TS star.component.ts x TS product-list.component.ts
  1 import { Component, OnChanges, Input, EventEmitter, Out
  2
  3 @Component({
  4   selector: 'pm-star',
  5   templateUrl: './star.component.html',
  6   styleUrls: ['./star.component.css']
  7 })
  8 export class StarComponent implements OnChanges {
  9   @Input() rating: number;
 10   starWidth: number;
 11   @Output() ratingClicked: EventEmitter<string> =
 12     new EventEmitter<string>();
 13
 14   ngOnChanges(): void {
 15     this.starWidth = this.rating * 86/5;
 16   }
 17
 18   onClick(): void {
 19     console.log(`The rating ${this.rating} was clicked!`);
 20   }
 21 }
 22
```

```
TS star.component.ts ✘ TS product-list.component.ts
  1 import { Component, OnChanges, Input, EventEmitter, Output } from '@angular/core';
  2
  3 @Component({
  4   selector: 'pm-star',
  5   templateUrl: './star.component.html',
  6   styleUrls: ['./star.component.css']
  7 })
  8 export class StarComponent implements OnChanges {
  9   @Input() rating: number;
 10   starWidth: number;
 11   @Output() ratingClicked: EventEmitter<string> =
 12     new EventEmitter<string>();
 13
 14   ngOnChanges(): void {
 15     this.starWidth = this.rating * 86/5;
 16   }
 17
 18   onClick(): void {
 19     console.log(`The rating ${this.rating} was clicked!`);
 20   }
 21 }
 22
```

```
<> product-list.component.html <> star.component.html ⌂ ... TS star.component.ts <> TS product-list.component.ts ...
```

```
product-list.component.html (Line numbers 35-55)
```

```
35
36 *ngFor='let product of filteredProducts'
37 <td>
38   <img *ngIf='showImage'
39     [src]='product.imageUrl'
40     [title]='product.productName'
41     [style.width.px]='imageWidth'
42     [style.margin.px]='imageMargin'>
43 </td>
44 <td>{{ product.productName }}</td>
45 <td>{{ product.productCode | lowercase | convertToSpaces:>
46 <td>{{ product.releaseDate }}</td>
47 <td>{{ product.price | currency:'USD':true:'1.2-2'}}</td>
48 <td>
49   <pmm-star [rating]='product.starRating'
50     (ratingClicked)=''/pmm-star>
51 </td>
52 >
53 ...
54
55
```

```
star.component.ts (Line numbers 1-21)
```

```
1 import { Component, OnChanges, Input, EventEmitter, Output
2
3 @Component({
4   selector: 'pmm-star',
5   templateUrl: './star.component.html',
6   styleUrls: ['./star.component.css']
7 })
8 export class StarComponent implements OnChanges {
9   @Input() rating: number;
10  starWidth: number;
11  @Output() ratingClicked: EventEmitter<string> =
12    new EventEmitter<string>();
13
14  ngOnChanges(): void {
15    this.starWidth = this.rating * 86/5;
16  }
17
18  onClick(): void {
19    this.ratingClicked.emit(`The rating ${this.rating}`);
20  }
21 }
```

```
product-list.component.html
35
36 *ngFor='let product of filteredProducts'>
37 <td>
38     <img *ngIf='showImage'
39         [src]= 'product.imageUrl'
40         [title]= 'product.productName'
41         [style.width.px]= 'imageWidth'
42         [style.margin.px]= 'imageMargin'>
43 </td>
44 <td>{{ product.productName }}</td>
45 <td>{{ product.productCode | lowercase | convertToSpaces:>
46 <td>{{ product.releaseDate }}</td>
47 <td>{{ product.price | currency:'USD':true:'1.2-2'}}</td>
48 <td>
49     <pm-star [rating]= 'product.starRating'
50         (ratingClicked)= 'onRatingClicked($event)'></pm-star>
51 </td>
52 >
53 .
54
55
56
```

```
star.component.ts
1 import { Component, OnChanges, Input, EventEmitter, Output } from '@angular/core';
2
3 @Component({
4     selector: 'pm-star',
5     templateUrl: './star.component.html',
6     styleUrls: ['./star.component.css']
7 })
8 export class StarComponent implements OnChanges {
9     @Input() rating: number;
10    starWidth: number;
11    @Output() ratingClicked: EventEmitter<string> = new EventEmitter<string>();
12
13    ngOnChanges(): void {
14        this.starWidth = this.rating * 86/5;
15    }
16
17
18    onClick(): void {
19        this.ratingClicked.emit(`The rating ${this.rating}`);
20    }
21 }
22
```

```
TS star.component.ts      TS product-list.component.ts x
43           "starRating": 4.8,
44           "imageUrl": "http://openclipart.org/image/300px_
45       }
46   ];
47
48   constructor() {
49     this.filteredProducts = this.products;
50     this.listFilter = 'cart';
51   }
52
53   onRatingClicked(message: string): void {
54     this.pageTitle = 'Product List: ' + message;
55   }
56
57   performFilter(filterBy: string): IProduct[] {
58     filterBy = filterBy.toLocaleLowerCase();
59     return this.products.filter((product: IProduct)
60       |   product.productName.toLocaleLowerCase().in_
61     }
```

Acme Product Management

Product List: The rating 4.2 was clicked!

Filter by:

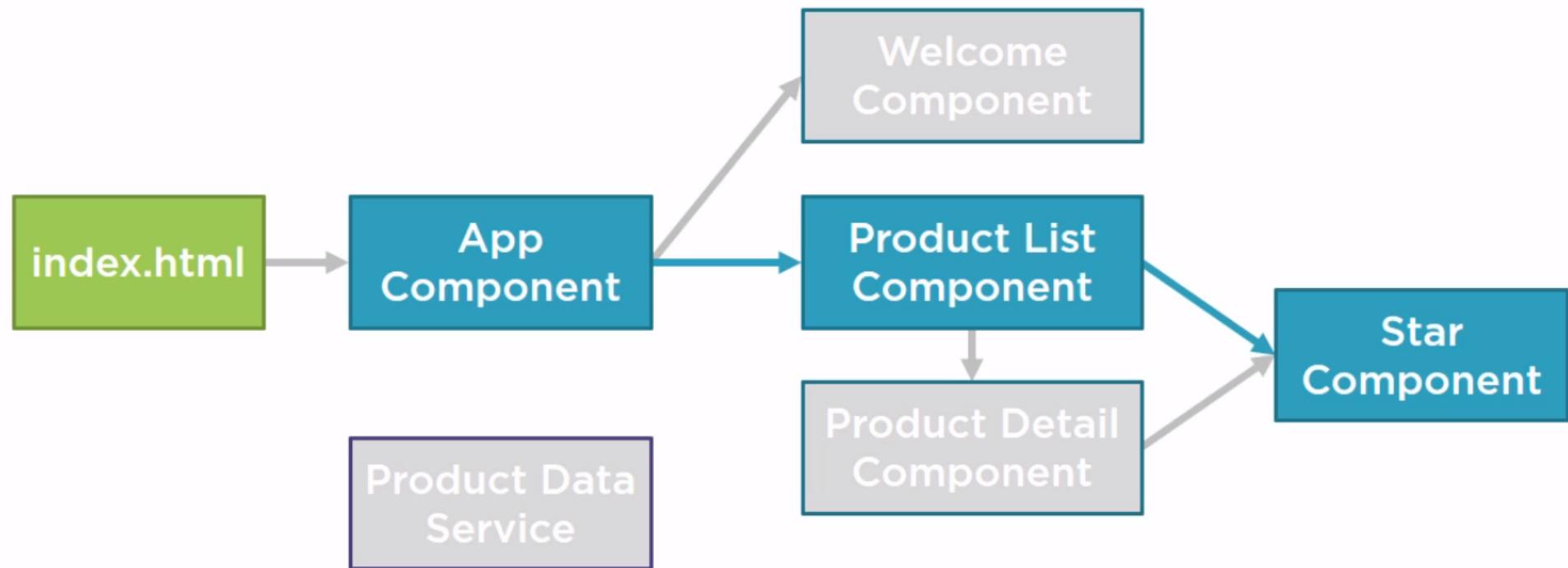
Filtered by: cart

Show Image	Product	Code	Available	Price	5 Star Rating
	Garden Cart	gdn 0023	March 18, 2016	\$32.99	

Services and Dependency Injection

Service

- A Service is a class with a focused purpose .
- Used for features that are :
- Independent from any particular component
- Provide shared data or logic across components
- Encapsulate external interactions



How does it work

Service

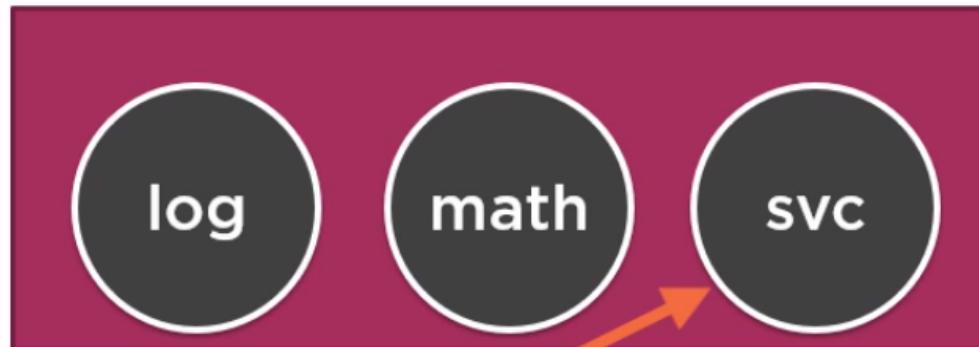
```
export class myService {}
```

Component

```
let svc = new myService();
```



SVC

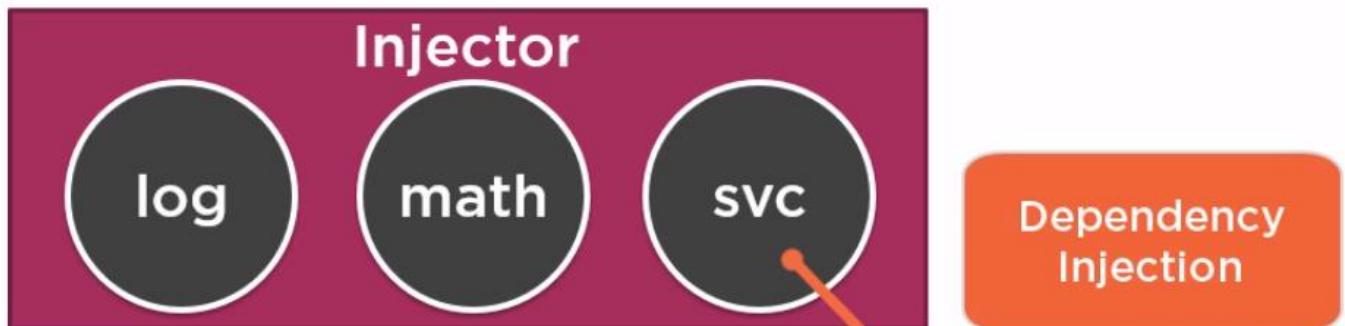


Service

```
export class myService {}
```

Component





Service

```
export class myService {}
```

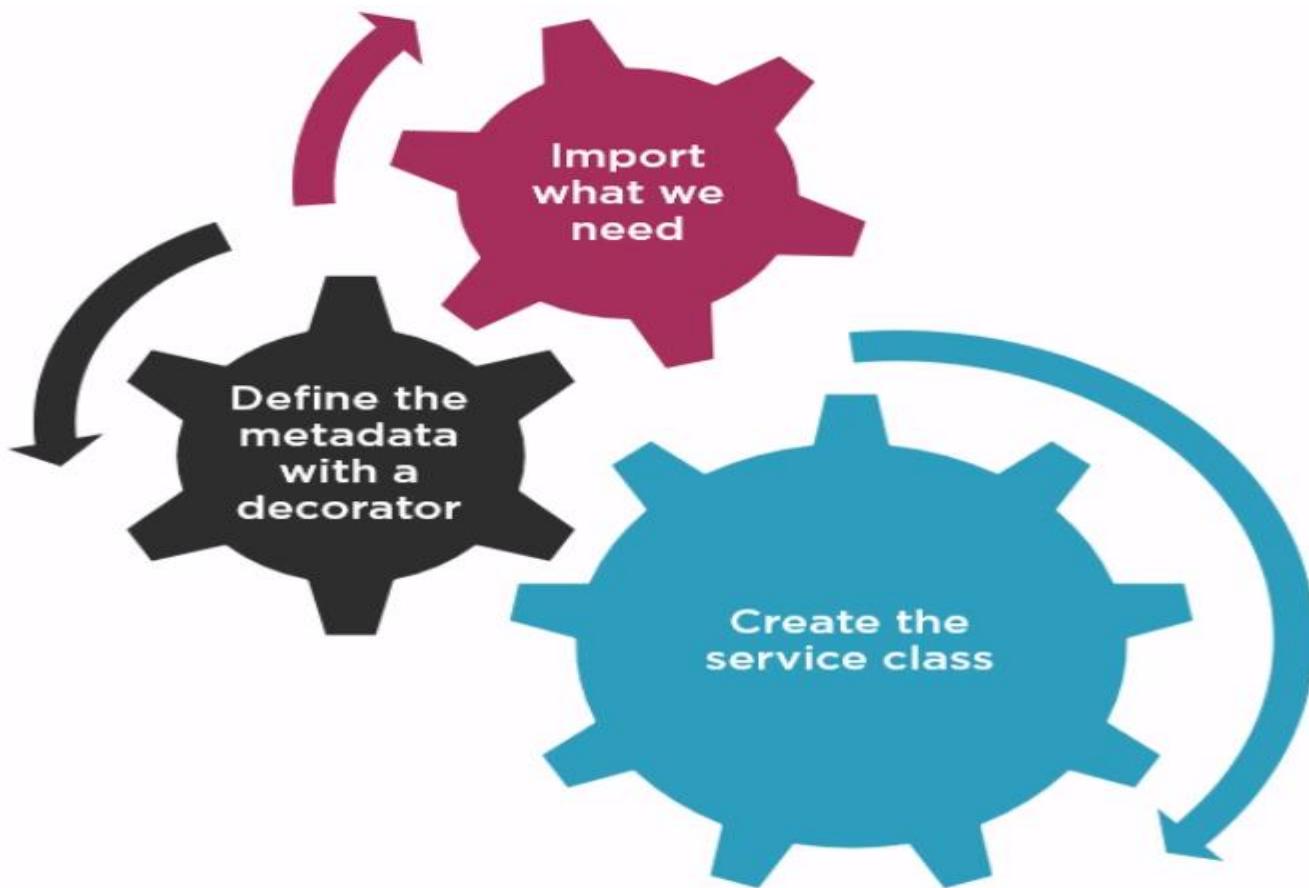
Component

```
constructor(private _myService) {}
```

Dependency Injection

A coding pattern in which a class receives the instances of objects it needs (called **dependencies**) from an external source rather than creating them itself.

Building a Service



Building a Service

product.service.ts

```
import { Injectable } from '@angular/core'

@Injectable()
export class ProductService {

  getProducts(): IProduct[] {
  }

}
```

Contd.,

The screenshot shows a file explorer window and a code editor window.

File Explorer:

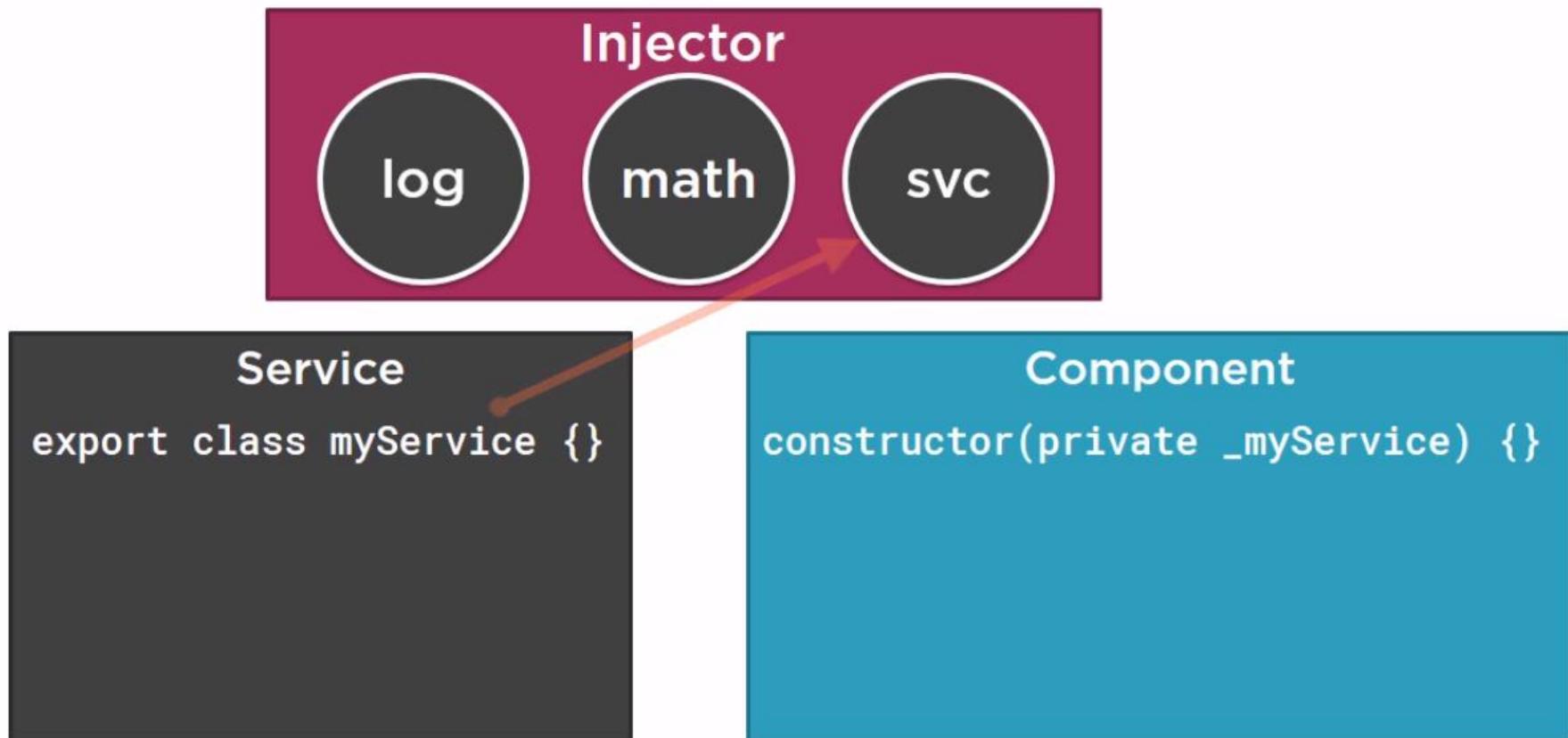
- src
 - api
 - app
 - home
 - products
 - product.service.ts
 - # product-list.component.css
 - <> product-list.component.html
 - product-list.component.ts
 - product.ts
 - shared
 - convert-to-spaces.pipe.ts
 - # star.component.css

```
ts product.service.ts x {} products.json
1 import { Injectable } from '@angular/core';
2
3 import { IProduct } from './product';
4
5 @Injectable()
6 export class ProductService {
7
8     getProducts(): IProduct[] {
9         return [
10            {
11                "productId": 2,
12                "productName": "Garden Cart",
13                "productCode": "GDN-0023",
14                "releaseDate": "March 18, 2016",
15                "description": "15 gallon capacity rolling garden cart",
16                "price": 32.99,
17                "starRating": 4.2,
18                "imageUrl": "http://openclipart.org/image/300px/svg_to_png/58471/garden_cart.png"
19            },
20            {
21                "productId": 5,
22                "productName": "Hammer",
23                "productCode": "TBX-0048",
24                "releaseDate": "May 21, 2016",
25                "description": "Curved claw steel hammer",
26                "price": 8.9,
27                "starRating": 4.8,
```

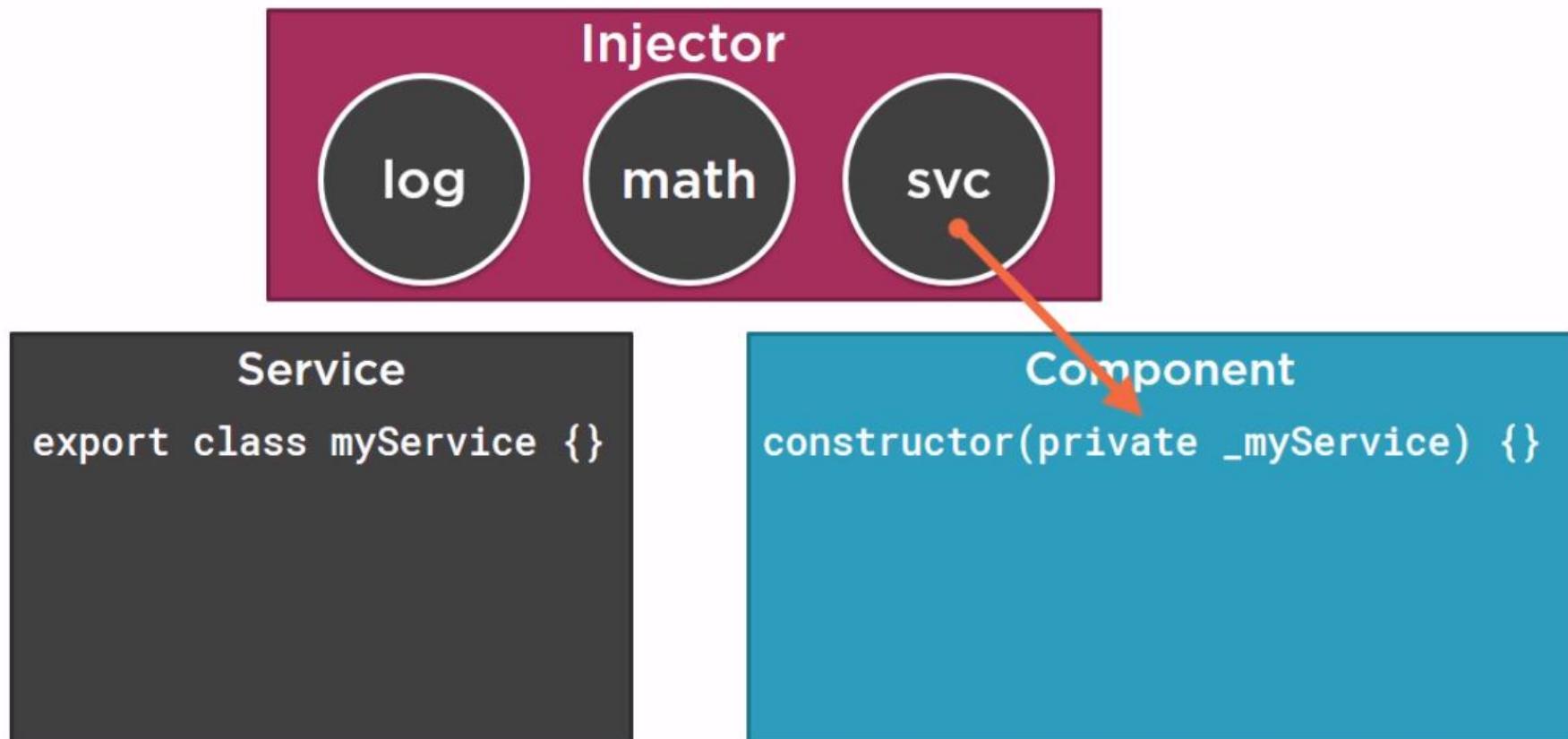
```
src
  api
    products
    products.json
  app
    home
  products
    # product-list.component.css
    <> product-list.component.html
    TS product-list.component.ts
    TS product.service.ts
    TS product.ts
  shared
  # app.component.css
  <> app.component.html
  TS app.component.spec.ts
  TS app.component.ts
  TS app.module.ts
  assets
```

```
  @Injectable()
6  export class ProductService {
7
8    getProducts(): IProduct[] {
9      return [
10        {
11          "productId": 2,
12          "productName": "Garden Cart",
13          "productCode": "GDN-0023",
14          "releaseDate": "March 18, 2016",
15          "description": "15 gallon capacity rolling garden cart",
16          "price": 32.99,
17          "starRating": 4.2,
18          "imageUrl": "http://openclipart.org/image/300px/svg_to_png/",
19        },
20        {
21          "productId": 5,
22          "productName": "Hammer",
23          "productCode": "TBX-0048",
24          "releaseDate": "May 21, 2016",
25          "description": "Curved claw steel hammer",
26          "price": 8.9,
```

Registering a Service



Registering a Service





Register a provider

- Code that can create or return a service
- Typically the service class itself

Define in component OR Angular module metadata

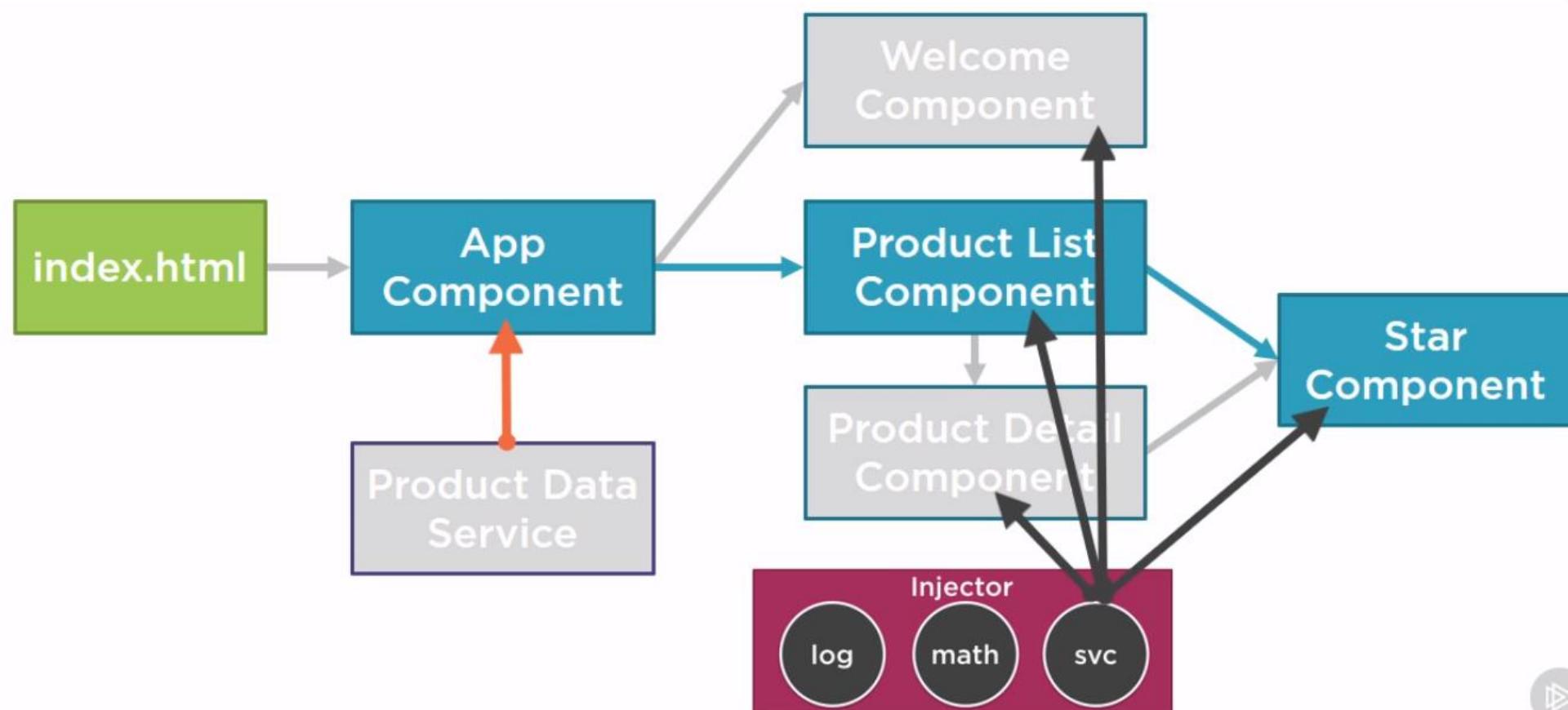
Registered in component:

- Injectable to component AND its children

Registered in Angular module:

- Injectable everywhere in the application

Application Architecture



Registering a Provider

app.component.ts

```
...
import { ProductService } from './products/product.service';

@Component({
  selector: 'pm-root',
  template:
    <div><h1>{{pageTitle}}</h1>
      <pm-products></pm-products>
    </div>
    ,
  providers: [ProductService]
})
export class AppComponent { }
```

TS product.service.ts

TS app.component.ts ✘

```
1 import { Component } from '@angular/core';
2 import { ProductService } from "app/products/product.service";
3
4 @Component({
5   selector: 'pm-root',
6   template: `
7     <div><h1>{{pageTitle}}</h1>
8       <pm-products></pm-products>
9     </div>
10    `,
11   providers:[ ProductService ]
12 })
13 export class AppComponent {
14   pageTitle: string = 'Acme Product Management';
15 }
16
```

Injecting the Service

product-list.component.ts

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

  constructor() {
  }

}
```

Injecting the Service

product-list.component.ts

```
...
import { ProductService } from './product.service';

@Component({
  selector: 'pm-products',
  templateUrl: './product-list.component.html'
})
export class ProductListComponent {
  private _productService;
  constructor(productService: ProductService) {
    _productService = productService;
  }
}
```

product-list.component.ts

```
...
import { ProductService } from './product.service';

@Component({
  selector: 'pm-products',
  templateUrl: './product-list.component.html'
})
export class ProductListComponent {
  private _productService;
  constructor(productService: ProductService) {
    _productService = productService;
  }
}
```

product-list.component.ts

```
...
import { ProductService } from './product.service';

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

  constructor(private _productService: ProductService) {
  }

}
```

```
TS product.service.ts      TS app.component.ts      TS product-list.component.ts ✘
1 import { Component, OnInit } from '@angular/core';
2
3 import { IProduct } from './product';
4
5 @Component({
6   selector: 'pm-products',
7   templateUrl: './product-list.component.html',
8   styleUrls: ['./product-list.component.css']
9 })
10 export class ProductListComponent implements OnInit {
11   pageTitle: string = 'Product List';
12   imageWidth: number = 50;
13   imageMargin: number = 2;
14   showImage: boolean = false;
15
16   _listFilter: string;
17   get listFilter(): string {
18     return this._listFilter;
19   }
20   set listFilter(value: string) {
21     this._listFilter = value;
22     this.filteredProducts = this.listFilter ? this.performFilter(this.listFilter) : this.products;
23   }
24
25   filteredProducts: IProduct[];
26   products: IProduct[] = [
27     {
```

```
constructor(private _productService: ProductService) {
    this.filteredProducts = this.products;
    this.listFilter = 'cart';
}

onRatingClicked(message: string): void {
    this.pageTitle = 'Product List: ' + message;
}

performFilter(filterBy: string): IProduct[] {
    filterBy = filterBy.toLocaleLowerCase();
    return this.products.filter((product: IProduct) =>
        product.productName.toLocaleLowerCase().indexOf(filterBy) !== -1)
}

toggleImage(): void {
    this.showImage = !this.showImage;
}

ngOnInit(): void {
    console.log('In OnInit');
}
```

```
];

constructor(private _productService: ProductService) {
    this.filteredProducts = this.products;
    this.listFilter = 'cart';
}

onRatingClicked(message: string): void {
    this.pageTitle = 'Product List: ' + message;
}

performFilter(filterBy: string): IProduct[] {
    filterBy = filterBy.toLocaleLowerCase();
    return this.products.filter((product: IProduct) =>
        product.productName.toLocaleLowerCase().indexOf(filterBy) !== -1);
}

toggleImage(): void {
    this.showImage = !this.showImage;
}

ngOnInit(): void {
    console.log('In OnInit');
}
}
```

```
TS product.service.ts      TS app.component.ts      TS product-list.component.ts x
29     constructor(private _productService: ProductService) {
30         this.filteredProducts = this.products;
31         this.listFilter = 'car';
32     }
33
34     onRatingClicked(message: string): void {
35         this.pageTitle = 'Product List: ' + message;
36     }
37
38     performFilter(filterBy: string): IProduct[] {
39         filterBy = filterBy.toLocaleLowerCase();
40         return this.products.filter((product: IProduct) =>
41             product.productName.toLocaleLowerCase().indexOf(filterBy) !== -1);
42     }
43
44     toggleImage(): void {
45         this.showImage = !this.showImage;
46     }
47
48     ngOnInit(): void {
49         this.products = this._productService|
50     }
51 }
52 }
```

TS product.service.ts TS app.component.ts TS product-list.component.ts X

```
29     constructor(private _productService: ProductService) {
30         this.filteredProducts = this.products;
31         this.listFilter = 'cart';
32     }
33
34     onRatingClicked(message: string): void {
35         this.pageTitle = 'Product List: ' + message;
36     }
37
38     performFilter(filterBy: string): IProduct[] {
39         filterBy = filterBy.toLocaleLowerCase();
40         return this.products.filter((product: IProduct) =>
41             product.productName.toLocaleLowerCase().indexOf(filterBy) !== -1);
42     }
43
44     toggleImage(): void {
45         this.showImage = !this.showImage;
46     }
47
48     ngOnInit(): void {
49         this.products = this._productService.|
```

getProducts (method) ProductService.getProducts(..)

50 }

51 }

52 }

```
TS product.service.ts      TS app.component.ts      TS product-list.component.ts ×
29   constructor(private _productService: ProductService) {
30     this.filteredProducts = this.products;
31     this.listFilter = 'cart';
32   }
33
34   onRatingClicked(message: string): void {
35     this.pageTitle = 'Product List: ' + message;
36   }
37
38   performFilter(filterBy: string): IProduct[] {
39     filterBy = filterBy.toLocaleLowerCase();
40     return this.products.filter((product: IProduct) =>
41       product.productName.toLocaleLowerCase().indexOf(filterBy) !== -1);
42   }
43
44   toggleImage(): void {
45     this.showImage = !this.showImage;
46   }
47
48   ngOnInit(): void {
49     this.products = this._productService.getProducts();
50   }
51 }
52 }
```

```
TS product.service.ts      TS app.component.ts      TS product-list.component.ts x
29     constructor(private _productService: ProductService) {
30         this.listFilter = 'cart';
31     }
32
33     onRatingClicked(message: string): void {
34         this.pageTitle = 'Product List: ' + message;
35     }
36
37     performFilter(filterBy: string): IProduct[] {
38         filterBy = filterBy.toLocaleLowerCase();
39         return this.products.filter((product: IProduct) =>
40             product.productName.toLocaleLowerCase().indexOf(filterBy) !== -1);
41     }
42
43     toggleImage(): void {
44         this.showImage = !this.showImage;
45     }
46
47     ngOnInit(): void {
48         this.products = this._productService.getProducts();
49         this.filteredProducts = this.products;
50     }
51 }
```

Observables and Reactive Extensions

Help manage asynchronous data

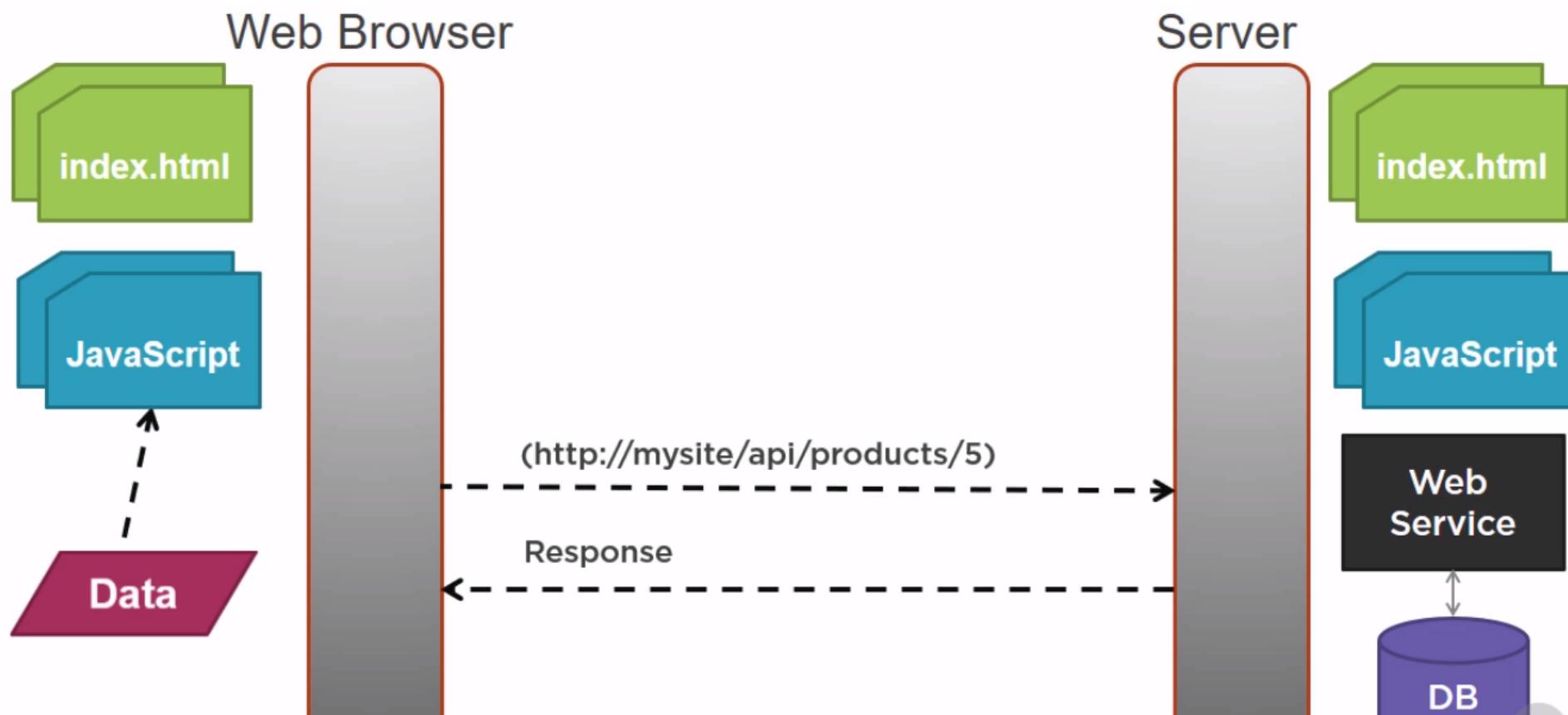
Treat events as a collection

- An array whose items arrive asynchronously over time

Are a proposed feature for ES 2016

Use Reactive Extensions (RxJS)

Retrieving Data using HTTP



Observable Operators

Methods on observables that compose new observables

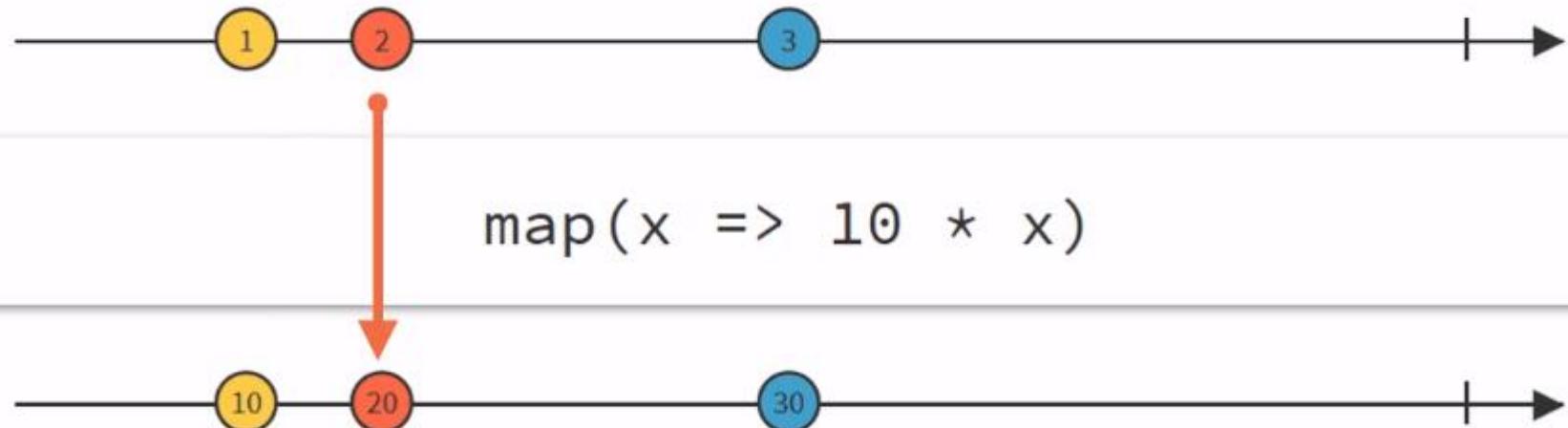
Transform the source observable in some way

Process each value as it is emitted

Examples: map, filter, take, merge, ...

Contd.,

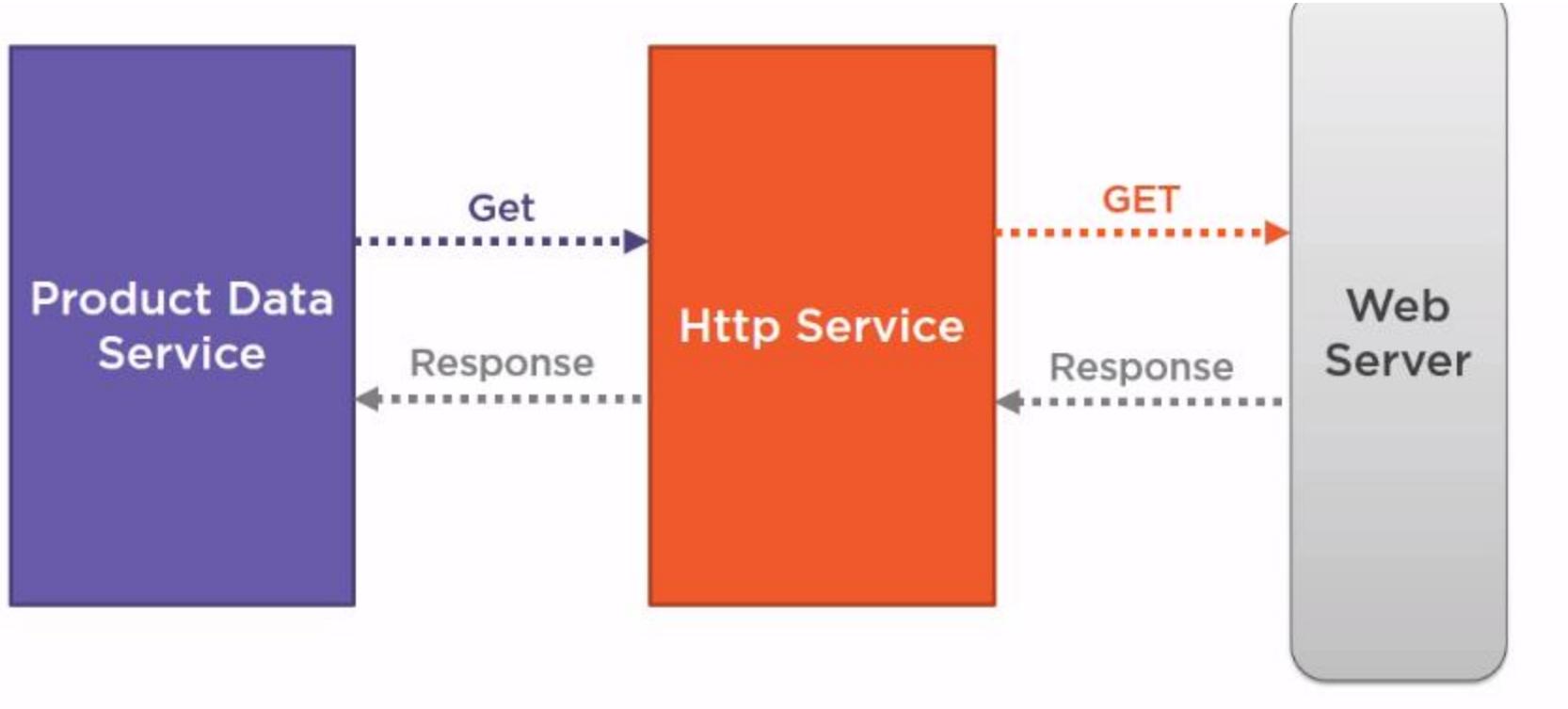
Interactive diagrams of Rx Observables



Promise Vs Observable

Promise	Observable
Provides a single future value	Emits multiple values over time
Not lazy	Lazy
Not cancellable	Cancellable
	Supports map, filter, reduce and similar operators

Sending an Http request



Registering the HTTP service provider

product.service.ts

```
...
import { HttpClient } from '@angular/common/http';

@Injectable()
export class ProductService {
  private _productUrl = 'www.myWebService.com/api/products';

  constructor(private _http: HttpClient) { }

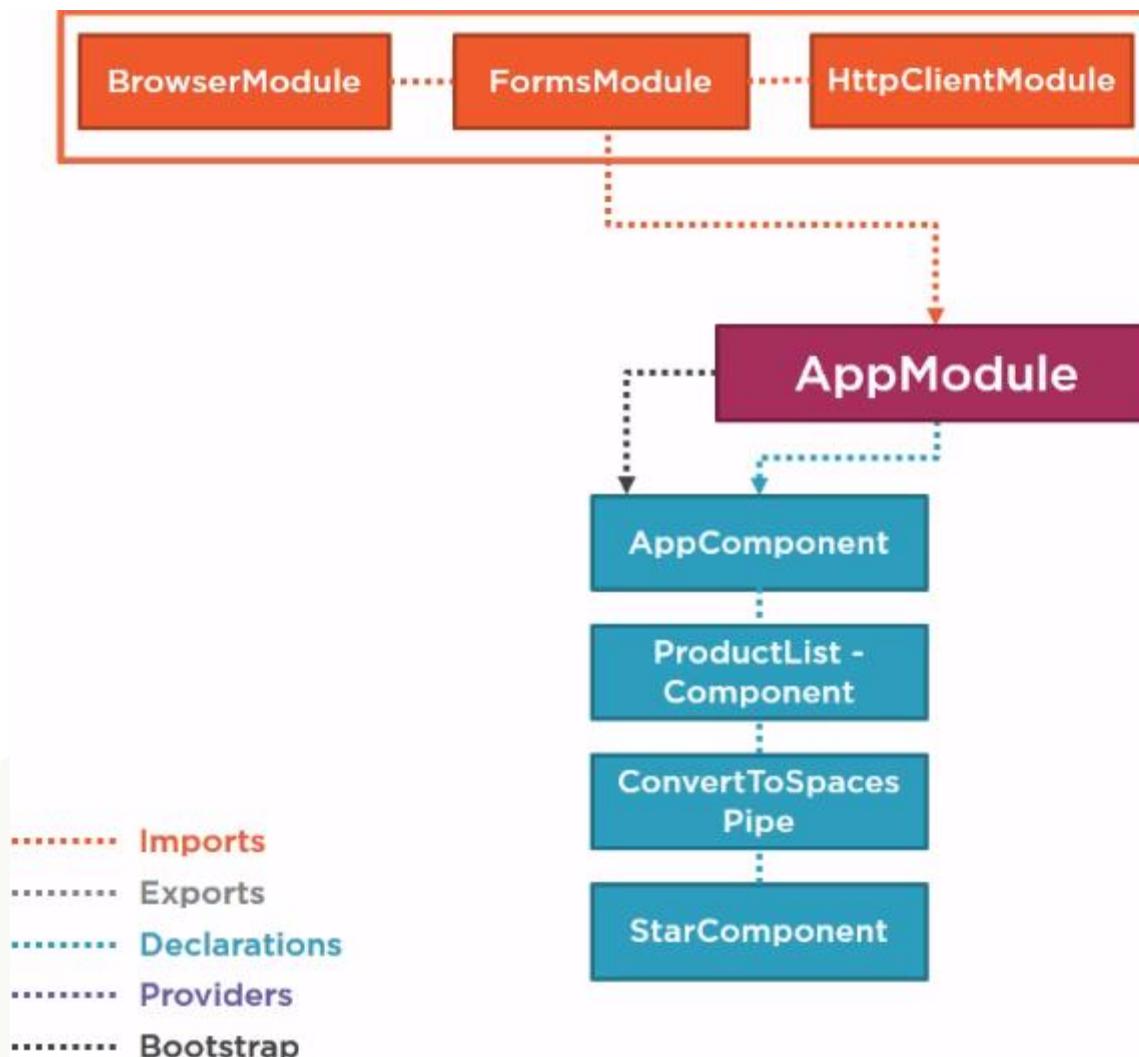
  getProducts() {
    return this._http.get(this._productUrl);
  }
}
```

Contd.,

app.module.ts

```
...
import { HttpClientModule } from '@angular/common/http';

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



Contd.,

product.service.ts

```
...
import { HttpClient } from '@angular/common/http';

@Injectable()
export class ProductService {
  private _productUrl = 'www.myWebService.com/api/products';

  constructor(private _http: HttpClient) { }

  getProducts() {
    return this._http.get(this._productUrl);
  }
}
```

Contd.,

product.service.ts

```
...
import { HttpClient } from '@angular/common/http';

@Injectable()
export class ProductService {
  private _productUrl = 'www.myWebService.com/api/products';

  constructor(private _http: HttpClient) { }

  getProducts() {
    return this._http.get<IProduct[]>(this._productUrl);
  }
}
```

product.service.ts

```
...
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs/Observable';

@Injectable()
export class ProductService {
  private _productUrl = 'www.myWebService.com/api/products';

  constructor(private _http: HttpClient) { }

  getProducts(): Observable<IProduct[]> {
    return this._http.get<IProduct[]>(this._productUrl);
  }
}
```

```
TS app.module.ts ✘  TS product.service.ts

 1 import { BrowserModule } from '@angular/platform-browser';
 2 import { NgModule } from '@angular/core';
 3 import { FormsModule } from '@angular/forms';
 4 import { HttpClientModule } from '@angular/common/http';
 5
 6 import { AppComponent } from './app.component';
 7 import { ProductListComponent } from './products/product-list.component';
 8 import { ConvertToSpacesPipe } from './shared/convert-to-spaces.pipe';
 9 import { StarComponent } from './shared/star.component';
10
11 @NgModule({
12   declarations: [
13     AppComponent,
14     ProductListComponent,
15     ConvertToSpacesPipe,
16     StarComponent
17   ],
18   imports: [
19     BrowserModule,
20     FormsModule,
21     HttpClientModule
22   ],
23   providers: [],
24   bootstrap: [AppComponent]
25 })
26 export class AppModule { }
```

```
TS app.module.ts      TS product.service.ts ✘

1 import { Injectable } from '@angular/core';
2 import { HttpClient } from '@angular/common/http';
3 import { Observable } from 'rxjs/Observable';
4
5 import { IProduct } from './product';
6
7 @Injectable()
8 export class ProductService {
9
10    getProducts(): IProduct[] {
11        return [
12            {
13                "productId": 2,
14                "productName": "Garden Cart",
15                "productCode": "GDN-0023",
16                "releaseDate": "March 18, 2016",
17                "description": "15 gallon capacity rolling garden cart",
18                "price": 32.99,
19                "starRating": 4.2,
20                "imageUrl": "http://openclipart.org/image/300px/svg_to_png/58471/garden_cart.png"
21            },
22            {
23                "productId": 5,
24                "productName": "Hammer",
25                "productCode": "TBX-0048",
26                "releaseDate": "May 21, 2016",
```

```
TS app.module.ts      TS product.service.ts ✘
1 import { Injectable } from '@angular/core';
2 import { HttpClient } from '@angular/common/http';
3 import { Observable } from 'rxjs/Observable';
4
5 import { IProduct } from './product';
6
7 @Injectable()
8 export class ProductService {
9   private _productUrl = './api/products/products.json';
10
11   constructor(private _http: HttpClient) {}
12
13   getProducts(): IProduct[] {
```

```
TS app.module.ts    TS product.service.ts ✘

1  import { Injectable } from '@angular/core';
2  import { HttpClient } from '@angular/common/http';
3  import { Observable } from 'rxjs/Observable';
4
5  import { IProduct } from './product';
6
7  @Injectable()
8  export class ProductService {
9      private _productUrl = './api/products/products.json';
10
11     constructor(private _http: HttpClient) {}
12
13     getProducts(): Observable<IProduct[]> {
14         return this._http.get<IProduct[]>(this._productUrl);
15     }
16 }
17
```

Exception Handling

product.service.ts

```
...
import { HttpClient, HttpErrorResponse } from '@angular/common/http';
import { Observable } from 'rxjs/Observable';
import 'rxjs/add/operator/catch';
import 'rxjs/add/operator/do';
...

getProducts(): Observable<IProduct[]> {
  return this._http.get<IProduct[]>(this._productUrl)
    .do(data => console.log('All: ' + JSON.stringify(data)))
    .catch(this.handleError);
}

private handleError(err: HttpErrorResponse) {
}
```

```
TS app.module.ts    TS product.service.ts ✘

1  import { Injectable } from '@angular/core';
2  import { HttpClient, HttpErrorResponse } from '@angular/common/http';
3  import { Observable } from 'rxjs/Observable';
4  import 'rxjs/add/observable/throw';
5  import 'rxjs/add/operator/catch';
6  import 'rxjs/add/operator/do';
7
8  import { IProduct } from './product';
9
10 @Injectable()
11 export class ProductService {
12     private _productUrl = './api/products/products.json';
13
14     constructor(private _http: HttpClient) { }
15
16     getProducts(): Observable<IProduct[]> {
17         return this._http.get<IProduct[]>(this._productUrl)
18             .catch(this.handleError);
19     }
20
21     private handleError(err: HttpErrorResponse) {
22         console.log(err.message);
23         return Observable.throw(err.message);
24     }
25 }
26
```

Subscribing to an observable

```
x.then(valueFn, errorFn)          //Promise  
x.subscribe(valueFn, errorFn)      //Observable  
x.subscribe(valueFn, errorFn, completeFn) //Observable  
let sub = x.subscribe(valueFn, errorFn, completeFn)
```

product-list.component.ts

```
ngOnInit(): void {  
    this._productService.getProducts()  
        .subscribe(  
            products => this.products = products,  
            error => this.errorMessage = <any>error);  
}
```

```
26     filteredProducts: IProduct[];
27     products: IProduct[] = [];
28
29     constructor(private _productService: ProductService) {
30
31     }
32
33     onRatingClicked(message: string): void {
34         this.pageTitle = 'Product List: ' + message;
35     }
36
37     performFilter(filterBy: string): IProduct[] {
38         filterBy = filterBy.toLocaleLowerCase();
39         return this.products.filter((product: IProduct) =>
40             product.productName.toLocaleLowerCase().indexOf(filterBy) !== -1);
41     }
42     [ts]
43     toggle() {
44         Type 'Observable<IProduct[]>' is not assignable to type 'IProduct[]'.
45         Property 'includes' is missing in type 'Observable<IProduct[]
46         >'.
47     }
48     ngOnChanges(): void {
49         (property) ProductListComponent.products: IProduct[]
50         this.products = this._productService.getProducts();
51         this.filteredProducts = this.products;
52     }
53 }
```

```
26     filteredProducts: IProduct[];  
27     products: IProduct[] = [];  
28  
29     constructor(private _productService: ProductService) {  
30     }  
31  
32  
33     onRatingClicked(message: string): void {  
34         this.pageTitle = 'Product List: ' + message;  
35     }  
36  
37     performFilter(filterBy: string): IProduct[] {  
38         filterBy = filterBy.toLocaleLowerCase();  
39         return this.products.filter((product: IProduct) =>  
40             product.productName.toLocaleLowerCase().indexOf(filterBy) !== -1);  
41     }  
42  
43     toggleImage(): void {  
44         this.showImage = !this.showImage;  
45     }  
46  
47     ngOnInit(): void {  
48         this._productService.getProducts()  
49             .subscribe(products => this.products = products);  
50         this.filteredProducts = this.products;
```

```
errorMessage: string;  
  
_listFilter: string;  
get listFilter(): string {  
    return this._listFilter;  
}  
set listFilter(value: string) {
```

```
toggleImage(): void {
    this.showImage = !this.showImage;
}

ngOnInit(): void {
    this._productService.getProducts()
        .subscribe(products => this.products = products,
                  error => this.errorMessage = <any>error);
    this.filteredProducts = this.products;
}
```

Filtered by:

Show Image

Product

Code

Available

Price

5 Star Rating

```
ts product-list.component.ts x
42 }
43
44 toggleImage(): void {
45     this.showImage = !this.showImage;
46 }
47
48 ngOnInit(): void {
49     1 this._productService.getProducts()
50     2 .subscribe(products => this.products = products,
51                 error => this.errorMessage = <any>err
52     3
53     4
54     5 this.filteredProducts = this.products;
55 }
```

```
ts product.service.ts x
10  jectable()
11  100rt class ProductService {
12  private _productUrl = './api/products/products.json';
13
14  constructor(private _http: HttpClient) { }
15
16  110getProducts(): Observable<IProduct[]> {
17  120    2 return this._http.get<IProduct[]>(this._productUrl)
18  130        .do(data => console.log('All: ' + JSON.stringify(data)))
19  140        .catch(this.handleError);
20  150
21
22  160private handleError(err: HttpErrorResponse) {
23  170    console.log(err.message);
24  180    return Observable.throw(err.message);
25  190
26
27  200
```

```
TS product-list.component.ts x
42 }
43
44 toggleImage(): void {
45     this.showImage = !this.showImage;
46 }
47
48 ngOnInit(): void {
49     this._productService.getProducts()
50         .subscribe(2 (products => this.products = products,
51             error => this.errorMessage = <any>err
52             this.filteredProducts = this.products;
53 )
54
55

TS product.service.ts x
10   jectable()
11   ort class ProductService {
12       private _productUrl = './api/products/products.json';
13
14   constructor(private _http: HttpClient) { }
15
16   getProducts(): Observable<IProduct[]> {
17       return this._1 get<IProduct[]>(this._productUr
18           .do(data => console.log('All: ' + JSON.stringify(data)))
19           .catch(this.handleError);
20   }
21
22   private handleError(err: HttpErrorResponse) {
23       console.log(err.message);
24       return Observable.throw(err.message);
25   }
26
27
```

ts product-list.component.ts x

```
42  }
43
44 toggleImage(): void {
45   this.showImage = !this.showImage;
46 }
47
48 ngOnInit(): void {
49   this._productService.getProducts()
50     .subscribe(products => this.products = products,
51               error => this.errorMessage = <any>err);
52   this.filteredProducts = this.products;
53 }
```

ts product.service.ts x

```
10  jectable()
11  port class ProductService {
12    private _productUrl = './api/products/products.json';
13
14    constructor(private _http: HttpClient) { }
15
16    getProducts(): Observable<IProduct[]> {
17      return this._http.get<IProduct[]>(this._productUrl)
18        .do(data => console.log('All: ' + JSON.stringify(data)))
19        .catch(this.handleError);
20    }
21
22    private handleError(err: HttpErrorResponse) {
23      console.log(err.message);
24      return Observable.throw(err.message);
25    }
26
27
```

```
product-list.component.ts
42 }
43
44 toggleImage(): void {
45     this.showImage = !this.showImage;
46 }
47
48 ngOnInit(): void {
49     this._productService.getProducts()
50         .subscribe(products => {
51             this.products = products;
52             this.filteredProducts = this.products;
53         },
54             error => this.errorMessage = <any>error);
55 }
56
57
```

```
product.service.ts
10   jectable()
11   ort class ProductService {
12       private _productUrl = './api/products/products.json';
13
14   constructor(private _http: HttpClient) { }
15
16   getProducts(): Observable<IProduct[]> {
17       return this._http.get<IProduct[]>(this._productUrl)
18           .do(data => console.log('All: ' + JSON.stringify(data)))
19           .catch(this.handleError);
20   }
21
22   private handleError(err: HttpErrorResponse) {
23       console.log(err.message);
24       return Observable.throw(err.message);
25   }
26
27
```

Filtered by:

Show Image	Product	Code	Available	Price	5 Star Rating
	Leaf Rake	gdn 0011	March 19, 2016	\$19.95	★★★
	Garden Cart	gdn 0023	March 18, 2016	\$32.99	★★★★★
	Hammer	tbx 0048	May 21, 2016	\$8.90	★★★★★
	Saw	tbx 0022	May 15, 2016	\$11.55	★★★★★
	Video Game Controller	gmg 0042	October 15, 2015	\$35.95	★★★★★

```
product-list.component.ts
42 }
43
44 toggleImage(): void {
45     this.showImage = !this.showImage;
46 }
47
48 ngOnInit(): void {
49     this._ productService.getProducts()
50         .subscribe(products => {
51             this.products = products;
52             this.filteredProducts = this.products;
53         },
54         error => this.errorMessage = <any>error);
55 }
56
57
```

```
product.service.ts
10  @Injectable()
11  export class ProductService {
12      private _productUrl = './api/products/products.json';
13
14      constructor(private _ http: HttpClient) { }
15
16      getProducts(): Observable<IProduct[]> {
17          return this._ http.get<IProduct[]>(this._productUrl)
18              .do(data => console.log('All: ' + JSON.stringify(data)))
19              .catch(this.handleError);
20      }
21
22      private handleError(err: HttpErrorResponse) {
23          console.log(err.message);
24          return Observable.throw(err.message);
25      }
26
27
```

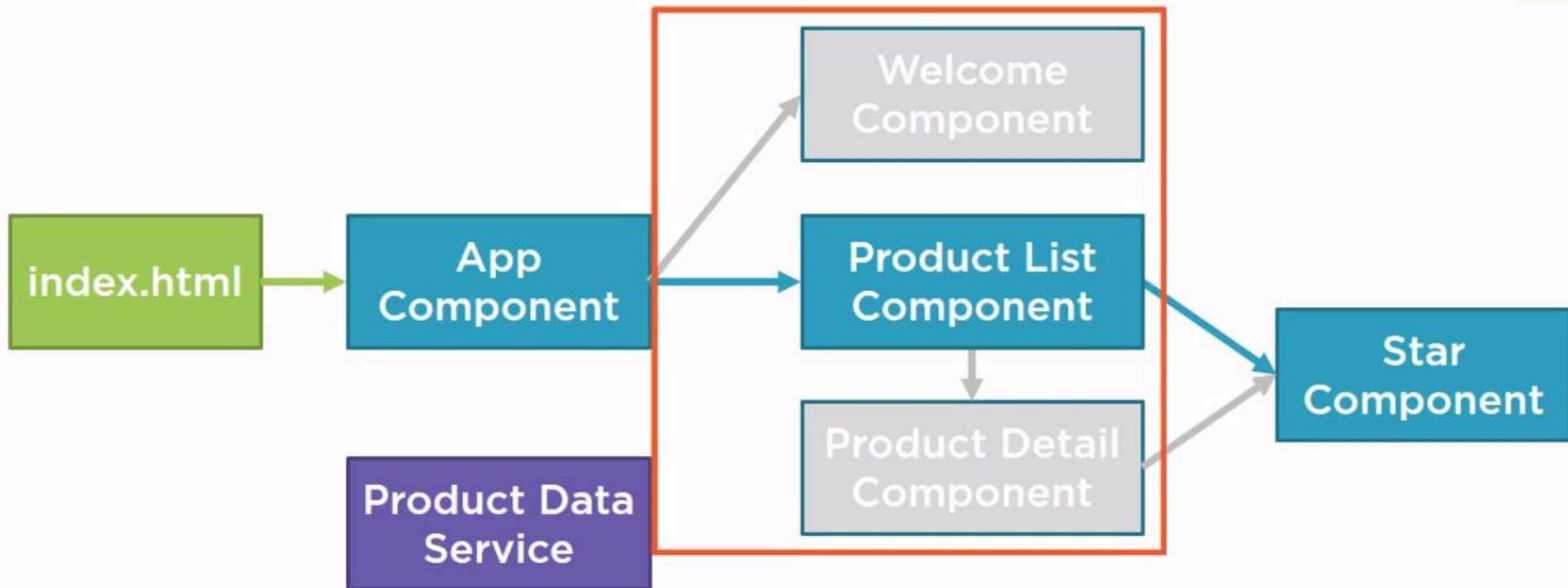
Navigation and Routing

- How does Routing work ?

- Configuring Routes

- Tying routes to Actions

- Placing the views



Product Detail: Video Game Controller

Name: Video Game Controller
Code: GMG-0042
Description: Standard two-button video game controller
Availability: October 15, 2015
Price: \$35.95
5 Star Rating: ★★★★☆



◀ Back

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

1: pow

Windows PowerShell

Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\Deborah\Pluralsight\Angular Getting Started\APM> `npm install -g @angular/cli` █

PS C:\Users\Deborah\Pluralsight\Angular Getting Started\APM> `ng g c products/product-detail.component --flat`
installing component

```
create  src\app\products\product-detail.component.css
create  src\app\products\product-detail.component.html
create  src\app\products\product-detail.component.spec.ts
create  src\app\products\product-detail.component.ts
update  src\app\app.module.ts
```

PS C:\Users\Deborah\Pluralsight\Angular Getting Started\APM> █

- ▶ home
- ◀ products
 - # product-detail.component.css
 - ↳ product-detail.component.html
 - TS product-detail.component.spec.ts
 - TS product-detail.component.ts
 - # product-list.component.css
 - ↳ product-list.component.html
 - TS product-list.component.ts
 - TS product.service.ts
 - TS product.ts
- ▶ shared

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

1: powershell.exe

Windows PowerShell

Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\Deborah\Pluralsight\Angular Getting Started\APM> `ng g c products/product-detail.component --flat`
installing component

```
create  src\app\products\product-detail.component.css
create  src\app\products\product-detail.component.html
create  src\app\products\product-detail.component.spec.ts
create  src\app\products\product-detail.component.ts
update  src\app\app.module.ts
```

PS C:\Users\Deborah\Pluralsight\Angular Getting Started\APM> █

Interpolation

```
<> product-detail.component.html ✘
```

```
1 <div class='panel panel-primary'>
2   <div class='panel-heading'>
3     {{pageTitle + ': ' + product.productName}}
4   </div>
5 </div>
6 |
```

```
<> product-detail.component.html ✘
```

```
1 <div class='panel panel-primary'>
2   <div class='panel-heading'>
3     {{pageTitle + ': ' + product.productName}}
4   </div>
5 </div>
```

undefined

```
<> product-detail.component.html ✘  
1   <div class='panel panel-primary'>  
2     <div class='panel-heading'>  
3       {{pageTitle + ': ' + product?.productName}}  
4     </div>  
5   </div>
```

safe
navigation
operator

```
<> product-detail.component.html ✘  
1   <div class='panel panel-primary'>  
2     <div class='panel-heading'>  
3       {{pageTitle + ': ' + product?.productName}}  
4     </div>  
5   </div>
```

~~[(ngModel)]='product?.productName'~~

```
<> product-detail.component.html <pre>
1   <div class='panel panel-primary'>
2     <div class='panel-heading'>
3       {{pageTitle + ': ' + product?.productName}}
4     </div>
5   </div>
<div class='col-md-6'>
  <div class='row'>
    <div class='col-md-3'>Name:</div>
    <div class='col-md-6'>{{product?.productName}}</div>
  </div>
  <div class='row'>
    <div class='col-md-3'>Code:</div>
    <div class='col-md-6'>{{product?.productCode}}</div>
  </div>
  <div class='row'>
    <div class='col-md-3'>Description:</div>
    <div class='col-md-6'>{{product?.description}}</div>
  </div>
  <div class='row'>
    <div class='col-md-3'>Availability:</div>
    <div class='col-md-6'>{{product?.releaseDate}}</div>
  </div>
  <div class='row'>
    <div class='col-md-3'>Price:</div>
    <div class='col-md-6'>{{product?.price | currency:'USD':true}}</div>
  </div>
</div>
</pre>
```

```
<> product-detail.component.html ✘  
1  <div class='panel panel-primary' *ngIf='product'>  
2    <div class='panel-heading'>  
3      {{pageTitle + ': ' + product.productName}}  
4    </div>  
5  </div>
```

```
<> product-detail.component.html      ts product-detail.component.ts ✘
```

```
1 import { Component, OnInit } from '@angular/core';
2
3 import { IProduct } from './product';  
4
5 @Component({
6   selector: 'pm-product-detail',
7   templateUrl: './product-detail.component.html',
8   styleUrls: ['./product-detail.component.css']
9 })
10 export class ProductDetailComponent implements OnInit {
11   pageTitle: string = 'Product Detail';
12   product: IProduct;  
13
14   constructor() { }
15
16   ngOnInit() {
17   }
18
19 }
20
```

Deleting the selector

```
<> product-detail.component.html      ts product-detail.component.ts x
1  import { Component, OnInit } from '@angular/core';
2
3  import { IProduct } from './product';
4
5  @Component({
6    selector: 'pm-product-detail',
7    templateUrl: './product-detail.component.html',
8    styleUrls: ['./product-detail.component.css']
9  })
10 export class ProductDetailComponent implements OnInit {
11   pageTitle: string = 'Product Detail';
12   product: IProduct;
13
14   constructor() { }
15
16   ngOnInit() {
17   }
18
19 }
20
```

Contd.,

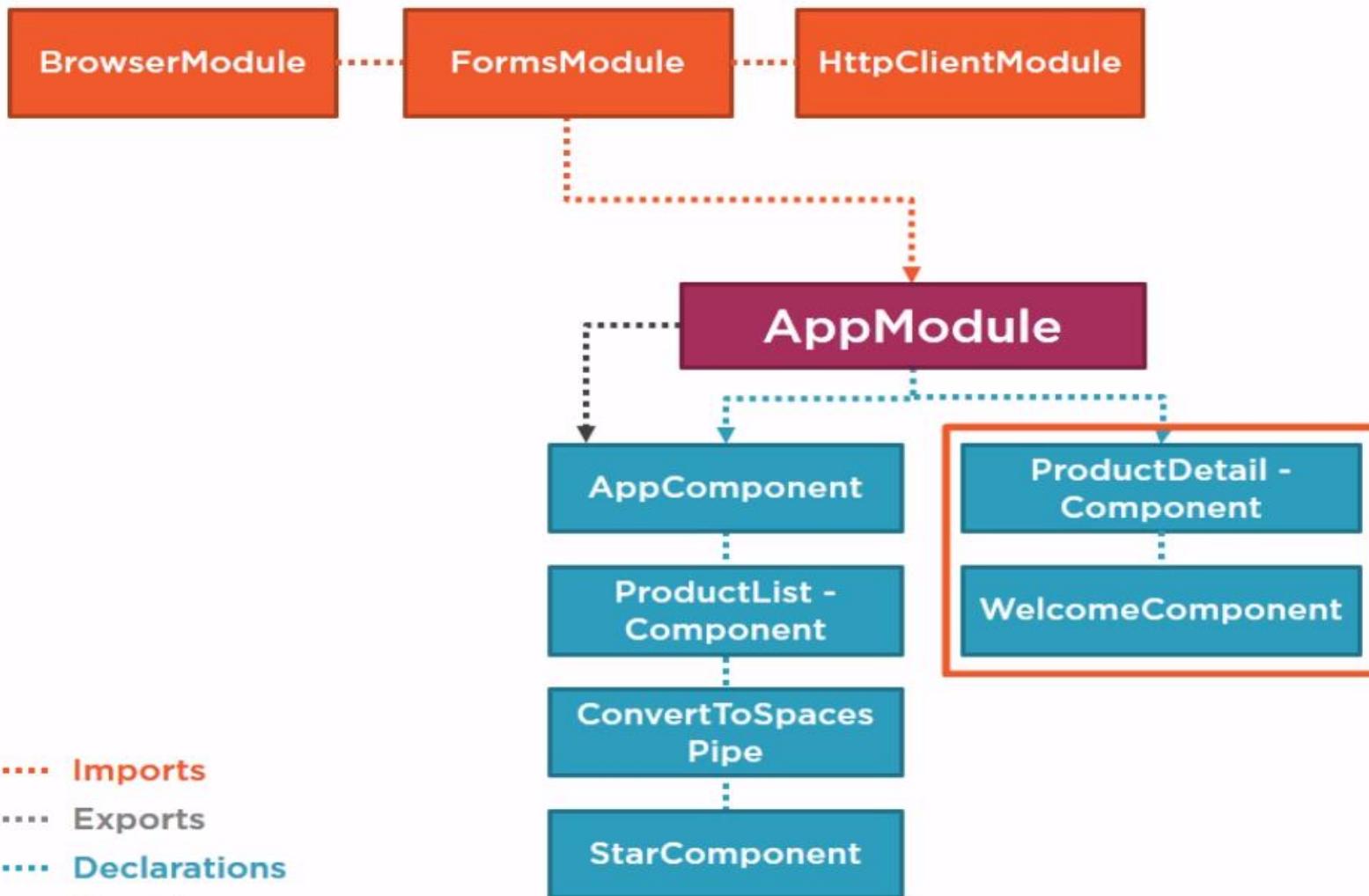
```
TS app.module.ts ✘
15  @NgModule({
16    declarations: [
17      AppComponent,
18      ProductListComponent,
19      ConvertToSpacesPipe,
20      StarComponent,
21      ProductDetailComponent,
22      WelcomeComponent
23    ],
24    imports: [
25      BrowserModule,
26      FormsModule,
27      HttpClientModule,
28      RouterModule.forRoot([
29        { path: 'products', component: ProductListComponent },
30        { path: 'products/:id',
31          canActivate: [ ProductGuardService ],
32          component: ProductDetailComponent },
33        { path: 'welcome', component: WelcomeComponent },
34        { path: '', redirectTo: 'welcome', pathMatch: 'full' },
35        { path: '**', redirectTo: 'welcome', pathMatch: 'full' }
36      ])
37    ],
38    providers: [ProductGuardService],
39    bootstrap: [AppComponent]
40  })
41  export class AppModule {}
```

product-detail.component.html

TS product-detail.component.ts

TS app.module.ts ✘

```
1 import { BrowserModule } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3 import { FormsModule } from '@angular/forms';
4 import { HttpClientModule } from '@angular/common/http';
5
6 import { AppComponent } from './app.component';
7 import { ProductListComponent } from './products/product-list.component';
8 import { ConvertToSpacesPipe } from './shared/convert-to-spaces.pipe';
9 import { StarComponent } from './shared/star-component';
10 import { ProductDetailComponent } from './products/product-detail.component';
11
12 @NgModule({
13   declarations: [
14     AppComponent,
15     ProductListComponent,
16     ConvertToSpacesPipe,
17     StarComponent,
18     ProductDetailComponent
19   ],
20   imports: [
21     BrowserModule,
22     FormsModule,
23     HttpClientModule
24   ],
25   providers: [],
26   bootstrap: [AppComponent]
```



- **Imports**
- **Exports**
- **Declarations**
- **Providers**
- **Bootstrap**

How Routing works

```
▼<pm-app>
  ▼<div>
    ►<nav class="navbar navbar-default">...</nav>
    ▼<div class="container">
      ::before
      <router-outlet></router-outlet>
      ▼<ng-component _ngcontent-jfk-3>
        ▼<div _ngcontent-jfk-3 class="panel panel-primary">
          <div _ngcontent-jfk-3 class="panel-heading">
            Product List
          </div>
          ▼<div _ngcontent-jfk-3 class="panel-body">
            ::before
            ►<div _ngcontent-jfk-3 class="row">...</div>
            ▼<div _ngcontent-jfk-3 class="table-responsive">
              ▼<table _ngcontent-jfk-3 class="table">
                ►<thead _ngcontent-jfk-3>...</thead>
                ►<tbody _ngcontent-jfk-3>...</tbody>
              </table>
            </div>
            ::after
            </div>
          </div>
        </ng-component>
      ::after
    </div>
  </div>
</pm-app>
```

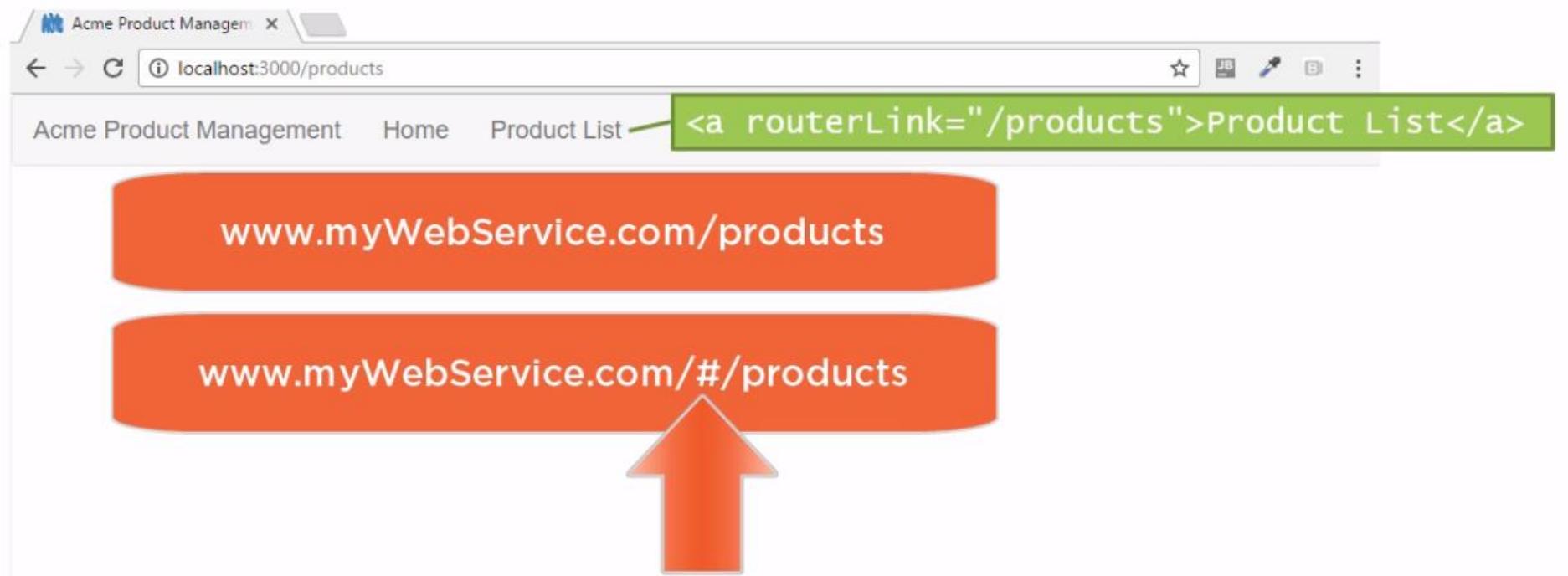
Configure a route for each component

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



How routing works

Acme Product Management x

localhost:3000/products

Acme Product Management Home Product List

Product List

```
{ 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 { }
```

Contd.,

Acme Product Management X

localhost:3000/products

Acme Product Management Home Product List Product List

Product List { path: 'products', component: ProductListComponent }

Filter by:

Show Image	Product	Code	Available	Price	Rating
	Leaf Rake	gdn-0011	March 19, 2016	\$11.55	★★★★
	Garden Cart	gdn-0023	March 18, 2016	\$11.55	★★★★
	Hammer	tbx-0048	May 21, 2016	\$11.55	★★★★
	Saw	tbx-0022	May 15, 2016	\$11.55	★★★★
	Video Game Controller	gmg-0042	October 15, 2015	\$35.95	★★★★★

product-list.component.ts

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

Configuring Routes

app.module.ts

```
...  
import { RouterModule } from '@angular/router';  
  
@NgModule({  
  imports: [  
    BrowserModule,  
    FormsModule,  
    HttpClientModule,  
    RouterModule  
  ],  
  declarations: [  
    ...  
  ],  
  bootstrap: [ AppComponent ]  
})  
export class AppModule { }
```



**Registers the router service
Declares the router directives
Exposes configured routes**

Contd.,

app.module.ts

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

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

Contd.

,

www.myWebService.com/hello

[

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

]

index.html x TS app.module.ts

```
1  <!doctype html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8">
5      <title>Acme Product Management</title>
6      <base href="/" style="border: 2px solid red; padding: 2px; display: inline-block;">
7
8      <meta name="viewport" content="width=device-width, initial-scale=1">
9      <link rel="icon" type="image/x-icon" href="favicon.ico">
10     </head>
11     <body>
12         <pm-root></pm-root>
13     </body>
14     </html>
15 
```

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { HttpClientModule } from '@angular/common/http';
import { RouterModule } from '@angular/router';

import { AppComponent } from './app.component';
import { ProductListComponent } from './products/product-list.component';
import { ConvertToSpacesPipe } from './shared/convert-to-spaces.pipe';
import { StarComponent } from './shared/star.component';
import { ProductDetailComponent } from './products/product-detail.component';
import { WelcomeComponent } from './home/welcome.component';

@NgModule({
  declarations: [
    AppComponent,
    ProductListComponent,
    ConvertToSpacesPipe,
    StarComponent,
    ProductDetailComponent,
    WelcomeComponent
  ],
  imports: [
    BrowserModule,
    FormsModule,
    HttpClientModule
  ]
})
```

```
<> index.html      TS app.module.ts ✘
14  @NgModule({
15    declarations: [
16      AppComponent,
17      ProductListComponent,
18      ConvertToSpacesPipe,
19      StarComponent,
20      ProductDetailComponent,
21      WelcomeComponent
22    ],
23    imports: [
24      BrowserModule,
25      FormsModule,
26      HttpClientModule,
27      RouterModule.forRoot([
28        { path: 'products', component: ProductListComponent },
29        { path: 'products/:id', component: ProductDetailComponent }
30      ])
31    ],
32    providers: [],
33    bootstrap: [AppComponent]
34  })
35  export class AppModule { }
36
```

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 Action

Acme Product Management

Home

Product List

Welcome

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>
  `
})
```

TS app.component.ts ×

TS product-list.component.ts

```
1 import { Component } from '@angular/core';
2
3 import { ProductService } from './products/product.service';
4
5 @Component({
6   selector: 'pm-root',
7   template: `
8     <div><h1>{{pageTitle}}</h1>
9     |   <pm-products></pm-products>
10    </div>
11    `,
12   providers: [ ProductService ]
13 })
14 export class AppComponent {
15   pageTitle: string = 'Acme Product Management';
16 }
17
```

TS app.component.ts X

TS product-list.component.ts

```
1 import { Component } from '@angular/core';
2
3 import { ProductService } from './products/product.service';
4
5 @Component({
6   selector: 'pm-root',
7   template: `
8     <div>
9       <nav class='navbar navbar-default'>
10         <div class='container-fluid'>
11           <a class='navbar-brand'>{{pageTitle}}</a>
12           <ul class='nav navbar-nav'>
13             <li><a>Home</a></li>
14             <li><a>Product List</a></li>
15           </ul>
16         </div>
17       </nav>
18     </div>
19   `,
20   providers: [ ProductService ]
21 })
22 export class AppComponent {
23   pageTitle: string = 'Acme Product Management';
24 }
25
```



```
1 import { Component } from '@angular/core';
2
3 import { ProductService } from './products/product.service';
4
5 @Component({
6   selector: 'pm-root',
7   template: `
8     <div>
9       <nav class='navbar navbar-default'>
10         <div class='container-fluid'>
11           <a class='navbar-brand'>{{pageTitle}}</a>
12           <ul class='nav navbar-nav'>
13             <li><a [routerLink]=["'/welcome']>Home</a></li>
14             <li><a [routerLink]=["'/products']>Product List</a></li>
15           </ul>
16         </div>
17       </nav>
18     </div>
19   `,
20   providers: [ ProductService ]
21 })
22 export class AppComponent {
23   pageTitle: string = 'Acme Product Management';
24 }
25
```

TS app.component.ts ✘

```
1 import { Component } from '@angular/core';
2
3 import { ProductService } from './products/product.service';
4
5 @Component({
6   selector: 'pm-root',
7   template: `
8     <div>
9       <nav class='navbar navbar-default'>
10         <div class='container-fluid'>
11           <a class='navbar-brand'>{{pageTitle}}</a>
12           <ul class='nav navbar-nav'>
13             <li><a [routerLink]="/welcome">Home</a></li>
14             <li><a [routerLink]="/products">Product List</a></li>
15           </ul>
16         </div>
17       </nav>
18     </div>
19   `,
20   providers: [ ProductService ]
21 })
22 export class AppComponent {
23   pageTitle: string = 'Acme Product Management';
24 }
25
```

Placing the views

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>
  `})

```

```
1 import { Component } from '@angular/core';
2
3 import { ProductService } from './products/product.service';
4
5 @Component({
6   selector: 'pm-root',
7   template: `
8     <div>
9       <nav class='navbar navbar-default'>
10         <div class='container-fluid'>
11           <a class='navbar-brand'>{{pageTitle}}</a>
12           <ul class='nav navbar-nav'>
13             <li><a [routerLink]="/welcome">Home</a></li>
14             <li><a [routerLink]="/products">Product List</a></li>
15           </ul>
16         </div>
17       </nav>
18       <div class='container'>
19         <router-outlet></router-outlet>
20       </div>
21     </div>
22   `,
23   providers: [ ProductService ]
24 })
25 export class AppComponent {
26   pageTitle: string = 'Acme Product Management';
27 }
```

localhost:3000/products

Product Management Home Product List

Product List

Filter by:

Filtered by:

Show Image	Product	Code	Available	Price	5 Star Rating
	Leaf Rake	gdn-0011	March 19, 2016	\$19.95	★★★
	Garden Cart	gdn-0023	March 18, 2016	\$32.99	★★★★
	Hammer	tbx-0048	May 21, 2016	\$8.90	★★★★★
	Saw	tbx-0022	May 15, 2016	\$11.55	★★★★★
	Video Game Controller	gmg-0042	October 15, 2015	\$35.95	★★★★★

Acme Product Management X

localhost:3000/products

Acme Product Management Home Product List

<a [routerLink]="/products">Product List

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

product-list.component.ts

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

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

How Routing works

Acme Product Management X

localhost:3000/products

Acme Product Management Home Product List

< a [routerLink]="/products">Product List

Product List { path: 'products', component: ProductListComponent }

Filter by:

Show Image	Product	Code	Available	Price	Rating
	Leaf Rake	gdn-0011	March 19, 2016	\$11.55	★★★★
	Garden Cart	gdn-0023	March 18, 2016		
	Hammer	tbx-0048	May 21, 2016		
	Saw	tbx-0022	May 15, 2016	\$35.95	★★★★★
	Video Game Controller	gmg-0042	October 15, 2015		

product-list.component.ts

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

Displaying Components

Nestable components

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

Routed components

- No selector
- Configure routes
- Tie routes to actions

Doing Routing

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

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

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

Add the RouterOutlet directive

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

Passing Parameters to a Route

The screenshot shows a web browser window titled "Acme Product Management". The address bar displays the URL "http://localhost:3000/product/5", which is highlighted with a red rectangle. The main content area is titled "Product Detail: 5: Hammer". It contains the following product information:

Name:	Hammer
Code:	TBX-0048
Description:	Curved claw steel hammer
Availability:	May 21, 2016
Price:	\$8.9
5 Star Rating:	★★★★★

Next to the text, there is a large, detailed illustration of a hammer with a blue handle and a silver head.

[Back](#)

app.module.ts

```
@NgModule({
  imports: [
    ...,
    RouterModule.forRoot([
      { path: 'products', component: ProductListComponent },
      { path: 'products/:id', component: ProductDetailComponent },
      { path: 'welcome', component: WelcomeComponent },
      { path: '', redirectTo: 'welcome', pathMatch: 'full' },
      { path: '**', redirectTo: 'welcome', pathMatch: 'full' }
    ])
  ],
  declarations: [...],
  bootstrap: [ AppComponent ]
})
export class AppModule { }
```

Contd.,

product-list.component.html

```
<td>
  <a [routerLink]=["/products", product.productId]>
    {{product.productName}}
  </a>
</td>
```

app.module.ts

```
{ path: 'products/:id', component: ProductDetailComponent }
```

Reading parameters from a Route

product-detail.component.ts

```
import { ActivatedRoute } from '@angular/router';

constructor(private _route: ActivatedRoute) {
  console.log(this._route.snapshot.paramMap.get('id'));
}
```

app.module.ts

```
{ path: 'products/:id', component: ProductDetailComponent }
```

Snapshot to retrieve the initial value of the Parameter

product-detail.component.ts

```
import { ActivatedRoute } from '@angular/router';

constructor(private route: ActivatedRoute) {
  console.log(this._route.snapshot.paramMap.get('id'));
}
```

app.module.ts

```
{ path: 'products/:id', component: ProductDetailComponent }
```

Contd.,

product-detail.component.ts

```
import { ActivatedRoute } from '@angular/router';

constructor(private route: ActivatedRoute) {
  console.log(this._route.snapshot.paramMap.get('id'));
}
```

app.module.ts

```
{ path: 'products/:id', component: ProductDetailComponent }
```

Contd.,

```
TS app.module.ts ✘  ↵ product-list.component.html  TS product-detail.component.ts

14  @NgModule({
15    declarations: [
16      AppComponent,
17      ProductListComponent,
18      ConvertToSpacesPipe,
19      StarComponent,
20      ProductDetailComponent,
21      WelcomeComponent
22    ],
23    imports: [
24      BrowserModule,
25      FormsModule,
26      HttpClientModule,
27      RouterModule.forRoot([
28        { path: 'products', component: ProductListComponent },
29        { path: 'products/:id', component: ProductDetailComponent }, // Line 29
30        { path: 'welcome', component: WelcomeComponent },
31        { path: '', redirectTo: 'welcome', pathMatch: 'full' },
32        { path: '**', redirectTo: 'welcome', pathMatch: 'full' }
33      ])
34    ],
35    providers: [],
36    bootstrap: [AppComponent]
37  })
38  export class AppModule { }
```

Contd.,

```
TS app.module.ts      <> product-list.component.html x  TS product-detail.component.ts

33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
```

```
</tr>
</thead>
<tbody>
    <tr *ngFor='let product of filteredProducts'>
        <td>
            <img *ngIf='showImage'
                [src]='product.imageUrl'
                [title]='product.productName'
                [style.width.px]='imageWidth'
                [style.margin.px]='imageMargin'>
        </td>
        <td><a [routerLink]=["'/products', product.productId"]>
            {{ product.productName }}</a></td>
        <td>{{ product.productCode | lowercase | convertToSpaces: '-' }}</td>
        <td>{{ product.releaseDate }}</td>
        <td>{{ product.price | currency:'USD':true:'1.2-2' }}</td>
        <td>
            <pm-star [rating]='product.starRating'
                (ratingClicked)='onRatingClicked($event)'>
            </pm-star>
        </td>
    </tr>
</tbody>
</table>
</div>
```

Product List

Filter by:

Filtered by:

Show Image	Product	Code	Available	Price	5 Star Rating
	Leaf Rake	gdn 0011	March 19, 2016	\$19.95	★★★
	Garden Cart	gdn 0023	March 18, 2016	\$32.99	★★★★★
	Hammer	tbx 0048	May 21, 2016	\$8.90	★★★★★
	Saw	tbx 0022	May 15, 2016	\$11.55	★★★★★
	Video Game Controller	gmg 0042	October 15, 2015	\$35.95	★★★★★

Product List

Filter by:

Filtered by:

Show Image	Product	Code	Available	Price	5 Star Rating
	Leaf Rake	gdn 0011	March 19, 2016	\$19.95	★★★
	Garden Cart	gdn 0023	March 18, 2016	\$32.99	★★★★
	Hammer	tbx 0048	May 21, 2016	\$8.90	★★★★★
	Saw	tbx 0022	May 15, 2016	\$11.55	★★★★
	<u>Video Game Controller</u>	gmg 0042	October 15, 2015	\$35.95	★★★★★



File Edit View Go Debug Help

TS app.module.ts

<> product-list.component.html

TS product-detail.component.ts ×

```
1 import { Component, OnInit } from '@angular/core';
2 import { ActivatedRoute } from '@angular/router';
3
4 import { IProduct } from './product';
5
6 @Component({
7   templateUrl: './product-detail.component.html',
8   styleUrls: ['./product-detail.component.css']
9 })
10 export class ProductDetailComponent implements OnInit {
11   pageTitle: string = 'Product Detail';
12   product: IProduct;
13
14   constructor() { }
15
16   ngOnInit() {
17   }
18
19 }
20
```

```
TS app.module.ts      <> product-list.component.html      TS product-detail.component.ts ✘
1 import { Component, OnInit } from '@angular/core';
2 import { ActivatedRoute } from '@angular/router';
3
4 import { IProduct } from './product';
5
6 @Component({
7   templateUrl: './product-detail.component.html',
8   styleUrls: ['./product-detail.component.css']
9 })
10 export class ProductDetailComponent implements OnInit {
11   pageTitle: string = 'Product Detail';
12   product: IProduct;
13
14   constructor(private _route: ActivatedRoute) { }
15
16   ngOnInit() {
17     let id = +this._route.snapshot.paramMap.get('id');
18   }
19 }
20
21 }
22
```

+ - to convert string to numeric

```
TS app.module.ts      <> product-list.component.html      TS product-detail.component.ts ✘
1 import { Component, OnInit } from '@angular/core';
2 import { ActivatedRoute } from '@angular/router';
3
4 import { IProduct } from './product';
5
6 @Component({
7   templateUrl: './product-detail.component.html',
8   styleUrls: ['./product-detail.component.css']
9 })
10 export class ProductDetailComponent implements OnInit {
11   pageTitle: string = 'Product Detail';
12   product: IProduct;
13
14   constructor(private _route: ActivatedRoute) { }
15
16   ngOnInit() {
17     let id = +this._route.snapshot.paramMap.get('id');
18
19   }
20
21 }
22
```



TS app.module.ts

product-list.component.html

TS product-detail.component.ts

```
6  @Component({
7    templateUrl: './product-detail.component.html',
8    styleUrls: ['./product-detail.component.css']
9  })
10 export class ProductDetailComponent implements OnInit {
11   pageTitle: string = 'Product Detail';
12   product: IProduct;
13
14   constructor(private _route: ActivatedRoute) { }
15
16   ngOnInit() {
17     let id = +this._route.snapshot.paramMap.get('id');
18     this.pageTitle += `: ${id}`;
19     this.product = {
20       "productId": id,
21       "productName": "Leaf Rake",
22       "productCode": "GDN-0011",
23       "releaseDate": "March 19, 2016",
24       "description": "Leaf rake with 48-inch wooden handle.",
25       "price": 19.95,
26       "starRating": 3.2,
27       "imageUrl": "http://openclipart.org/image/300px/svg_to_png/26215/Anonymous_Leaf_Rake.png"
28     }
29   }
30 }
31 }
```



Acme Product Management Home Product List

Product Detail: 10: Leaf Rake

Activating a route with Code

The screenshot shows a web browser window with the URL <http://localhost:3000/product/5> in the address bar. The title bar says "Acme Product Management". The main content area displays a product detail page for a hammer. The product information is as follows:

Name:	Hammer
Code:	TBX-0048
Description:	Curved claw steel hammer
Availability:	May 21, 2016
Price:	\$8.9
5 Star Rating:	★★★★★

A large image of a hammer is displayed on the right side of the page. At the bottom left, there is a red-bordered button labeled "Back".

Contd.,

product-detail.component.ts

```
import { Router} from '@angular/router';
...
constructor(private _router: Router) { }

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

TS product-detail.component.ts ✘ ⌂ product-detail.component.html

```
1 import { Component, OnInit } from '@angular/core';
2 import { ActivatedRoute, Router } from '@angular/router';
3
4 import { IProduct } from './product';
5
6 @Component({
7   templateUrl: './product-detail.component.html',
8   styleUrls: ['./product-detail.component.css']
9 })
10 export class ProductDetailComponent implements OnInit {
11   pageTitle: string = 'Product Detail';
12   product: IProduct;
13
14   constructor(private _route: ActivatedRoute,
15             private _router: Router) { }
16
17   ngOnInit() {
18     let id = +this._route.snapshot.paramMap.get('id');
19     this.pageTitle += `: ${id}`;
20     this.product = {
21       "productId": id,
22       "productName": "Leaf Rake",
23       "productCode": "GDN-0011",
24       "releaseDate": "March 19, 2016",
25       "description": "Leaf rake with 48-inch wooden handle.",
26       "price": 19.95,
27       "starRating": 3.2
28     }
29   }
30 }
```

TS product-detail.component.ts ✘ ⌂ product-detail.component.html

```
10  export class ProductDetailComponent implements OnInit {
11      pageTitle: string = 'Product Detail';
12      product: IProduct;
13
14      constructor(private _route: ActivatedRoute,
15                  private _router: Router) { }
16
17      ngOnInit() {
18          let id = +this._route.snapshot.paramMap.get('id');
19          this.pageTitle += `: ${id}`;
20          this.product = {
21              "productId": id,
22              "productName": "Leaf Rake",
23              "productCode": "GDN-0011",
24              "releaseDate": "March 19, 2016",
25              "description": "Leaf rake with 48-inch wooden handle.",
26              "price": 19.95,
27              "starRating": 3.2,
28              "imageUrl": "http://openclipart.org/image/300px/svg_to_png/26215/Anonymous_Leaf_Rake.png"
29          }
30      }
31
32      onBack(): void {
33          this._router.navigate ['/products'];
34      }
35  }
```

```
ts product-detail.component.ts    <> product-detail.component.html x
1  <div class='panel panel-primary' *ngIf='product'>
2      <div class='panel-heading'>
3          {{pageTitle + ': ' + product.productName}}
4      </div>
5      <div class='panel-footer'>
6          <a class='btn btn-default' (click)=onBack() style='width:80px'>
7              <i class='glyphicon glyphicon-chevron-left'></i> Back
8          </a>
9      </div>
10 </div>
```

Product List

Filter by:

Filtered by:

Show Image	Product	Code	Available	Price	5 Star Rating
	Leaf Rake	gdn 0011	March 19, 2016	\$19.95	★★★
	Garden Cart	gdn 0023	March 18, 2016	\$32.99	★★★★
	Hammer	tbx 0048	May 21, 2016	\$8.90	★★★★★
	Saw	tbx 0022	May 15, 2016	\$11.55	★★★★
	Video Game Controller	gmg 0042	October 15, 2015	\$35.95	★★★★★

Product Detail: 10: Leaf Rake

Back

```
10  export class ProductDetailComponent implements OnInit {
11      pageTitle: string = 'Product Detail';
12      product: IProduct;
13
14      constructor(private _route: ActivatedRoute,
15                  private _router: Router) { }
16
17      ngOnInit() {
18          let id = +this._route.snapshot.paramMap.get('id');
19          this.pageTitle += `: ${id}`;
20          this.product = {
21              "productId": id,
22              "productName": "Leaf Rake",
23              "productCode": "GDN-0011",
24              "releaseDate": "March 19, 2016",
25              "description": "Leaf rake with 48-inch wooden handle.",
26              "price": 19.95,
27              "starRating": 3.2,
28              "imageUrl": "http://openclipart.org/image/300px/svg_to_png/26215/Anonymous_Leaf_Rake.png"
29          }
30      }
31
32      onBack(): void {
33          this._router.navigate(['/products']);
34      }
35
36  }
```

Thank You For Your Time



People matter, results count.

