

Angular Component Communication

INTRODUCTION



Deborah Kurata

CONSULTANT | SPEAKER | AUTHOR | MVP | GDE

@deborahkurata | blogs.msmvps.com/deborahk/





Component
A

Service

Template
A

Component
B





Component <-> Template:

- View updates when data has changed





Component <-> Template:

- View updates when data has changed
- React to user changes





Component <-> Template:

- View updates when data has changed
- React to user changes
- Ask an element to set a property or perform a task



Is your form
currently valid?

OK, I'll save.

Yep!



Component <-> Template:

- View updates when data has changed
- React to user changes
- Ask an element to set a property or perform a task
- Check form or control state





Will you keep
track of my
current filter?

Sure!
Retaining your
current filter.

What was my
filter?

I have it. Here
you go ...

Component <-> Service <->
Component:

- Retain state

Component
A

Service

Component
B





Could you
get my data?

Sure!
I'll get it, retain it,
and share it.

I need it too.
Thanks for
sharing!

Component
A

Service

Component
B

Component <-> Service <-> Component:

- Retain state
- Share data





Component <-> Service <-> Component:

- Retain state
- Share data
- Send notifications





Component <-> Router <->
Component:

- Pass required data





I may have
search params
for you.

If you provide
them I'll pass
them along.

If I get
params, I'll
use them.

Component
A

Router

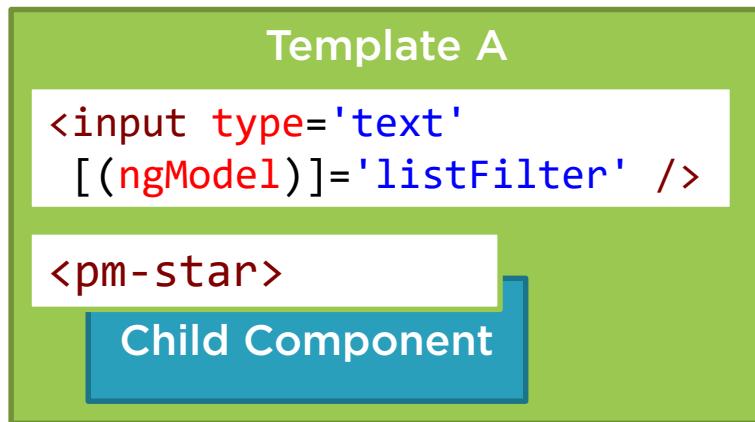
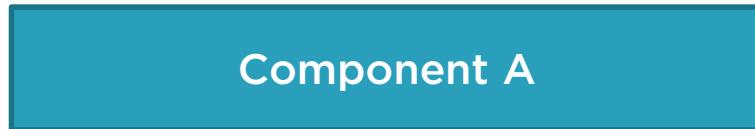
Component
B

Component <-> Router <-> Component:

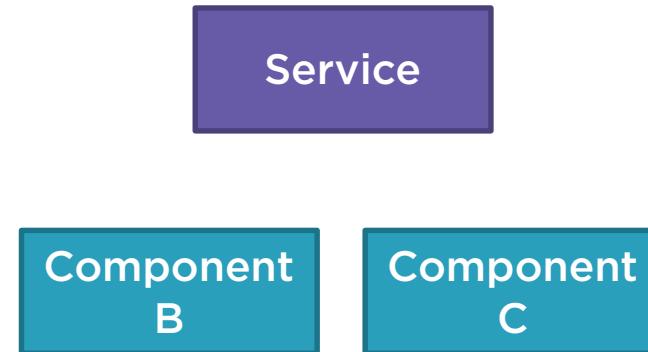
- Pass required data
- Pass optional data



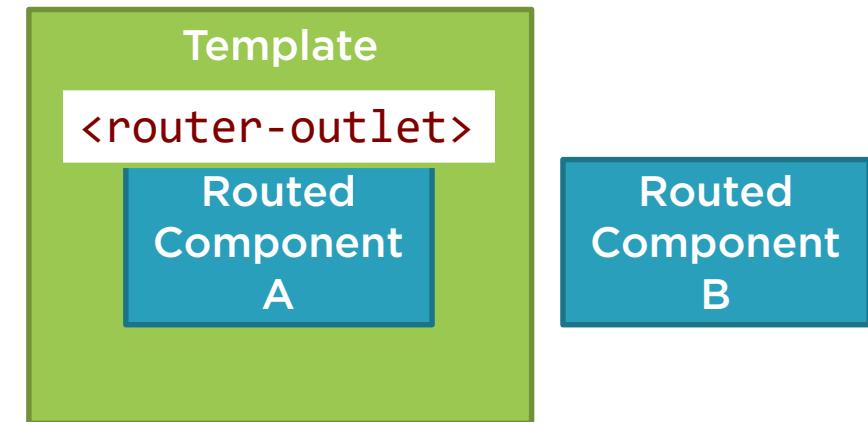
Component Communication



Template



Service



Routing



Module Overview



Getting the Most from This Course
Sample Application
Course Outline



Prerequisites

Required

- Angular modules
- Components
- Templates
- Services
- Routing

Suggested

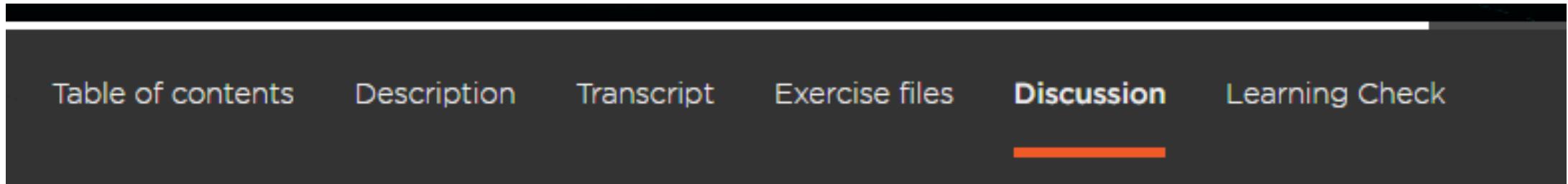
- Angular: Getting Started
- Angular: First Look

Not Required

- Extensive component communication experience



Thoughts? Comments? Questions?



@deborahkurata



Blog Post

[http://blogs.msmvps.com/deborahk/
angular-component-communication-problem-solver/](http://blogs.msmvps.com/deborahk/angular-component-communication-problem-solver/)

Deborah's Developer MindScape

"Angular Component Communication" Problem Solver

Filed under: Angular,Angular Course Problem Solver — deborahk @ 11:34 am

Tags: Angular

This blog post supports the sample code for the "Angular Component Communication" course on Pluralsight, identifying common issues along with their solutions.



Guidelines & Summary



Review module concepts
Which to use when



GitHub Repository

<https://github.com/DeborahK/Angular-Communication>

Materials for my Pluralsight course: Angular Component Communication.

Edit

Add topics

20 commits

1 branch

0 releases

1 contributor

MIT

Branch: master ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾



DeborahK First checkin of completed code.

Latest commit f9025a1 7 minutes ago



APM-Final First checkin of completed code.

7 minutes ago



APM-Start First checkin of completed code.

7 minutes ago



.gitignore Initial checkin

2 months ago



LICENSE Initial commit

2 months ago

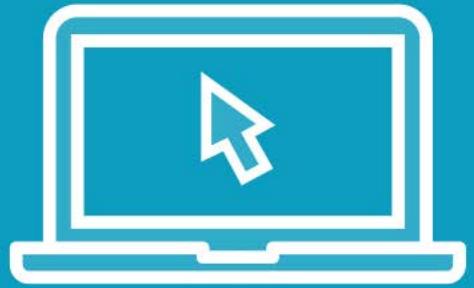


README.md Initial commit

2 months ago



Demo



Sample Application in Action



Number of months on the market: 7

Products

Leaf Rake (GDN-0011)

Garden Cart (GDN-0023)

Hammer (TBX-0048)

Saw (TBX-0022)

Video Game Controller (GMG-0042)

Product Detail: Hammer

Name: Hammer

Code: TBX-0048

Description: Curved claw steel
hammer

Availability: May 21, 2017

Price: \$8.90

5 Star 

Rating:



Edit

Course Outline



Communicating with a Template

ViewChild and ViewChildren

Communicating with a Child Component

Communicating with a Parent Component

Communicating Through a Service

Communicating Through a State Management Service

Communicating Through Service Notifications

Communicating Using the Router



Communicating with a Template



Deborah Kurata

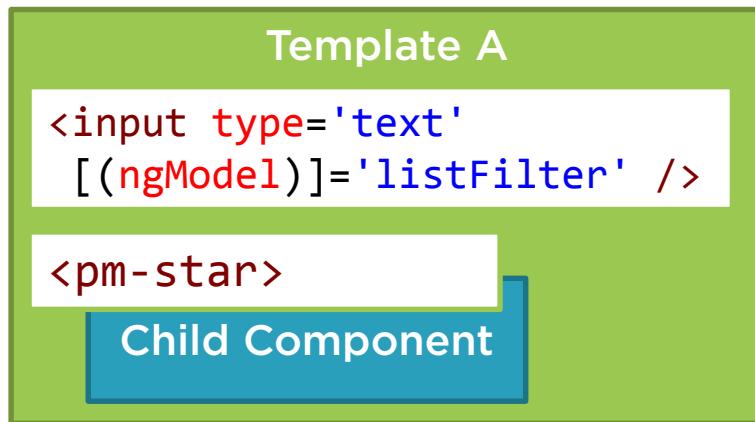
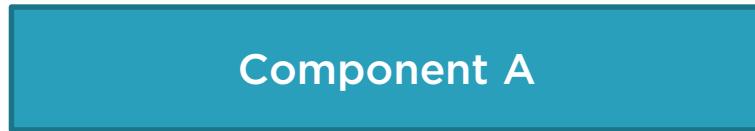
CONSULTANT | SPEAKER | AUTHOR | MVP | GDE

@deborahkurata | blogs.msmvps.com/deborahk/

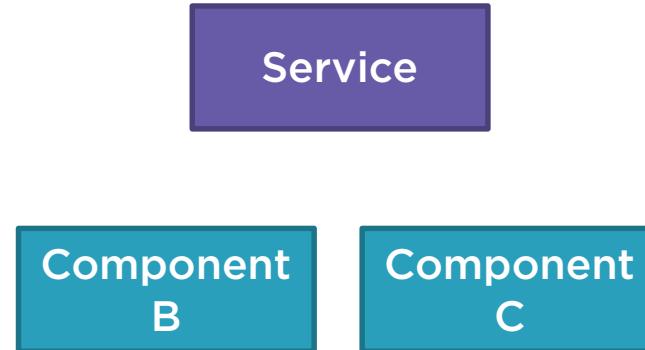




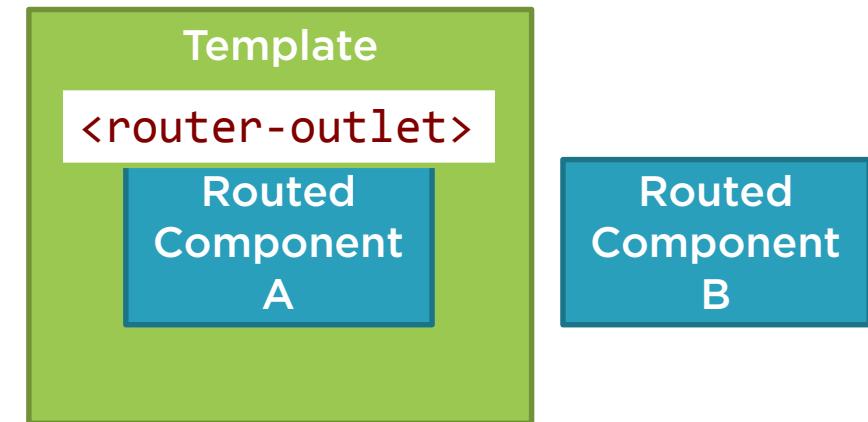
Component Communication



Template



Service



Routing



Module Overview



Binding and Structural Directives

Two-way Binding, the Long Way

Getters and Setters





Binding

Template

```
<div>  
  {{pageTitle}}  
</div>  
  
<img [style.width.px]=  
      'imageWidth'>  
  
<button  
  (click)='toggleImage()'>  
  Show Image  
</button>  
  
<input type='text'  
 [(ngModel)]='listFilter' />
```

Interpolation

Property Binding

Event Binding

Two-way Binding

Component

```
pageTitle: string = 'Product List';
```

```
imageWidth: number = 50;
```

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

```
listFilter: string;
```



Structural Directives

Template

```
<img *ngIf='showImage'  
      [src]='product.imageUrl'>  
  
<tr *ngFor='let product of products'>
```

Component

```
showImage: boolean = false;  
  
products: IProduct[];
```



Two-way Binding

Template

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

Two-way Binding

Component

```
listFilter: string;
```



Two-way Binding

Template

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

Two-way Binding

Component

```
listFilter: string = 'Cart';
```

Product List

Filter by:

Cart



Two-way Binding

Template

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

Two-way Binding

Component

```
listFilter: string = 'Game';
```

Product List

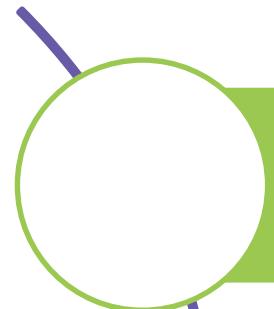
Filter by:

Game

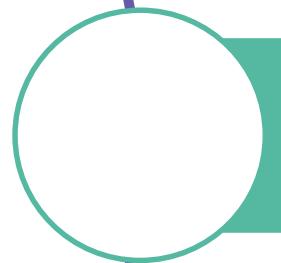




Notifying the Component of User Changes



Two-way binding, the long way



Getter and setter



valueChanges observable



Two-way Binding, the Long Way

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

```
<input type='text' [ngModel]=listFilter  
        (ngModelChange)=listFilter=$event />
```

```
<input type='text' [ngModel]=listFilter  
        (ngModelChange)=onFilterChange($event) />
```



Defining a Property

Property Declaration

```
listFilter: string;
```

Getter and Setter

```
get listFilter(): string {  
}  
  
set listFilter(value: string) {  
}
```



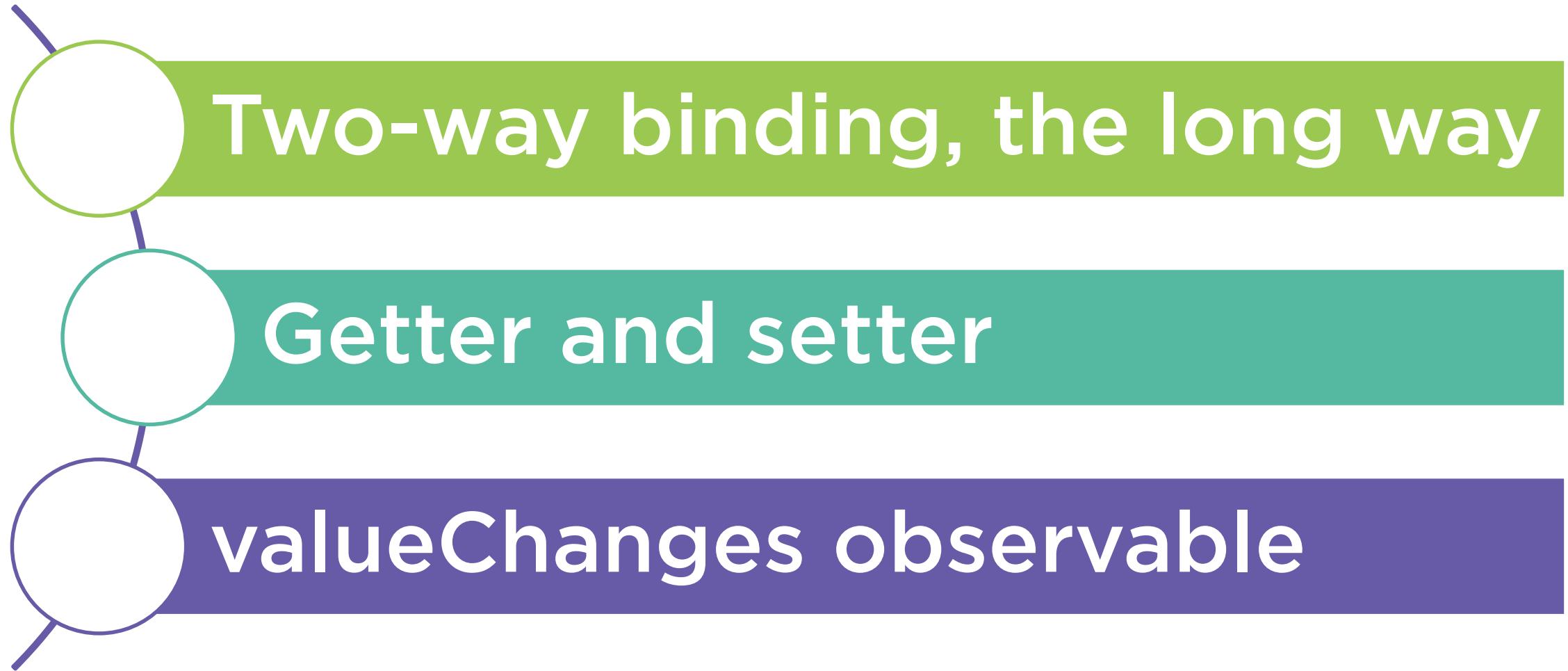
Specifying a Getter and Setter

Common Pattern

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



Notifying the Component of User Changes



Guidelines & Summary: Communicating with a Template



Binding & Structural Directives

Template

```
<div>{{pageTitle}}</div>  
  
<img [style.width.px]=  
      'imageWidth'>  
  
<button  
  (click)='toggleImage()'>  
  Show Image  
</button>  
  
<input type='text'  
 [(ngModel)]='listFilter' />  
  
<img *ngIf='showImage'  
 [src]='product.imageUrl'>  
  
<tr *ngFor=  
 'let product of products'>
```

Interpolation

Property Binding

Event Binding

Two-way Binding

*ngIf

*ngFor

Component

```
pageTitle: string = 'Product List';
```

```
imageWidth: number = 50;
```

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

```
listFilter: string;
```

```
showImage: boolean = false;
```

```
products: IProduct[];
```



Two-way Binding, the Long Way

```
<input type='text' [ngModel]='listFilter'  
        (ngModelChange)='onFilterChange($event)' />
```

Plus:

- Notifies the component when the user changes the value
- Allows any logic in the component method
- Caught in the template

Caveats:

- No two-way binding
- Caught in the template
- Uncommon syntax



Getter and Setter

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

Plus:

- Notifies the component when the user changes the value
- Allows any logic in the setter
- Caught in the component class

Caveats:

- One line of code becomes 7



Summary



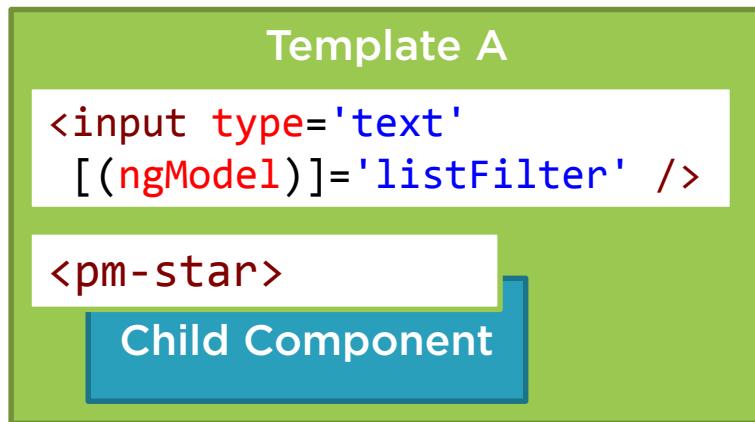
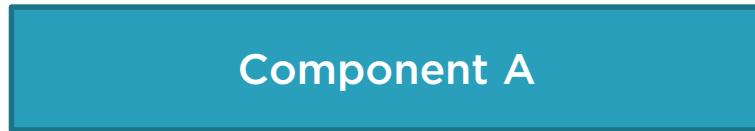
Binding and Structural Directives

Two-way Binding, the Long Way

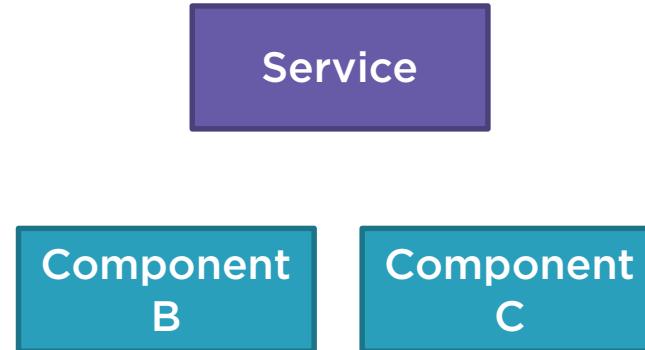
Getters and Setters



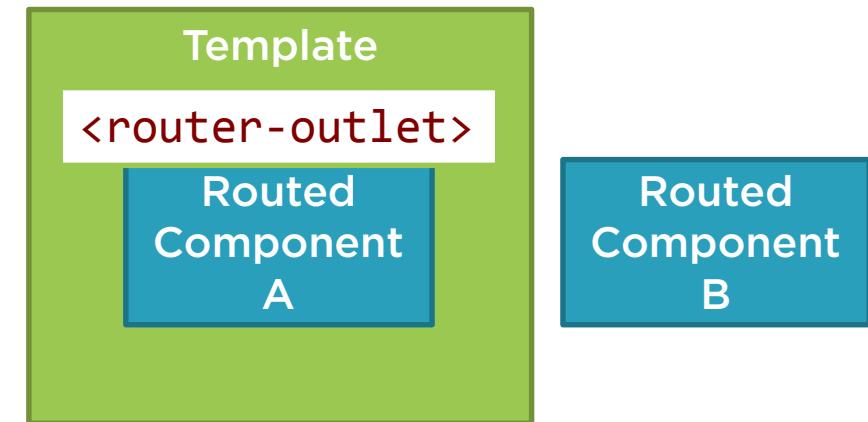
Component Communication



Template



Service



Routing



ViewChild and ViewChildren



Deborah Kurata

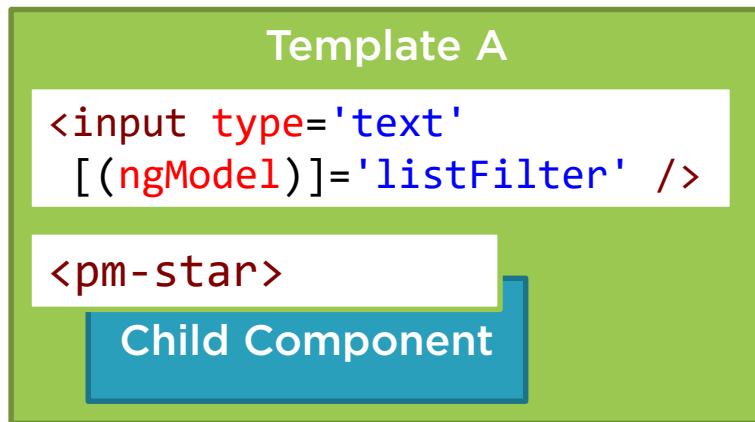
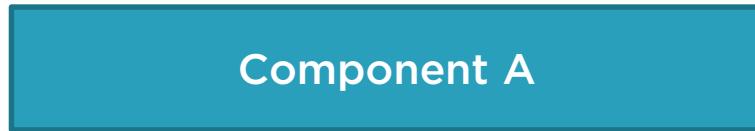
CONSULTANT | SPEAKER | AUTHOR | MVP | GDE

@deborahkurata | blogs.msmvps.com/deborahk/

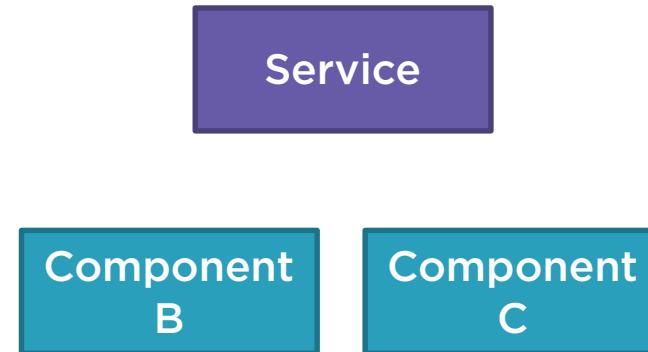




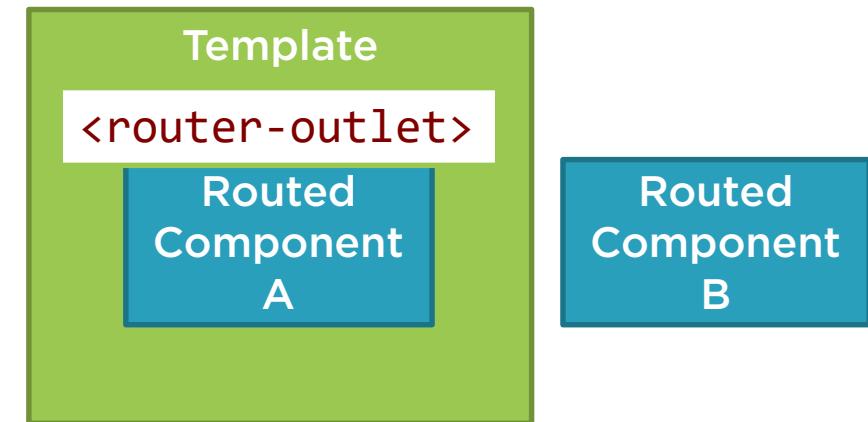
Component Communication



Template



Service



Routing



Module Overview



ViewChild

ViewChildren

ViewChild and Angular Forms

- valueChanges Observable

ViewChild and nglif





Is your form
currently valid?

OK, I'll save.

Yep!



Getting a Reference

DOM

```
let divElement = document.getElementById('divElementId');
```

Decorator

```
@ViewChild('divElementVar') divElementRef;
```



ViewChild

Angular Directive

```
@ViewChild(NgModel) filterInput: NgModel;
```

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

Custom Directive / Child Component

```
@ViewChild(StarComponent) star: StarComponent;
```

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

Template Reference Variable

```
@ViewChild('divElementVar') divElementRef: ElementRef;
```

```
<div #divElementVar>{{pageTitle}}</div>
```



Demo



ViewChild and Accessing the Native Html Element



Considerations When Using nativeElement

**Using nativeElement ->
directly accessing the DOM**

Tightly coupled to the browser

May not be able to use server-side rendering or web workers

```
if (this.filterElementRef.nativeElement) {  
  this.filterElementRef.nativeElement.focus();  
}
```



ViewChildren

```
@ViewChildren('divElementVar')  
divElementRefs: QueryList<ElementRef>;
```

Differences:

- Returns a **QueryList** of element or directive references
- Tracks changes in the DOM

```
this.divElementRefs.changes.subscribe(() => {  
  // Code here  
})
```



ViewChildren

Angular Directive

```
@ViewChildren(NgModel) inputs: QueryList<NgModel>;
```

Custom Directive / Child Component

```
@ViewChildren(StarComponent) stars: QueryList<StarComponent>;
```

Template Reference Variable

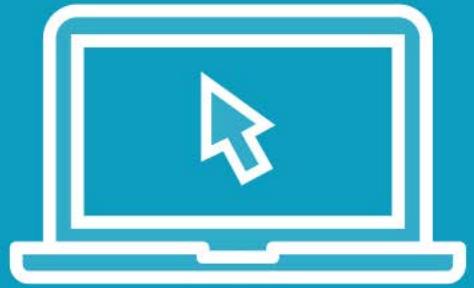
```
@ViewChildren('divElementVar')  
divElementRefs: QueryList<ElementRef>;
```

Template Reference Variables

```
@ViewChildren('filterElement, nameElement')  
divElementRefs: QueryList<ElementRef>;
```



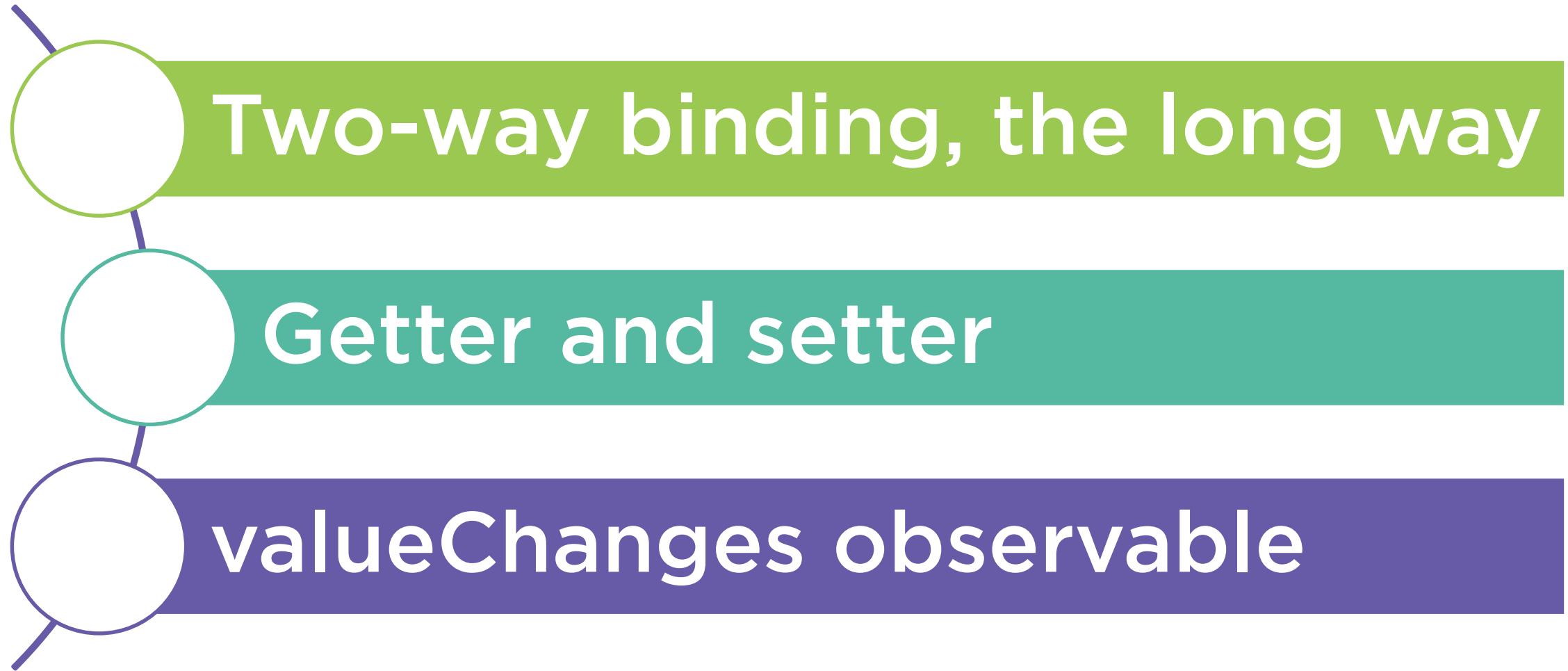
Demo



ViewChildren



Notifying the Component of User Changes



ViewChild and Angular Forms

Template

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

Component

```
@ViewChild(NgModel) filterInput: NgModel;  
  
this.filterInput.valueChanges.subscribe(  
  () => this.performFilter(this.listFilter)  
);
```



Angular Forms

Template-driven

- Angular creates the form data structures
- Based on info in the template
- Access reference with ViewChild

Reactive

- We create the form data structures
- Defined in the component class
- No need for ViewChild

```
this.filterInput.valueChanges.subscribe(  
    () => this.performFilter(this.listFilter)  
)
```



For More Information

Angular Forms

by Mark Zamoya

Angular Reactive Forms

by Deborah Kurata



Template-driven Forms / No Form

```
<form (ngSubmit)="saveProduct()">
```

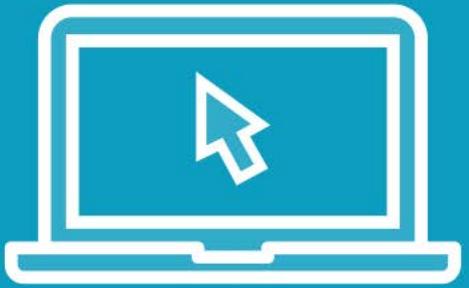
```
▼ NgForm {submitted: false, _d  
  control: (...)  
  ▼ controls: Object  
    ► description: FormControl  
    ► productCode: FormControl  
    ► productName: FormControl  
    ► __proto__: Object  
  dirty: false  
  disabled: (...)  
  enabled: (...)  
  errors: null  
  ► form: FormGroup {validator:  
  ► formDirective: NgForm  
    invalid: (...)  
  ► ngSubmit: EventEmitter {_is  
    path: (...)  
    pending: (...)  
    pristine: (...)  
    status: (...)  
    statusChanges: (...)  
    submitted: false  
    touched: (...)  
    untouched: (...)  
    valid: (...)  
    value: (...)  
    valueChanges: (...)
```

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

```
▼ NgModel {_parent: null,  
  asyncValidator: (...)  
  ► control: FormControl  
  dirty: (...)  
  disabled: (...)  
  enabled: (...)  
  errors: (...)  
  formDirective: null  
  invalid: (...)  
  model: undefined  
  name: null  
  path: (...)  
  pending: (...)  
  pristine: (...)  
  status: (...)  
  statusChanges: (...)  
  touched: (...)  
  untouched: (...)  
  ► update: EventEmitter  
  valid: (...)  
  validator: (...)  
  value: (...)  
  ► valueAccessor: Default  
  valueChanges: (...)  
  viewModel: undefined
```



Demo



ViewChild and Angular Forms



ViewChild and *ngIf

```
<div *ngIf='products'>
  <div>Filter by:</div>
  <div>
    <input type='text' [(ngModel)]=listFilter' />
  </div>
</div>
```

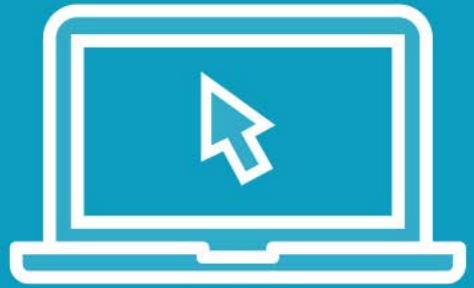
```
@ViewChild(NgModel) filterInput: NgModel;

ngAfterViewInit(): void {
  this.filterInput.valueChanges.subscribe(
    () => this.performFilter(this.listFilter)
  );
}
```

```
@ViewChild(NgModel)
set filterInput(value: NgModel) {
  this._filterInput = value;
}
```



Demo



ViewChild and ngIf



Guidelines & Summary: ViewChild and ViewChildren



ViewChild/ViewChildren: Html Element

```
@ViewChild('divElementVar') divElementRef: ElementRef;
```

Plus:

- Provides a nativeElement property
- Access any Html element properties
- Call any Html element methods

Caveats:

- ViewChild reference not reliably available until AfterViewInit
- ViewChild reference not available if the element is not in the DOM
- Does not work with server-side rendering or web workers

```
if (this.filterElementRef.nativeElement) {  
    this.filterElementRef.nativeElement.focus();  
}
```



ViewChild/ViewChildren: Angular Directive

```
@ViewChild(NgModel) filterInput: NgModel;
```

Plus:

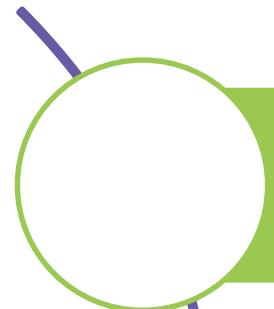
- Provides reference to the directive's data structures
- Access any properties

Caveats:

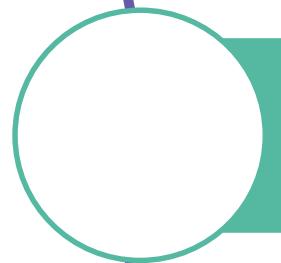
- ViewChild reference not reliably available until AfterViewInit
- ViewChild reference not available if the element is not in the DOM
- NgForm and NgModel data structures are read-only



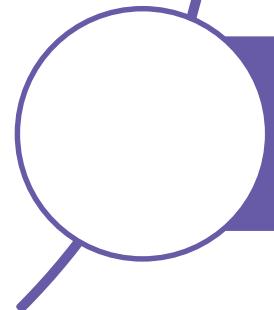
Notifying the Component of User Changes



Two-way binding, the long way



Getter and setter



valueChanges observable



Subscribe to the valueChanges Observable

```
@ViewChild(NgModel) filterInput: NgModel;  
  
this.filterInput.valueChanges.subscribe(  
  () => this.performFilter(this.listFilter)  
);
```

Plus:

- Favor this technique if using other NgModel information

Caveats:

- Watch out for ngIf
- Reference not reliably available until AfterViewInit



Summary



ViewChild

ViewChildren

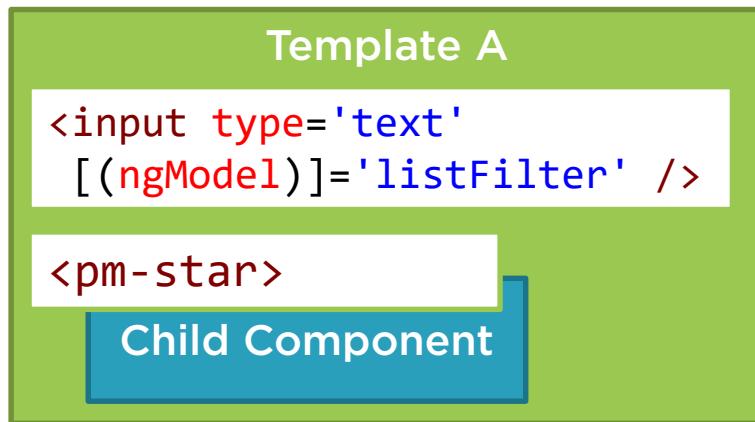
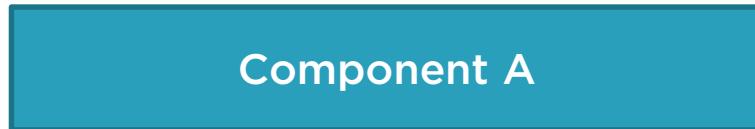
ViewChild and Angular Forms

- valueChanges Observable

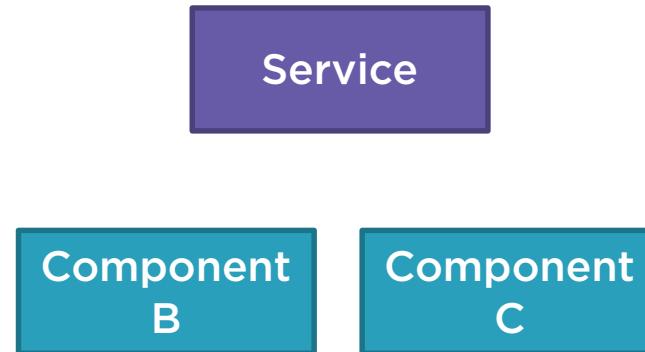
ViewChild and nglif



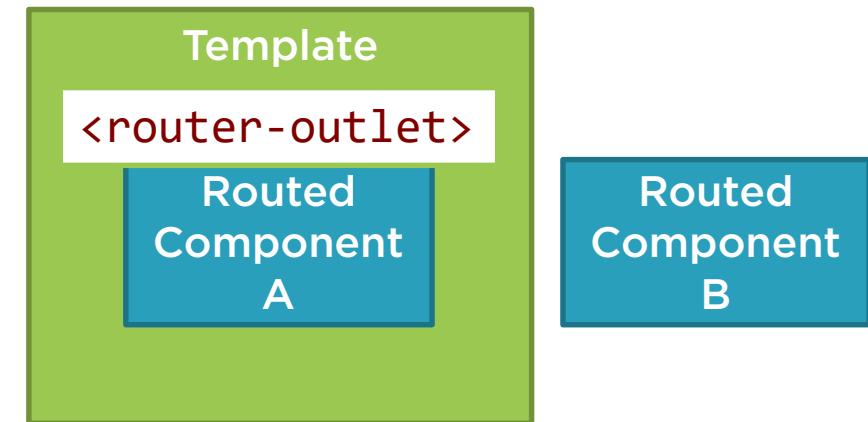
Component Communication



Template



Service



Routing



Communicating with a Child Component



Deborah Kurata

CONSULTANT | SPEAKER | AUTHOR | MVP | GDE

@deborahkurata | blogs.msmvps.com/deborahk/





Child Components

Product List

Filter by:

Filtered by: r

Show Images

Product List Template

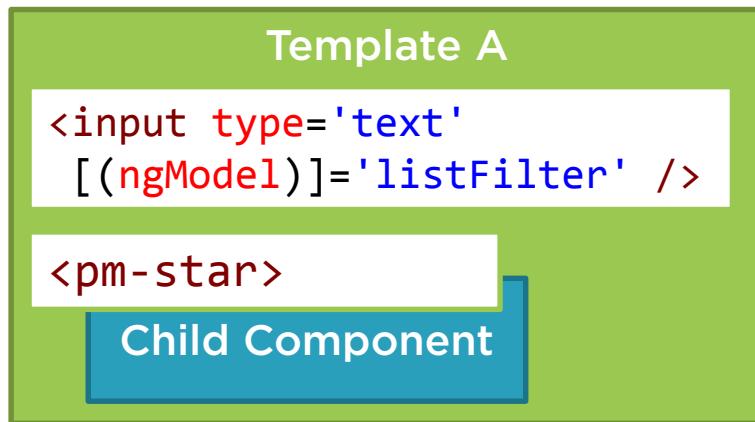
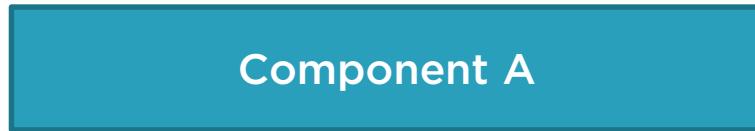
```
<div>
  <pm-criteria
    [displayDetail]='includeDetail'
    (valueChange)='onValueChange($event)'
  </pm-criteria>
</div>

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

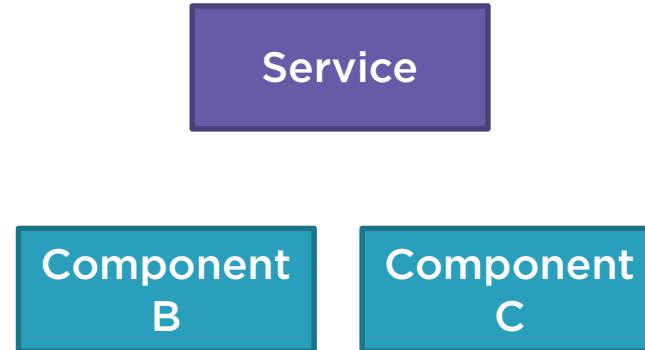
Available	Price	5 Star Rating	
March 19, 2016	\$19.95	★★★	Edit
March 18, 2016	\$32.99	★★★★	Edit
May 21, 2016	\$8.90	★★★★★	Edit
October 15, 2015	\$35.95	★★★★★	Edit



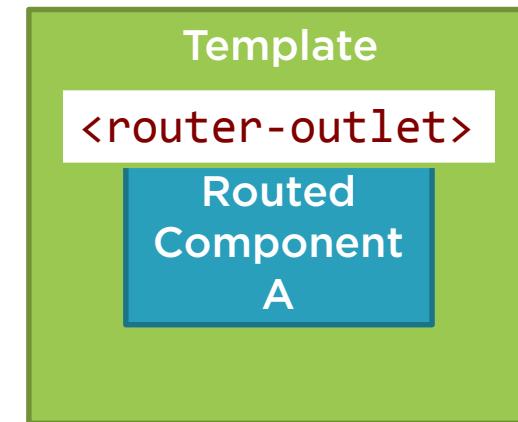
Component Communication



Template



Service



Routing



Module Overview



Child Components

Parent to Child Communication

Input Property

Watching for Changes

- Getter and Setter
- OnChanges Lifecycle Hook

Template Reference Variable

ViewChild Decorator



Building Child Components

Product List

Filter by:

Filtered by: r

Show Image

Product

Code

Leaf Rake

gdn-0011

Garden Cart

gdn-0023

Child Template

```
<div>Filter by:</div>
<div>
  <input type='text'
    [(ngModel)]="listFilter" />
</div>
<div>
  <h3>Filtered by: {{listFilter}} </h3>
</div>
```

Parent Template

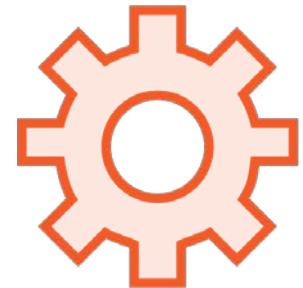
```
<div>
  <pm-criteria
    [displayDetail]='includeDetail'
    (valueChange)='onValueChange($event)'>
  </pm-criteria>
</div>
```

Child Component

```
@Component({
  selector: 'pm-criteria',
  templateUrl: './criteria.component.html'
})
export class CriteriaComponent {
```



Parent to Child

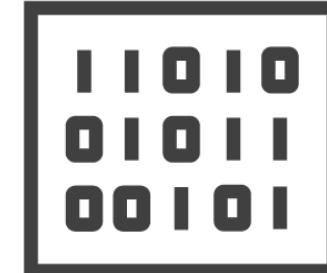


Configuration



Default value

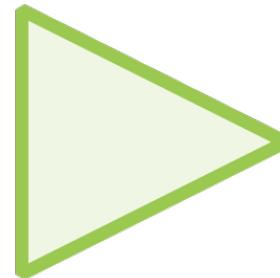
@Input
Getter/Setter
OnChanges



Item data



Request information



Perform an Action

Template Reference Variable
@ViewChild



Input: Passing Data to the Child

Parent Template

```
<pm-criteria  
  [displayDetail]='includeDetail'>  
</pm-criteria>
```

Child Component

```
@Input() displayDetail: boolean;
```

Parent Component

```
includeDetail: boolean = true;
```



Changes to an Input Property

Parent Template

```
<pm-criteria  
  [displayDetail]='includeDetail'>  
</pm-criteria>
```

Child Component

```
@Input() displayDetail: boolean;
```

Parent Component

```
includeDetail: boolean = true;
```

```
includeDetail: boolean = false;
```

Child Template

```
<div *ngIf='displayDetail'>  
  <h3>  
    Filtered by: {{listFilter}}  
  </h3>  
</div>
```



Watching for Changes to an Input Property

Child Component

```
private _hitCount: number;  
get hitCount(): number {  
  return this._hitCount;  
}  
@Input()  
set hitCount(value: number) {  
  this._hitCount = value;  
}
```

Getter and Setter

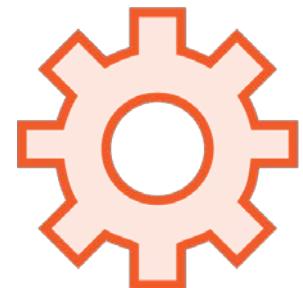
Child Component

```
@Input() hitCount: number;  
  
 ngOnChanges(changes: SimpleChanges) {  
}  
}
```

OnChanges Lifecycle Hook



Parent to Child

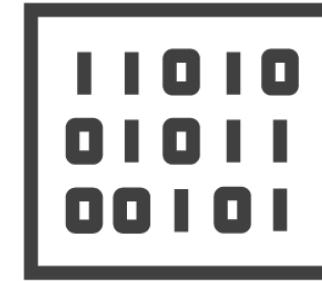


Configuration



Default value

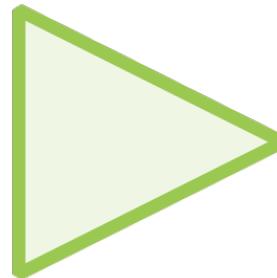
@Input
Getter/Setter
OnChanges



Item data



Request information



Perform an Action

Template Reference Variable
@ViewChild



Template Reference Variable: Referencing a Child Component

Parent Template

```
<pm-criteria  
  [displayDetail]='includeDetail'>  
</pm-criteria>  
  
&{{ filterCriteria.listFilter }}  
  
&{{ filterCriteria.clear() }}
```

Child Component

```
@Input() displayDetail: boolean;  
  
listFilter: string;  
  
clear(): void {  
  
}
```



ViewChild: Referencing a Child Component

Parent Template

```
<pm-criteria  
  [displayDetail]='includeDetail'>  
</pm-criteria>
```

Parent Component

```
export class ProductListComponent  
  implements OnInit, AfterViewInit  
  
@ViewChild(CriteriaComponent)  
filterComponent: CriteriaComponent;  
  
ngAfterViewInit(): void {  
  this.filterComponent.clear();  
}
```

Child Component

```
@Input() displayDetail: boolean;  
  
listFilter: string;  
  
clear(): void { ... }
```



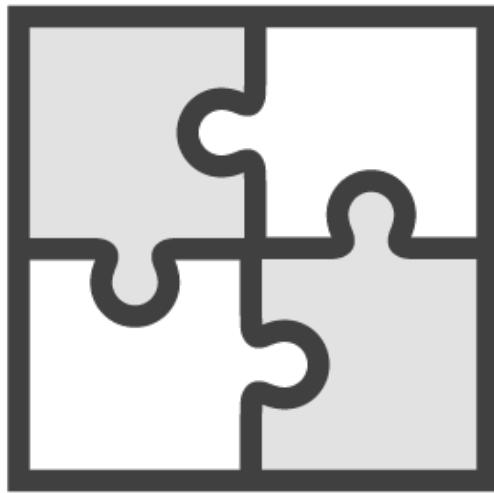
Guidelines & Summary: Communicating with a Child Component



Defining Child Components



Specific Task



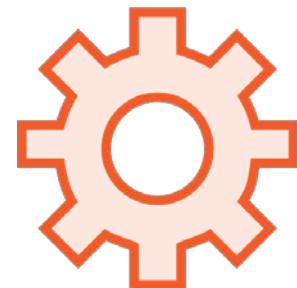
Complex



Reusable



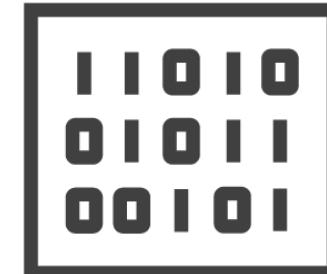
Parent to Child



Configuration



Default value



Item data

@Input
Getter/Setter
OnChanges

Getter/Setter:

- Favor to only react to changes to specific properties

OnChanges:

- Favor to react to any input property changes
- Favor if current and prior values are needed



Parent to Child

Template Reference Variable:

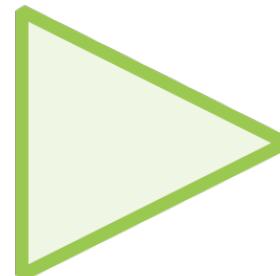
- Use from the parent's template



Request information

ViewChild:

- Use from the parent's class



Perform an Action

Template Reference Variable
@ViewChild



Summary



Child Components

Parent to Child Communication

Input Property

Watching for Changes

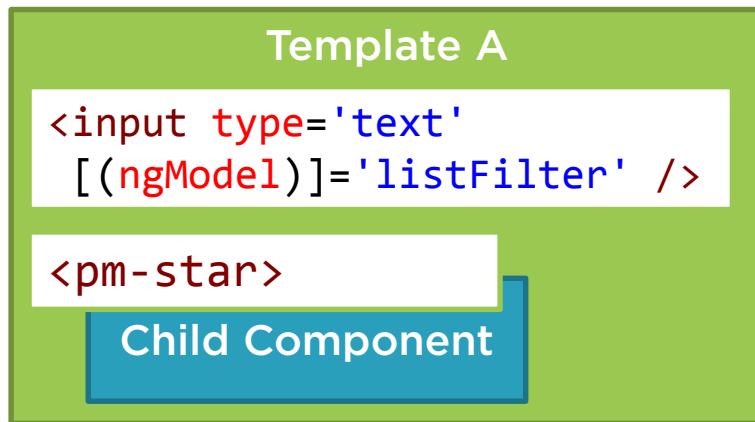
- Getter and Setter
- OnChanges Lifecycle Hook

Template Reference Variable

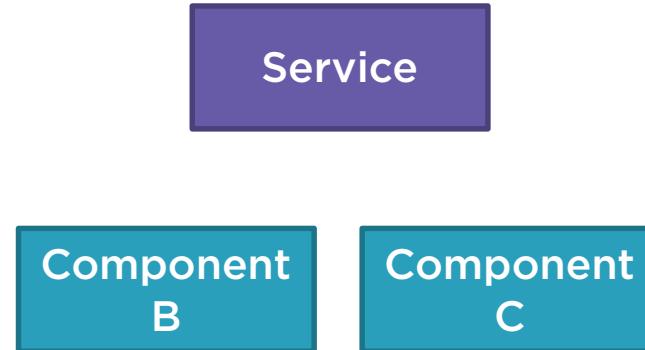
ViewChild Decorator



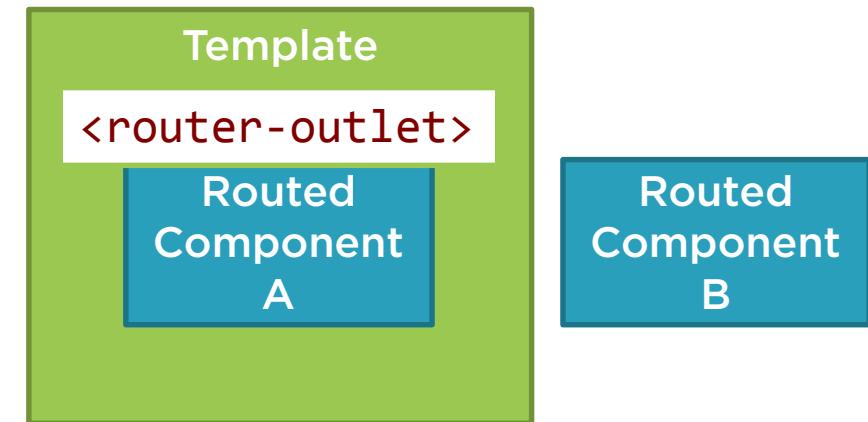
Component Communication



Template



Service



Routing



Communicating with a Parent Component



Deborah Kurata

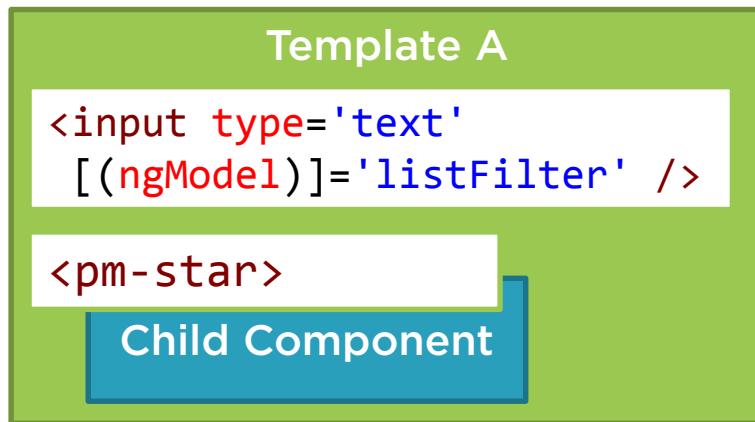
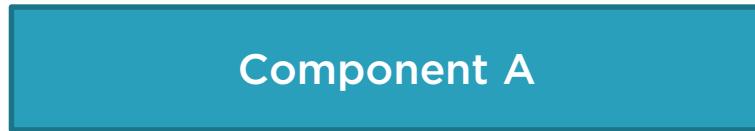
CONSULTANT | SPEAKER | AUTHOR | MVP | GDE

@deborahkurata | blogs.msmvps.com/deborahk/

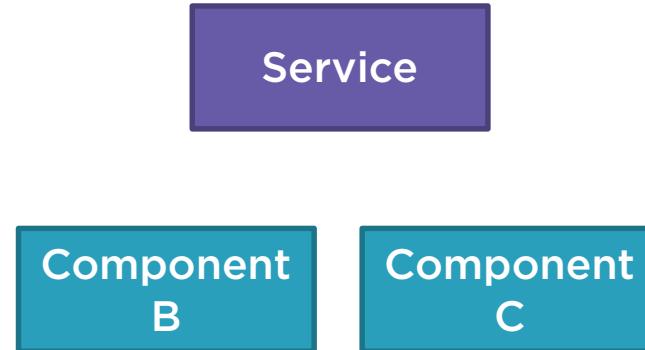




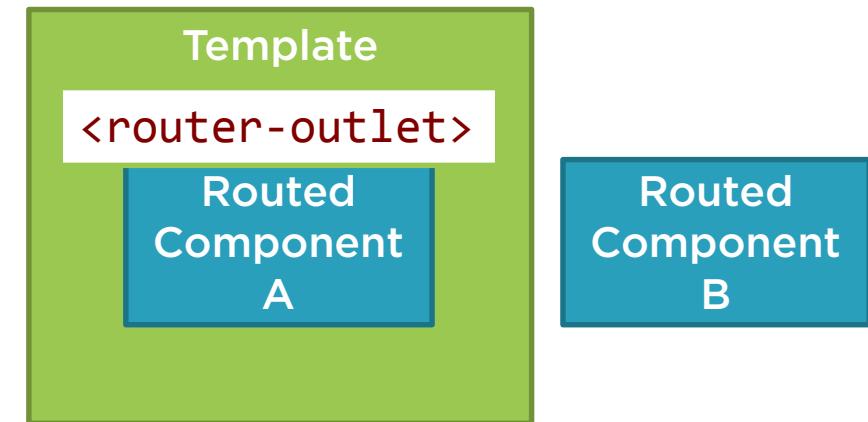
Component Communication



Template



Service



Routing



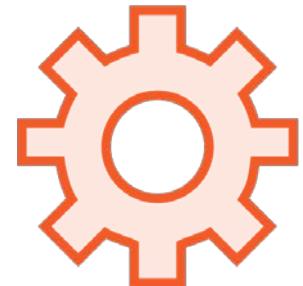
Module Overview



**Child to Parent Communication
Output Properties**



Parent to Child

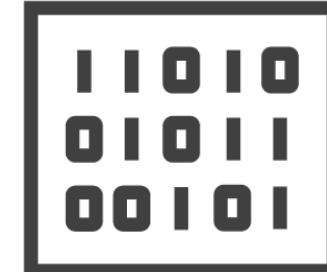


Configuration



Default value

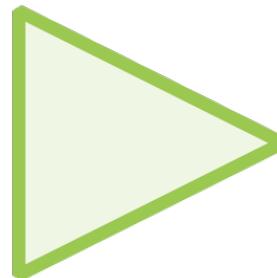
@Input
Getter/Setter
onChanges



Item data



Request information



Perform an Action

Template Reference Variable
@ViewChild





I'll refilter.
Thanks kiddo!

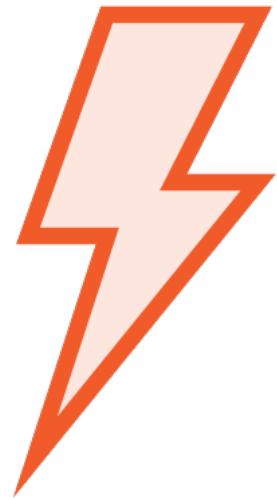
Hey Dad!
The user modified
a value.

Parent
Component

Child
Component

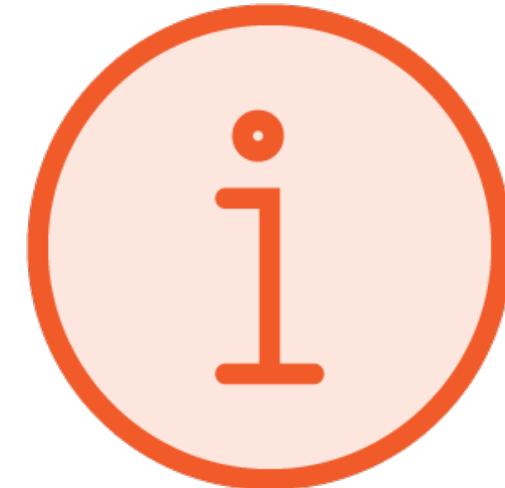


Child to Parent



Event notification

`@Output`



Provide information

Template Reference
Variable

`@ViewChild`

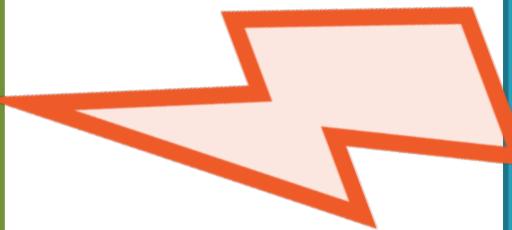


Output: Notifying the Parent

Parent Template

Child Component

Parent Component



Output: Notifying the Parent

Parent Template

```
<pm-criteria  
  [displayDetail]='includeDetail'  
  (valueChange)='onValueChange($event)'>  
</pm-criteria>
```

Child Component

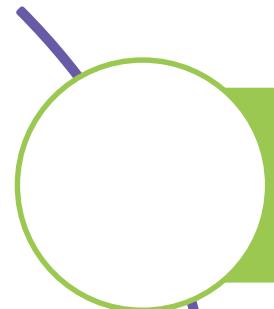
```
@Output() valueChange: EventEmitter<string>;  
  
this.valueChange.emit(value);
```

Parent Component

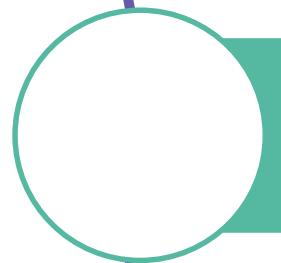
```
onValueChange(value: string): void {  
  this.performFilter(value);  
}
```



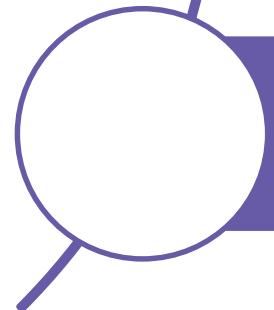
Notifying the Component of User Changes



Two-way binding, the long way



Getter and setter



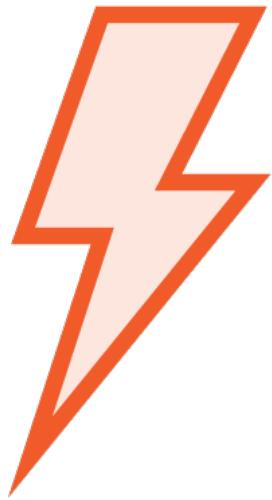
valueChanges observable



Guidelines & Summary: Communicating with a Parent Component



Child to Parent



Event notification

@Output

Output property:

- Notify parent of action
- Pass data to parent



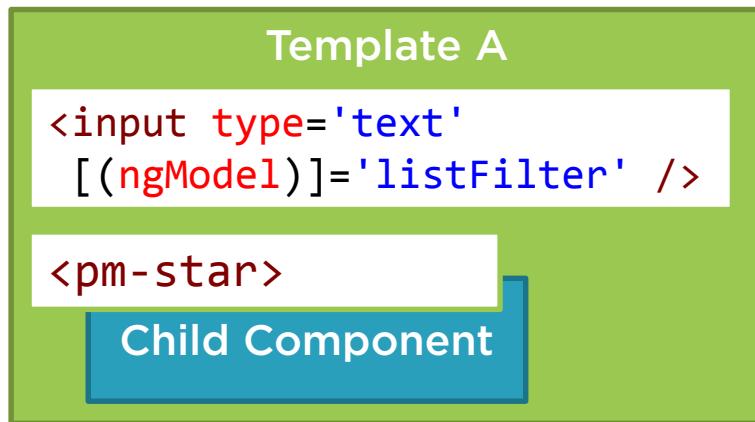
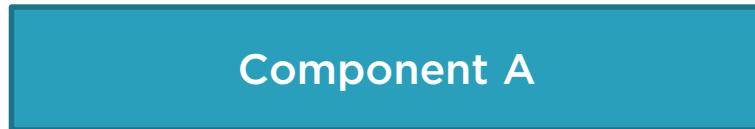
Summary



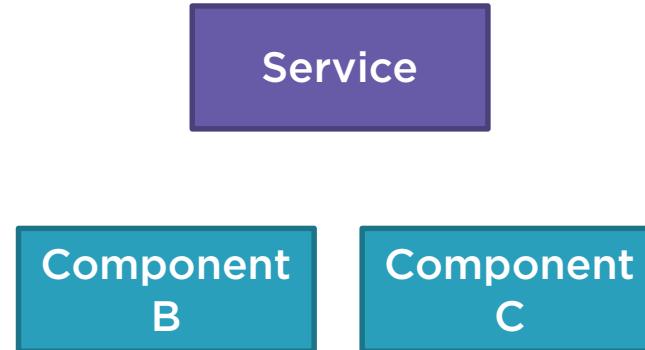
**Child to Parent Communication
Output Properties**



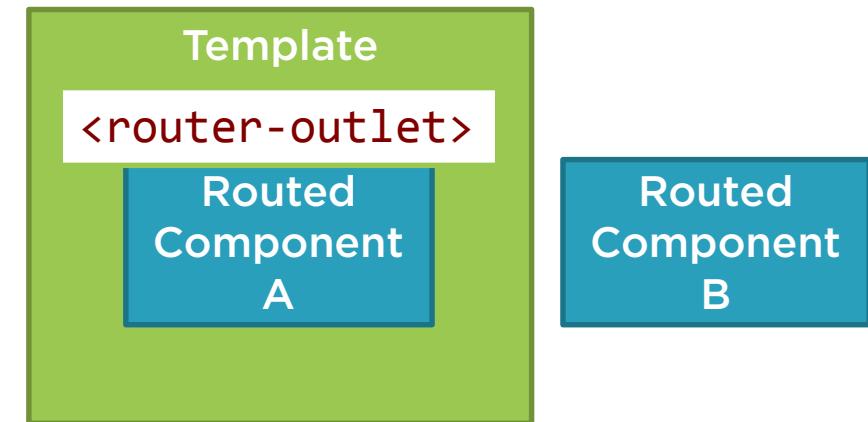
Component Communication



Template



Service



Routing



Communicating Through a Service



Deborah Kurata

CONSULTANT | SPEAKER | AUTHOR | MVP | GDE

@deborahkurata | blogs.msmvps.com/deborahk/





Yea!
You've got this
component!





Let's move the
party to my
place!



Communicating Through a Service

Component

```
showImage = true;
```



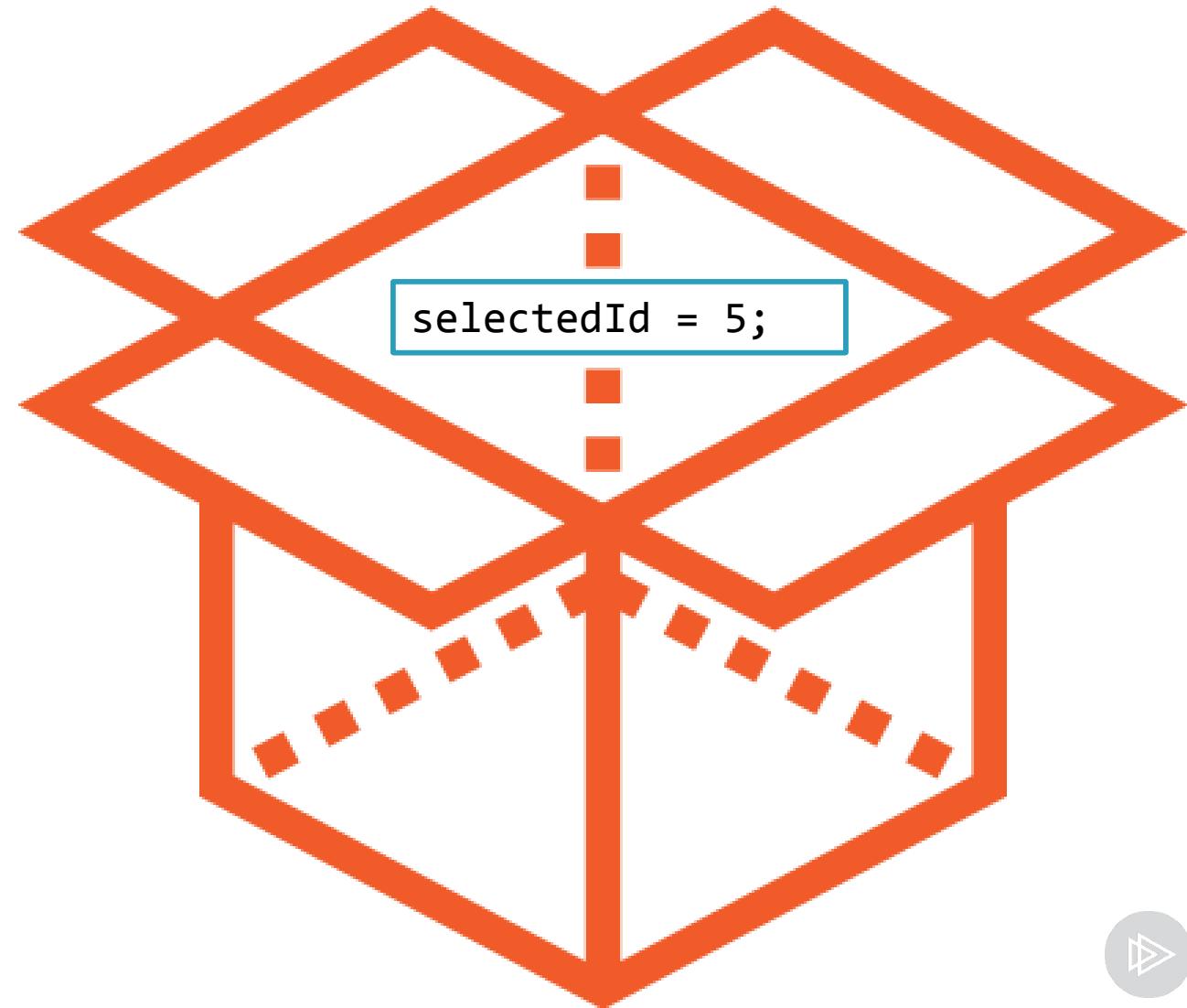
Communicating Through a Service

Component A

```
selectedId = 5;
```

Component B

Component C



Module Overview



Managing State

Property Bag Service

- Communication with Itself
- Communication with Others

Service Scope and Lifetime



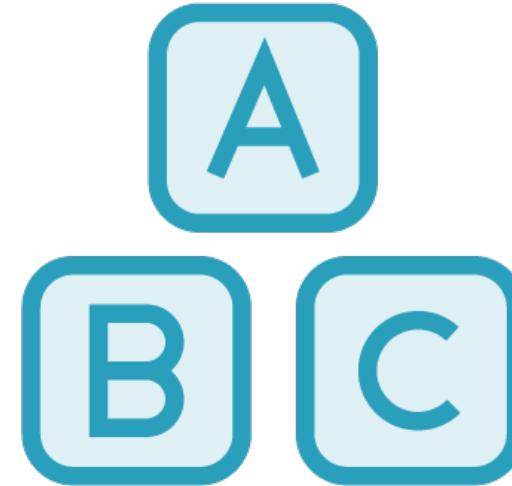
What Is State?



View State



User
Information



Entity Data



User Selection
and Input

...



Managing State

Property Bag

```
@Injectable()  
export class ParamService {  
  showImage: boolean;  
  filterBy: string;  
}  
}
```

Basic State Management

```
@Injectable()  
export class ProductService {  
  private products: IProduct[] = [  
    ...  
  ];  
  
  getProducts() {}  
  getProduct(id: number) {}  
  createProduct(product: IProduct) {}  
  ...  
}
```

State Management with Notifications

```
@Injectable()  
export class ProductService {  
  private selectedProduct: IProduct;  
  
  private selectedProductSource =  
    new BehaviorSubject<IProduct>();  
  selectedProductChanges$ =  
    this.selectedProductSource.asObservable();  
  ...  
}
```

ngrx/Redux

State:
Immutable

Will you keep
track of my
current filter?

Sure!
Retaining your
current filter.

What was my
filter?

I have it. Here
you go ...



Product List

Filter by:

Filtered by: none Hits:5

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



Product List

Filter by:

me

Filtered by: me Hits:2

Show Image	Product	Code	Available	Price	5 Star Rating	
	Hammer	tbx-0048	May 21, 2016	\$8.90	★★★★★	<button>Edit</button>
	Video Game Controller	gmg-0042	October 15, 2015	\$35.95	★★★★★	<button>Edit</button>



Product List

Filter by:

me

Filtered by: me Hits:2

Hide Image	Product	Code	Available	Price	5 Star Rating	
	Hammer	tbx-0048	May 21, 2016	\$8.90	 	
	Video Game Controller	gmg-0042	October 15, 2015	\$35.95	 	



Product Detail: Hammer

Name: Hammer
Code: tbx-0048
Description: Curved claw steel hammer
Availability: May 21, 2016
Price: \$8.90
5 Star Rating: ★★★★★

 [Back](#) [Edit](#)

Product List

Filter by:

Filtered by: none Hits:5

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



Product List					
Filter by: <input type="text" value="me"/>					
Filtered by: me Hits:2					
Hide Image	Product	Code	Available	Price	5 Star Rating
	Hammer	tbx-0048	May 21, 2016	\$8.90	★★★★★
	Video Game Controller	gmg-0042	October 15, 2015	\$35.95	★★★★★

Product List Component

```
export class ProductListComponent {  
  showImage: boolean;  
  listFilter: string;
```

```
  ...  
}
```

true

me



Product Detail: Hammer

Name: Hammer
Code: tbx-0048
Description: Curved claw steel hammer
Availability: May 21, 2016
Price: \$8.90
5 Star Rating: ★★★★☆



[« Back](#)

[Edit](#)



Product List					
Filter by: <input type="text"/>					
Filtered by: none Hits:5					
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

```
export class ProductListComponent {  
  showImage: boolean;  
  listFilter: string;
```

```
  ...  
}
```

undefined

undefined



Product List					
Filtered by: none Hits:5					
<input type="button" value="Show Image"/> Product Code Available Price 5 Star Rating					
Leaf Rake	gdn-0011	March 19, 2016	\$19.95	★★★	<input type="button" value="Edit"/>
Garden Cart	gdn-0023	March 18, 2016	\$32.99	★★★★	<input type="button" value="Edit"/>
Hammer	tbx-0048	May 21, 2016	\$8.90	★★★★★	<input type="button" value="Edit"/>
Saw	tbx-0022	May 15, 2016	\$11.55	★★★★	<input type="button" value="Edit"/>
Video Game Controller	gmg-0042	October 15, 2015	\$35.95	★★★★★	<input type="button" value="Edit"/>

Product List Component

```
export class ProductListComponent {  
  showImage: boolean;  
  listFilter: string;  
  ...  
}
```

Service

```
@Injectable()  
export class ParamService {  
  showImage: boolean;  
  filterBy: string;  
}
```

Acme Product Management Home Product List Add Product Welcome Deborah Log Out

Product List

Filter by: me

Filtered by: me Hits:2

true

	Code	Available	Price	5 Star Rating	Action
	tbx-0048	May 21, 2016	\$8.90	★★★★★	<input type="button" value="Edit"/>
	gmg-0042	October 15, 2015	\$35.95	★★★★★	<input type="button" value="Edit"/>

Product List Component

```
export class ProductListComponent {  
  get showImage(): boolean {  
    return this.paramService.showImage;  
  }  
  set showImage(value: boolean) {  
    this.paramService.showImage = value;  
  }  
  get listFilter(): string {  
    return this.paramService.filterBy;  
  }  
  set listFilter(value: string) {  
    this.paramService.filterBy = value;  
  }  
}
```

Service

```
@Injectable()  
export class ParamService {  
  showImage: boolean;  
  filterBy: string;  
}
```

Product Detail: Hammer

Name:	Hammer
Code:	tbx-0048
Description:	Curved claw steel hammer
Availability:	May 21, 2016
Price:	\$8.90
5 Star Rating:	★★★★★



[Back](#) [Edit](#)

Service

```
@Injectable()  
export class ParamService {  
  showImage: boolean; true  
  filterBy: string; me  
}
```

Acme Product Management Home Product List Add Product Welcome Deborah Log Out

Product List

Filter by:

Filtered by: me Hits:2

Hide Image	Product	Code	Available	Price	5 Star Rating	
	Hammer	tbx-0048	May 21, 2016	\$8.90	★★★★★	<button>Edit</button>
	Video Game Controller	gmg-0042	October 15, 2015	\$35.95	★★★★★	<button>Edit</button>

Product List Component

```
export class ProductListComponent {  
  get showImage(): boolean {  
    return this.paramService.showImage; true  
  }  
  set showImage(value: boolean) {  
    this.paramService.showImage = value;  
  }  
  get listFilter(): string {  
    return this.paramService.filterBy; me  
  }  
  set listFilter(value: string) {  
    this.paramService.filterBy = value;  
  }  
}
```

Service

```
@Injectable()  
export class ParamService {  
  showImage: boolean; true  
  filterBy: string; me  
}
```

Services

Provide functionality across components

Examples:

- Logging
- Calculations
- Data access
- Data sharing
- ...

Register it with the Angular injector

Inject into any component that needs it

Singlenton



Property Bag Service

Acme Product Management Home Product List Add Product Welcome Deborah Log Out

Product List

Filter by: me

Filtered by: me Hits:2

<input type="button" value="Hide Image"/>	Product	Code	Available	Price	5 Star Rating	<input type="button" value="Edit"/>
	Hammer	tbx-0048	May 21, 2016	\$8.90	★★★★★	<input type="button" value="Edit"/>
	Video Game Controller	gmg-0042	October 15, 2015	\$35.95	★★★★★	<input type="button" value="Edit"/>

Service

```
@Injectable()
export class ParamService {
  showImage: boolean;
  filterBy: string;
}
```



Property Bag Service

Product Search

Product Name: Controller

Product Code: gmg

Availability Date Range

Start: March 1, 2016

End: March 1, 2017

Search

Product List

Show Image	Product	Code	Available	Price	5 Star Rating	
	Video Game Controller	gmg-0042	October 15, 2015	\$35.95	★★★★★	Edit

Service

```
@Injectable()
export class ParamService {
  productName: string;
  productCode: string;
  startDate: string;
  endDate: string;
}
```



Property Bag Service

Service

```
@Injectable()  
export class ParamService {  
  showImage: boolean;  
  filterBy: string;  
}
```

Service

```
@Injectable()  
export class ParamService {  
  productName: string;  
  productCode: string;  
  startDate: string;  
  endDate: string;  
}
```



Registering a Service

product.component.ts

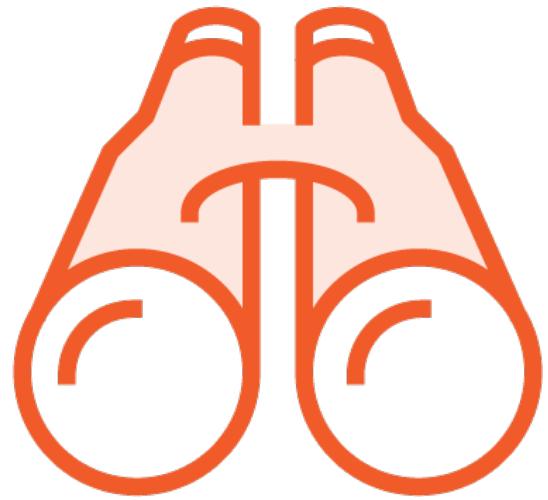
```
@Component({
  templateUrl: './product-list.component.html',
  styleUrls: ['./product-list.component.css'],
  providers: [ ProductService ]
})
export class ProductListComponent {
  pageTitle: string = 'Product List';
  includeDetail: boolean = true;
  ...
}
```

product.module.ts

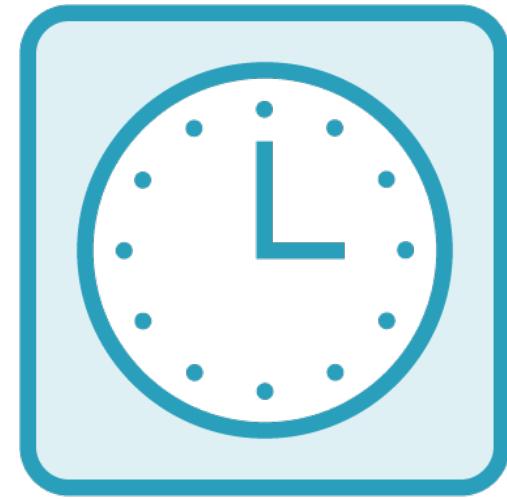
```
@NgModule({
  ...,
  declarations: [
    ProductListComponent,
    ProductDetailComponent,
    ProductEditComponent
  ],
  providers: [
    ProductService,
    ProductEditGuard,
    ProductParameterService
  ]
})
export class ProductModule { }
```



Service Scope and Lifetime



Service Scope



Service Lifetime



Registering in a Component

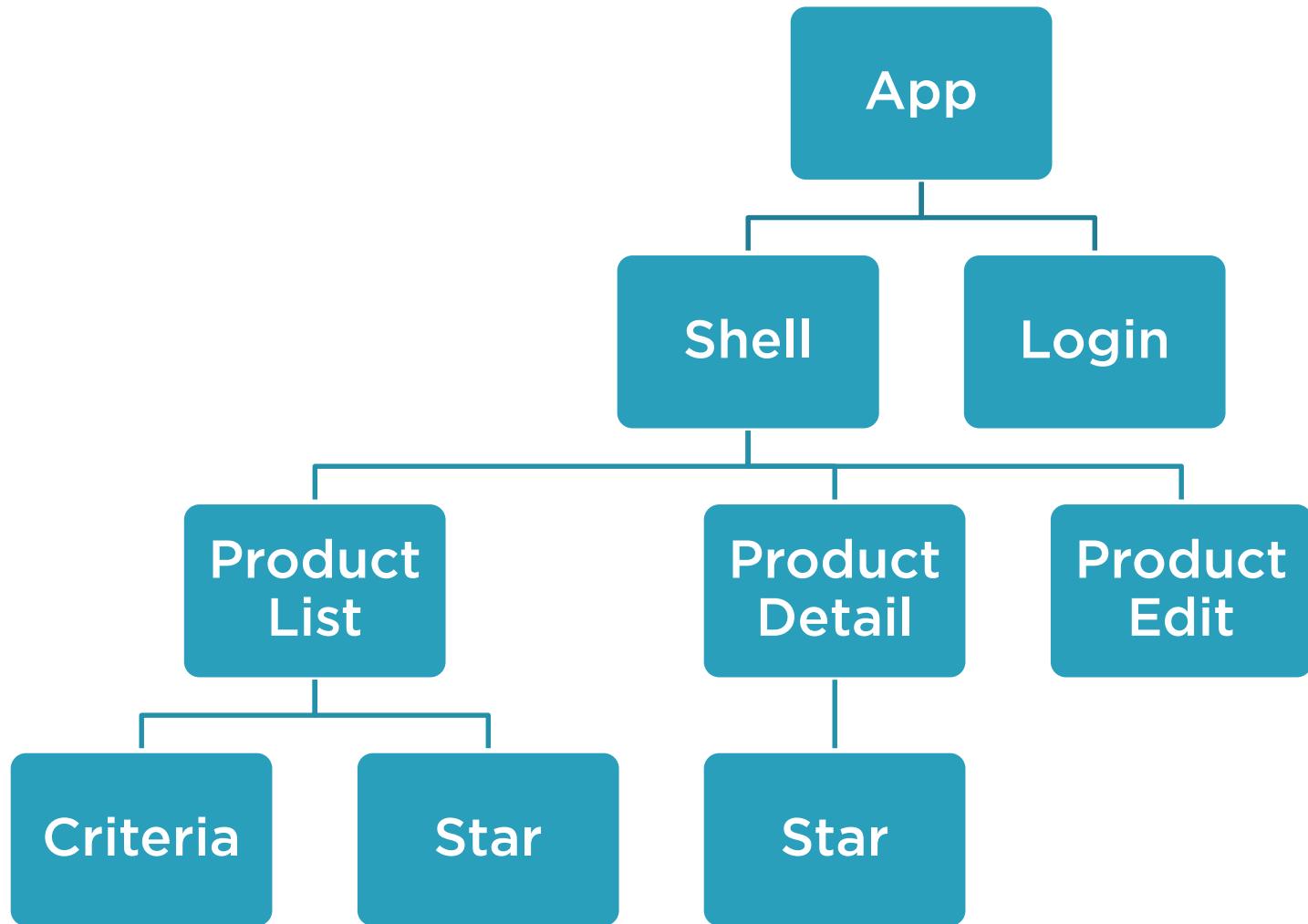
product.component.ts

```
@Component({
  templateUrl: './product-list.component.html',
  styleUrls: ['./product-list.component.css'],
  providers: [ ProductService ]
})
export class ProductListComponent {
  pageTitle: string = 'Product List';
  includeDetail: boolean = true;

  ...
}
```

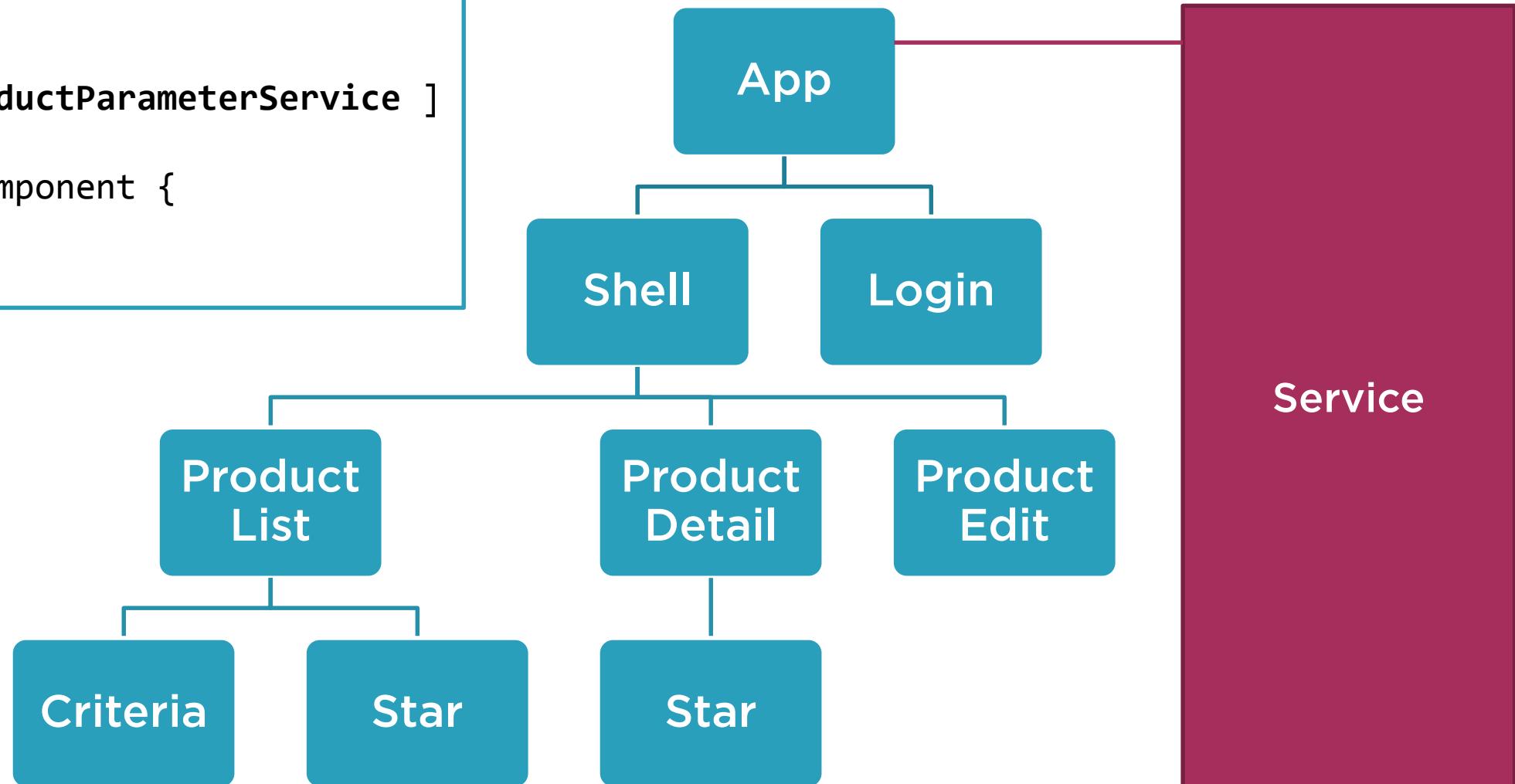


Component Hierarchy



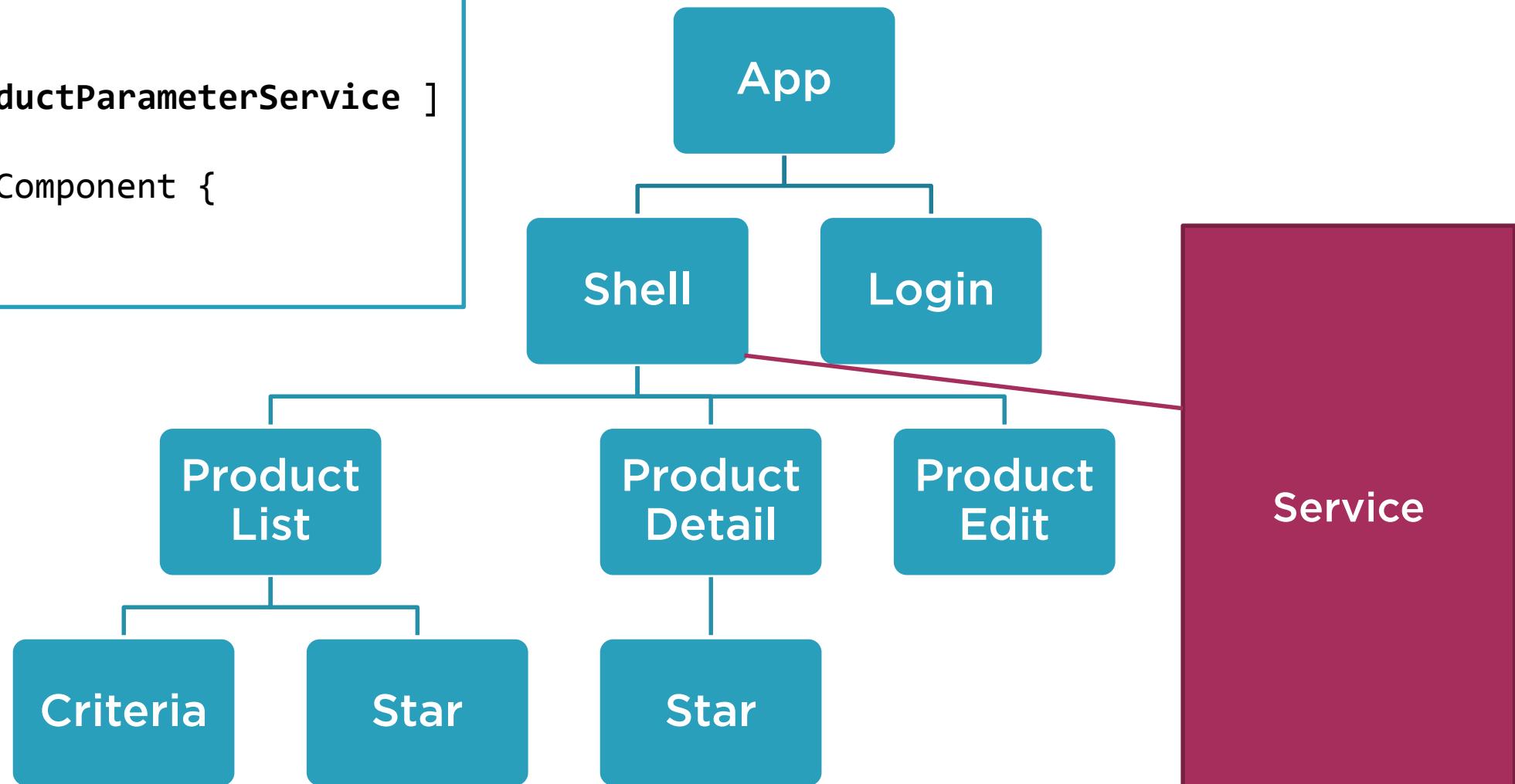
Registering in a Component

```
@Component({  
  ...  
  providers: [ ProductParameterService ]  
})  
export class AppComponent {  
  ...  
}
```



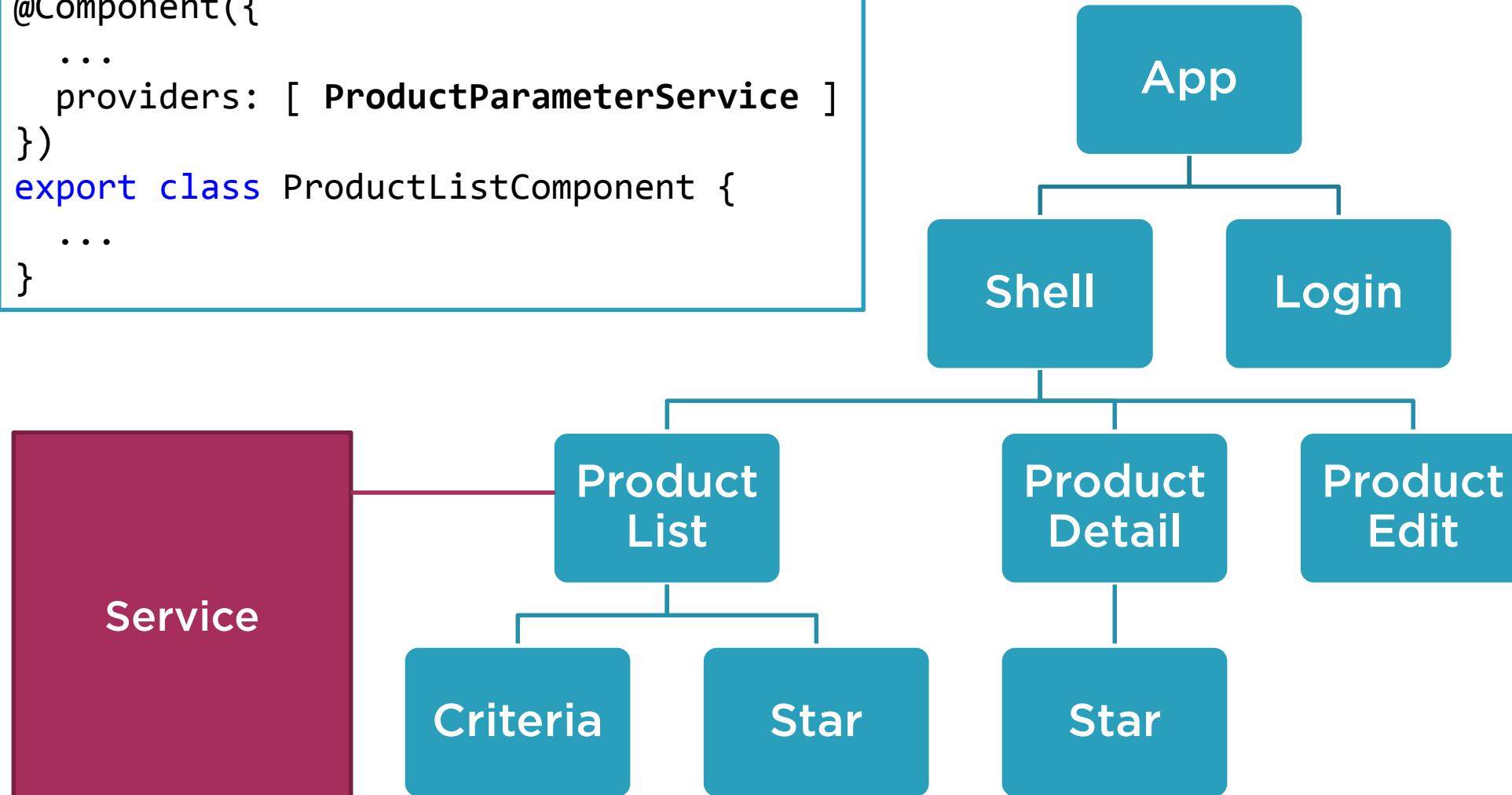
Registering in a Component

```
@Component({  
  ...  
  providers: [ ProductParameterService ]  
})  
export class ShellComponent {  
  ...  
}
```



Registering in a Component

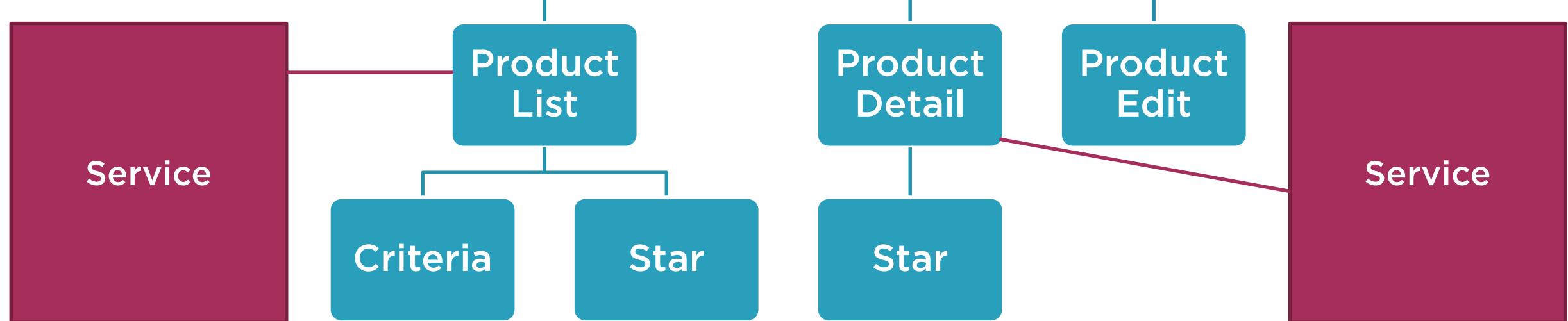
```
@Component({  
  ...  
  providers: [ ProductParameterService ]  
})  
export class ProductListComponent {  
  ...  
}
```



Registering in a Component

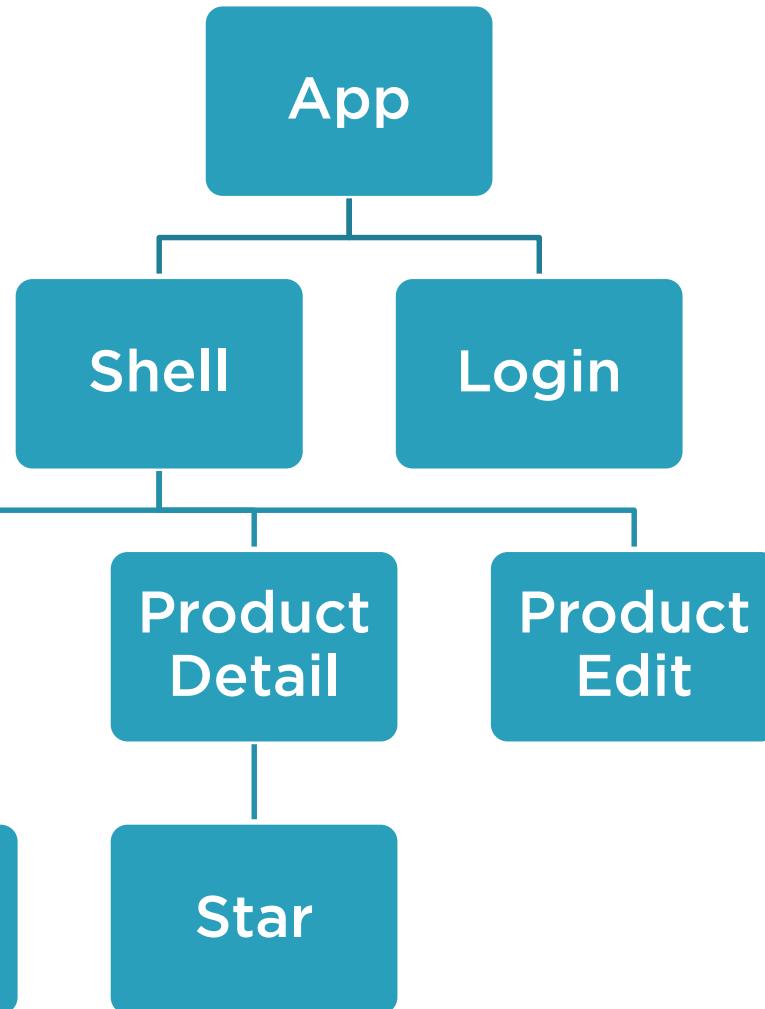
```
@Component({  
  ...  
  providers: [ ProductParameterService ]  
})  
export class ProductListComponent {  
  ...  
}
```

```
@Component({  
  ...  
  providers: [ ProductParameterService ]  
})  
export class ProductDetailComponent {  
  ...  
}
```



Multiple Instances

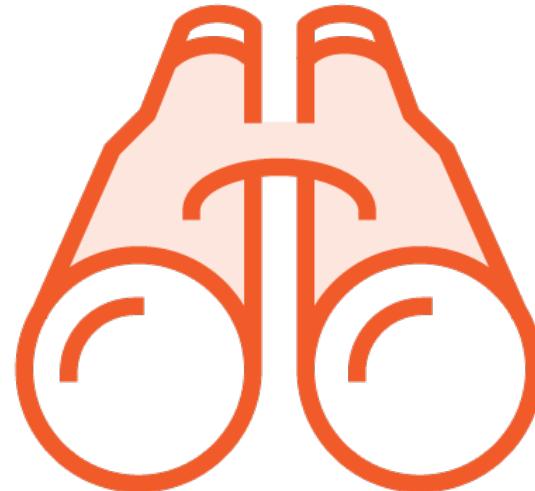
```
<div class='row'>
<div class='col-md-4'>
  <pm-product-list></pm-product-list>
</div>
<div class='col-md-8'>
  <pm-product-list></pm-product-list>
</div>
</div>
```



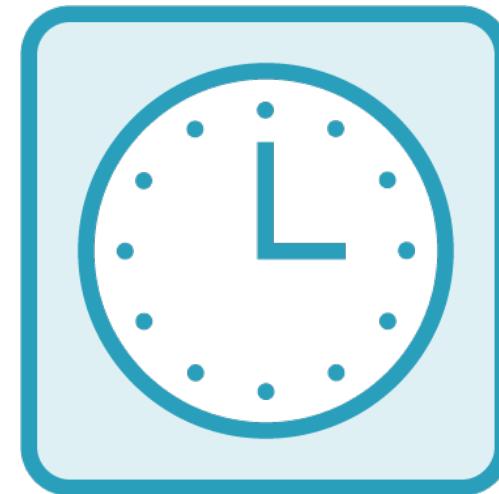
Registering in a Component

product.component.ts

```
@Component({
  templateUrl: './product-list.component.html',
  styleUrls: ['./product-list.component.css'],
  providers: [ ProductService ]
})
export class ProductListComponent {
  pageTitle: string = 'Product List';
  includeDetail: boolean = true;
  ...
}
```



Service
Scope



Service
Lifetime



Monitoring Service Lifetime

```
@Injectable()
export class ProductParameterService implements OnDestroy {
  showImage: boolean;
  filterBy: string;

  constructor() {
    console.log('Service created');
  }

  ngOnDestroy(): void {
    console.log('Service destroyed');
  }
}
```



Registering in a Module

product.component.ts

```
@Component({
  templateUrl: './product-list.component.html',
  styleUrls: ['./product-list.component.css'],
  providers: [ ProductService ]
})
export class ProductListComponent {
  pageTitle: string = 'Product List';
  includeDetail: boolean = true;
  ...
}
```

product.module.ts

```
@NgModule({
  ...,
  declarations: [
    ProductListComponent,
    ProductDetailComponent,
    ProductEditComponent
  ],
  providers: [
    ProductService,
    ProductEditGuard,
    ProductParameterService
  ]
})
export class ProductModule { }
```



Guidelines & Summary: Communicating through a Service



Property Bag: Communication with Itself

Great for:

- Retaining view state
- Retaining user selections

Note:

- Any component can read the values
- Any component can change the values



Property Bag: Communication with Others

Great for:

- Sharing data or other state
- Communicating state changes

Note:

- Any component can read the values
- Any component can change the values
- Components are only notified of state changes if they use template binding



Service Scope and Lifetime

Register service in a module or root App component:

- Available to all components
- Available for the lifetime of the application

Register service in a lazy loaded module:

- Available only to components within the module
- Available for the lifetime of the application

Register service in any other component:

- Injects a separate instance for each instance of the component

Register service in any other component:

- Available only to that component and all of its children
- Available for the lifetime of the component



Summary



Managing State

Property Bag Service

- Communication with Itself
- Communication with Others

Service Scope and Lifetime



Managing State



```
@Injectable()  
export class ParamService {  
  showImage: boolean;  
  filterBy: string;  
}
```



Communicating Through a State Management Service



Deborah Kurata

CONSULTANT | SPEAKER | AUTHOR | MVP | GDE

@deborahkurata | blogs.msmvps.com/deborahk/





Managing State



```
@Injectable()  
export class ParamService {  
  showImage: boolean;  
  filterBy: string;  
}
```



Module Overview



State Management Service

- Sharing Entity State

Displaying Concurrent Components

Keeping State in Sync

Change Detection

- Communicating State Changes



Retrieving Entity Data

Acme Product Management Home Product List Add Product Welcome Deborah Log Out

Product Detail: Hammer

Name:	Hammer
Code:	tbx-0048
Description:	Curved claw steel hammer
Availability:	May 21, 2016
Price:	\$8.90
5 Star Rating:	★★★★★



[Back](#) [Edit](#)

Data Access Service

```
@Injectable()
export class ProductService {
  getProducts() {}
  getProduct(id: number) {}

  createProduct(product: IProduct) {}
  deleteProduct(id: number} {}
  updateProduct(product: IProduct) {}

}
```

Backend Server





Sharing Entity State

Acme Product Management Home Product List Add Product Welcome Deborah Log Out

Product Detail: Hammer

Name:	Hammer
Code:	tbx-0048
Description:	Curved claw steel hammer
Availability:	May 21, 2016
Price:	\$8.90
5 Star Rating:	★★★★★



[Back](#) [Edit](#)

State Management Service

```
@Injectable()
export class ProductService {
    private products: IProduct[];

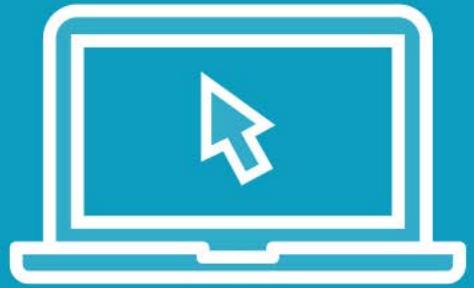
    getProducts() {}
    getProduct(id: number) {}

    createProduct(product: IProduct) {}
    deleteProduct(id: number) {}
    updateProduct(product: IProduct)
}
```

Backend Server



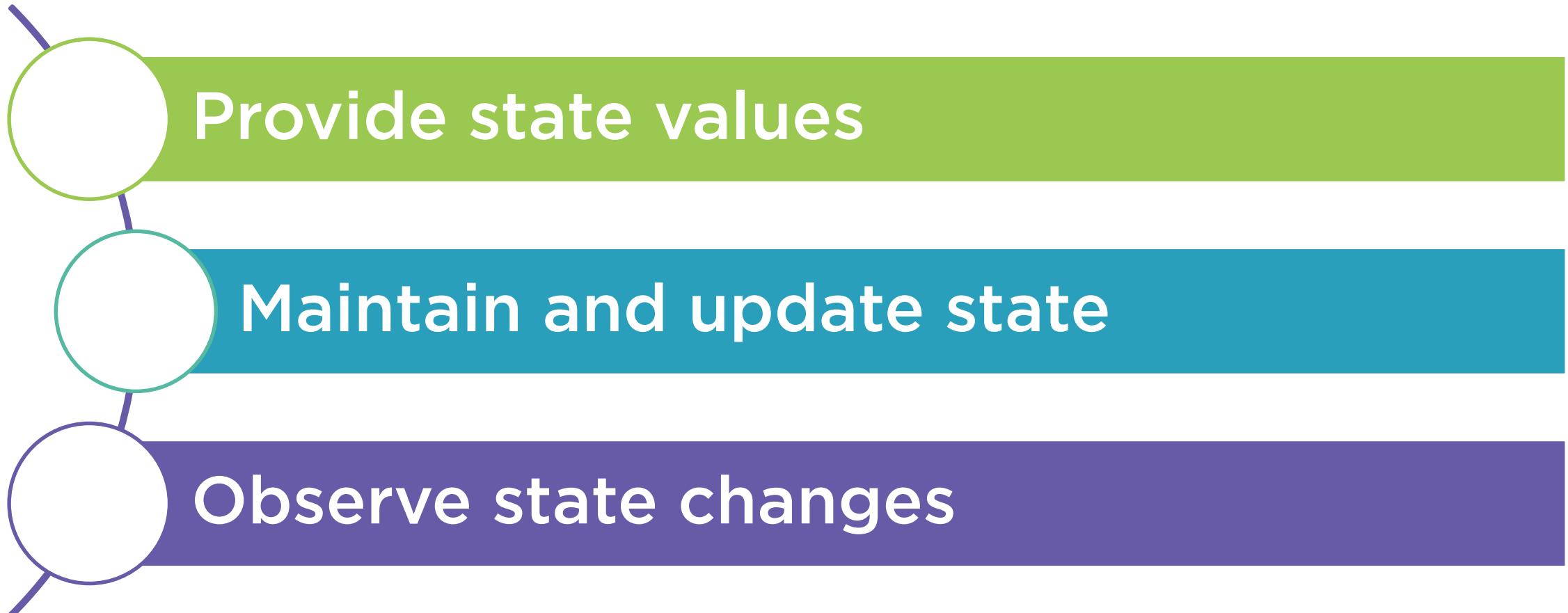
Demo



Current Data Access Service



State Management Service



State Management Service

Add a
property
to retain
the list

On get:
return the
list

On
get by Id:
return an
item from
the list

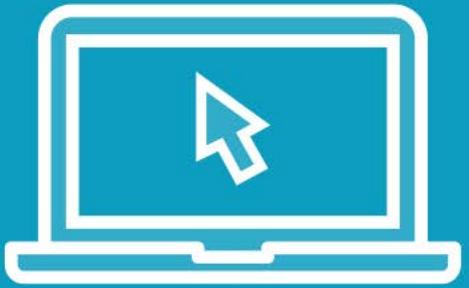
On **create:**
add the
item to the
list

On **delete:**
remove
the item
from the
list

Optionally
add code
to expire
the list



Demo



State Management Service: Retrieving State



Demo



State Management Service:
Maintaining State



Product List

Filter by:

Filtered by: none Hits:5

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



Displaying Concurrent Components

Acme Product Management Home Product List Add Product Welcome Deborah Log Out

Products

- Leaf Rake (GDN-0011)
- Garden Cart (GDN-0023)
- Hammer (TBX-0048)
- Saw (TBX-0022)
- Video Game Controller (GMG-0042)

Product Detail: Hammer

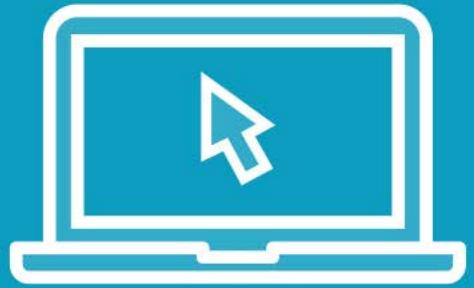
Name: Hammer
Code: TBX-0048
Description: Curved claw steel hammer
Availability: May 21, 2017
Price: \$8.90
5 Star Rating: ★★★★★



Edit



Demo



Keeping State in Sync



Binding

Product Detail Template

```
{{pageTitle}}
```

Product Detail

Name:	Hammer
Code:	tbx-0048
Description:	Curved claw steel hammer
Availability:	May 21, 2017
Price:	\$8.90
5 Star Rating:	★★★★★



Edit

Product Detail Component

```
export class ProductDetailComponent {  
  pageTitle: string = 'Product Detail';  
}
```



Binding

Product Detail Template

```
Name: {{product.productName}}  
Code: {{product.productCode}}  
Description: {{product.description}}
```

Product Detail

Name:	Hammer
Code:	tbx-0048
Description:	Curved claw steel hammer
Availability:	May 21, 2017
Price:	\$8.90
5 Star Rating:	★★★★★



Edit

Product Detail Component

```
export class ProductDetailComponent {  
  product: IProduct;  
}
```



Change Detection

Product Detail Template

```
Name: {{product.productName}}  
Code: {{product.productCode}}  
Description: {{product.description}}
```

Product Detail

Name:	Steel Hammer
Code:	tbx-0048
Description:	Curved claw steel hammer
Availability:	May 21, 2017
Price:	\$8.90
5 Star Rating:	★★★★★



Edit

Product Detail Component

```
export class ProductDetailComponent {  
  product: IProduct;  
}
```

```
this.product.productName = 'Steel Hammer';
```



Change Detection

Product Shell Detail Template

```
Name: {{product.productName}}  
Code: {{product.productCode}}  
Description: {{product.description}}
```

Product Detail: Hammer

Name:	Hammer
Code:	tbx-0048
Description:	Curved claw steel hammer
Availability:	May 21, 2017
Price:	\$8.90
5 Star Rating:	★★★★★



Edit

Product Shell Detail Component

```
get product(): IProduct {  
    return this.productService.currentProduct;  
}
```

Product Service

```
export class ProductService {  
    currentProduct: IProduct;  
}
```

Product Shell List Component

```
onSelected(product: IProduct) {  
    this.productService.currentProduct =  
        product;  
}
```

Change Detection

Product Shell Detail Template

```
Name: {{product.productName}}  
Code: {{product.productCode}}  
Description: {{product.description}}
```

Product Detail: Video Game Controller

Name:	Video Game Controller
Code:	gmg-0042
Description:	Standard two-button video game controller
Availability:	October 15, 2017
Price:	\$35.95
Rating:	★★★★★



Edit

Product Shell Detail Component

```
get product(): IProduct {  
    return this.productService.currentProduct;  
}
```

Product Service

```
export class ProductService {  
    currentProduct: IProduct;  
}
```

Product Shell List Component

```
onSelected(product: IProduct) {  
    this.productService.currentProduct =  
        product;  
}
```

Guidelines & Summary: Communicating through a State Management Service



Property Bag vs. State Management Service

Property Bag Service

```
@Injectable()  
export class ParamService {  
  showImage: boolean;  
  filterBy: string;  
}
```

- Retain view settings
- Pass data between components

State Management Service

```
@Injectable()  
export class ProductService {  
  private products: IProduct[];  
  
  getProducts() {}  
  getProduct(id: number) {}  
  
  createProduct(product: IProduct) {}  
  deleteProduct(id: number) {}  
  updateProduct(product: IProduct) {}  
}
```

- Retrieve, manage, store state
- Share state between components



State Management Service

Benefits:

- Encapsulates retrieve and store operations
- Retains and shares state values
- Minimizes hits to the backend server
 - Improves performance
- Provides change notifications for bound values using a getter

Considerations:

- Stale data
- No explicit change notifications
- State is not immutable

```
getProducts(): Observable<IProduct[]>
{
  if (this.products) {
    return of(this.products);
  }
  ...
}
```



Getter

Benefits:

- Keeps bound data in sync
 - Define a property in a service
 - Bind that property in a template
 - Use a getter in the component class
- Simple
- Easy to understand

Considerations:

- Only works with bound data

```
get product(): IProduct | null {  
    return this.productService.currentProduct;  
}
```



Timer

Benefits:

- Poll for changes

Considerations:

- Too short -> more processing than needed
- Too long -> delayed reaction
- Possible race conditions

```
ngOnInit() {  
  timer(0, 1000).subscribe(t =>  
    console.log(this.prod));  
}
```



Summary



State Management Service

- Sharing Entity State

Displaying Concurrent Components

Keeping State in Sync

Change Detection

- Communicating State Changes



Managing State

Property
Bag

Basic State
Management

State
Management
with
Notifications

ngrx/Redux

```
@Injectable()
export class ProductService {
  private products: IProduct[];

  getProducts() {}
  getProduct(id: number) {}
  createProduct(product: IProduct) {}
  ...
}
```



Communicating Through Service Notifications



Deborah Kurata

CONSULTANT | SPEAKER | AUTHOR | MVP | GDE

@deborahkurata | blogs.msmvps.com/deborahk/





The user just logged out!



Managing State

Property
Bag

Basic State
Management

State
Management
with
Notifications

ngrx/Redux

```
@Injectable()
export class ProductService {
  private products: IProduct[];

  getProducts() {}
  getProduct(id: number) {}
  createProduct(product: IProduct) {}
  ...
}
```



Module Overview



Broadcasting Service Notifications

- EventEmitter
- Subject
- BehaviorSubject



Displaying Concurrent Components

Acme Product Management Home Product List Add Product Welcome Deborah Log Out

Products

- Leaf Rake (GDN-0011)
- Garden Cart (GDN-0023)
- Hammer (TBX-0048)
- Saw (TBX-0022)
- Video Game Controller (GMG-0042)

Product Detail: Hammer

Name: Hammer
Code: TBX-0048
Description: Curved claw steel hammer
Availability: May 21, 2017
Price: \$8.90
5 Star Rating: ★★★★★



Edit



Sharing Data (Pull)

Detail Component

```
export class ProductDetailComponent {  
  get product(): IProduct {  
    return this.ps.currentProduct;  
  }  
  ...  
}
```

Service

```
@Injectable()  
export class ProductService {  
  currentProduct: IProduct;  
  Hammer
```

List Component

```
onSelected(product: IProduct) {  
  this.ps.currentProduct = product;  
}  
Hammer
```





Notifications (Push)

Products

Leaf Rake (GDN-0011)
Garden Cart (GDN-0023)
Hammer (TBX-0048)
Saw (TBX-0022)
Video Game Controller (GMG-0042)

Product Detail: Saw

Name:	Saw
Code:	tbx-0022
Description:	15-inch steel blade hand saw
Availability:	May 15, 2017
Price:	\$11.55
Rating:	★★★



Edit

List Template

```
<button type='button'  
*ngFor='let product of products'  
(click)='onSelected(product)'>
```

List Component

```
onSelected(product: IProduct) {  
  this.productService.changeSelectedProduct(product);  
}
```

Service

```
changeSelectedProduct(selectedProduct: IProduct) {  
  // Broadcast the notification  
}
```

Component or Service

```
// Listen for and respond to the notification
```

Event Emitter

Parent Template

```
<pm-criteria  
[displayDetail]='includeDetail'  
(valueChange)='onValueChange($event)'>  
</pm-criteria>
```

Child Component

```
export class CriteriaComponent {  
  
  @Output() valueChange: EventEmitter<string> =  
    new EventEmitter<string>();  
  
  private _listFilter: string;  
  get listFilter(): string {  
    return this._listFilter;  
  }  
  set listFilter(value: string) {  
    this._listFilter = value;  
    this.valueChange.emit(value);  
  }  
}
```



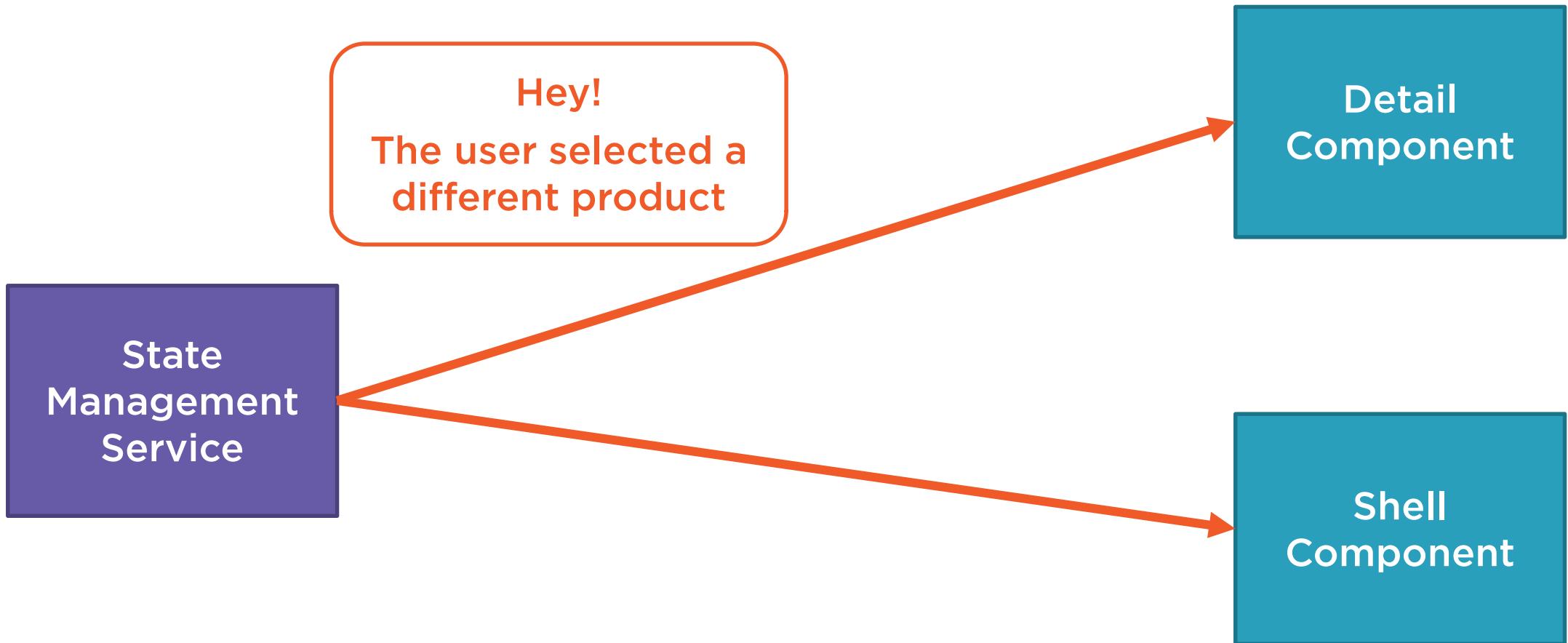
Service Notifications

Subject

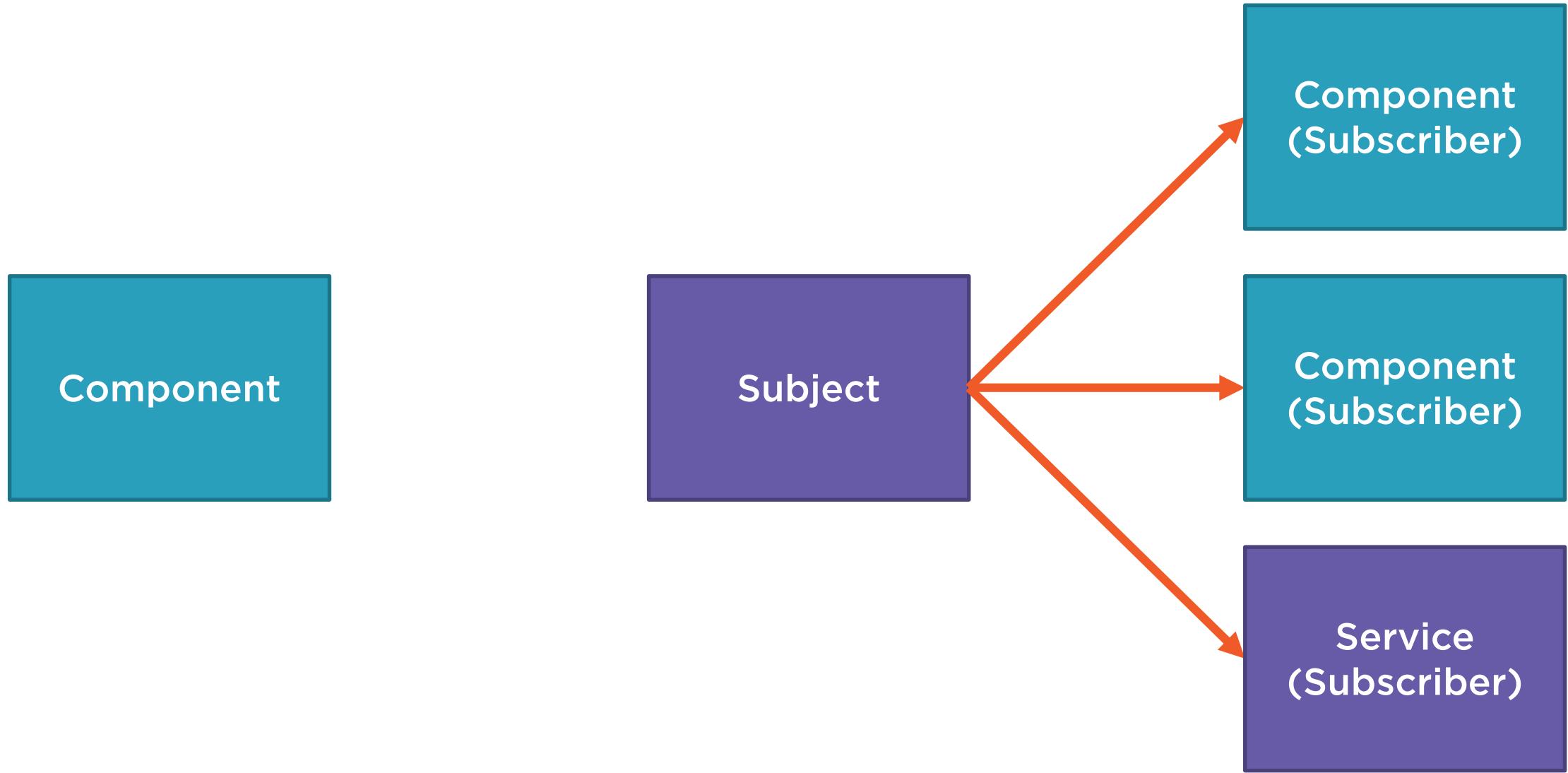
BehaviorSubject



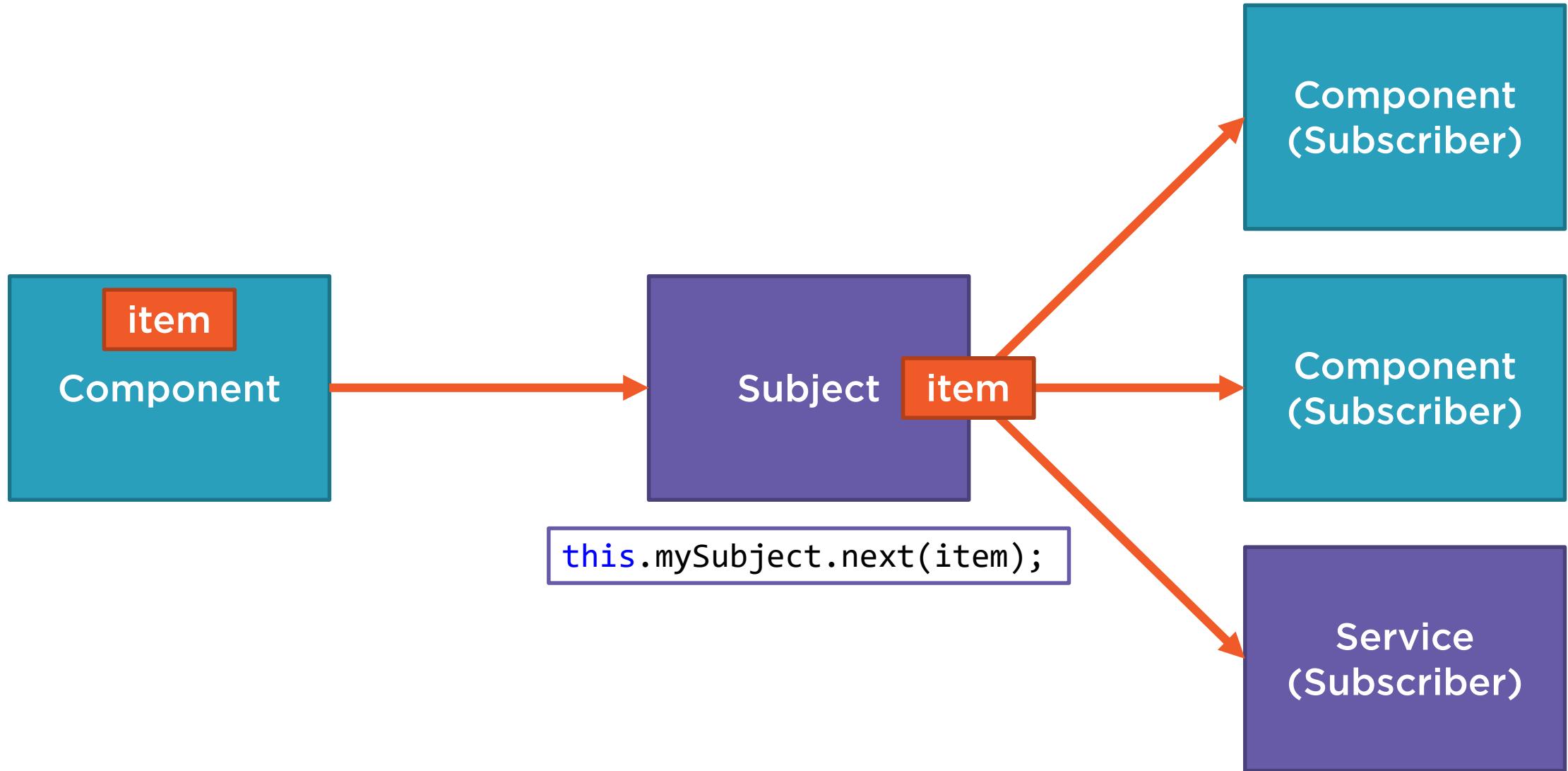
Subject



Subject: Observable



Subject: Observer



Products

Garden Cart (GDN-0023)

Hammer (TBX-0048)

Leaf Rake (GDN-0011)

Saw (TBX-0022)

Video Game Controller (GMG-0042)

Product Detail: Garden Cart

Name: Garden Cart

Code: gdn-0023

Description: 15 gallon capacity
rolling garden cart

Availability: March 18, 2017

Price: \$32.99

5 Star 

Rating:



Edit



Products

```
Leaf Rake (GDN-0011)  
Garden Cart (GDN-0023)  
Hammer (TBX-0048)  
Saw (TBX-0022)  
Video Game Controller (GMG-0042)
```

List Template

```
<button type='button'  
*ngFor='let product of products'  
(click)='onSelected(product)'>
```

Product Detail: Saw

Name: Saw
Code: tbx-0022
Description: 15-inch steel blade hand saw
Availability: May 15, 2017
Price: \$11.55
5 Star Rating: ★★★★



Edit

List Component

```
onSelected(product: IProduct) {  
    this.productService.changeSelectedProduct(product);  
}
```

Service

```
private selectedProductSource=new Subject<IProduct>();  
selectedProductChanges$ =  
    this.selectedProductSource.asObservable();  
  
changeSelectedProduct(selectedProduct: IProduct) {  
    this.selectedProductSource.next(selectedProduct);  
}
```

Detail Component

```
this.productService.selectedProductChanges$.subscribe(  
    selectedProduct => this.product = selectedProduct  
)
```



Demo



Using a Subject for Explicit Change Notifications

Product
Shell List

Product
Service

```
this.mySubject.next(item);
```

Product
Shell

Product
Shell
Detail



Demo



Implementing a Second Subscription

Product
Shell List

Product
Service

```
this.mySubject.next(item);
```

Product
Shell

Product
Shell
Detail



Acme Product Management Home Product List Add Product Welcome Deborah Log Out

Products

- Garden Cart (GDN-0023)
- Hammer (TBX-0048)
- Leaf Rake (GDN-0011)
- Saw (TBX-0022)
- Video Game Controller (GMG-0042)

Acme Product Management Home Product List Add Product Welcome Deborah Log Out

Edit Product: Garden Cart

Product Name Garden Cart

Product Code GDN-0023

Description 15 gallon capacity rolling garden cart

Save **Cancel** **Delete**

Service

```
private selectedProductSource = new Subject<IProduct>();  
selectedProductChanges$ =  
    this.selectedProductSource.asObservable();
```

Garden
Cart

```
changeSelectedProduct(selectedProduct: IProduct) {  
    this.selectedProductSource.next(selectedProduct);  
}
```



Subject vs. BehaviorSubject

Subject

BehaviorSubject

- Requires an initial value
- Provides the current value on a new subscription



Acme Product Management Home Product List Add Product Welcome Deborah Log Out

Products

- Garden Cart (GDN-0023)
- Hammer (TBX-0048)
- Leaf Rake (GDN-0011)
- Saw (TBX-0022)
- Video Game Controller (GMG-0042)

Acme Product Management Home Product List Add Product Welcome Deborah Log Out

Edit Product: Garden Cart

Product Name: Garden Cart

Product Code: GDN-0023

Description: 15 gallon capacity rolling garden cart

Buttons: Save, Cancel, Delete

Service

```
private selectedProductSource = new BehaviorSubject<IProduct>(null);  
selectedProductChanges$ =  
    this.selectedProductSource.asObservable();
```

Garden
Cart

```
changeSelectedProduct(selectedProduct: IProduct) {  
    this.selectedProductSource.next(selectedProduct);  
}
```



Demo



Using BehaviorSubject

Product
Shell List

Product
Service

```
this.myBehaviorSubject.next(item);
```

Product
Shell

Product
Shell
Detail



Demo



Cleaning up a Subscription

Product
Shell List

Product
Service

```
this.myBehaviorSubject.next(item);
```

Product
Shell

Product
Shell
Detail



Guidelines & Summary: Communicating Through Service Notifications



Does the Service Need a Subject?

Yes:

- Notifications are required
- Those notification are more than just changes to bound properties

No:

- Notifications are not required
- The only notifications are for changes to bound properties



Which Subject to Use?

Subject

- Requires no initial value
- Broadcasts items as they are pushed

BehaviorSubject

- Requires an initial value
- Provides the current value and then broadcasts items as they are pushed

<http://reactivex.io/documentation/subject.html>



Summary

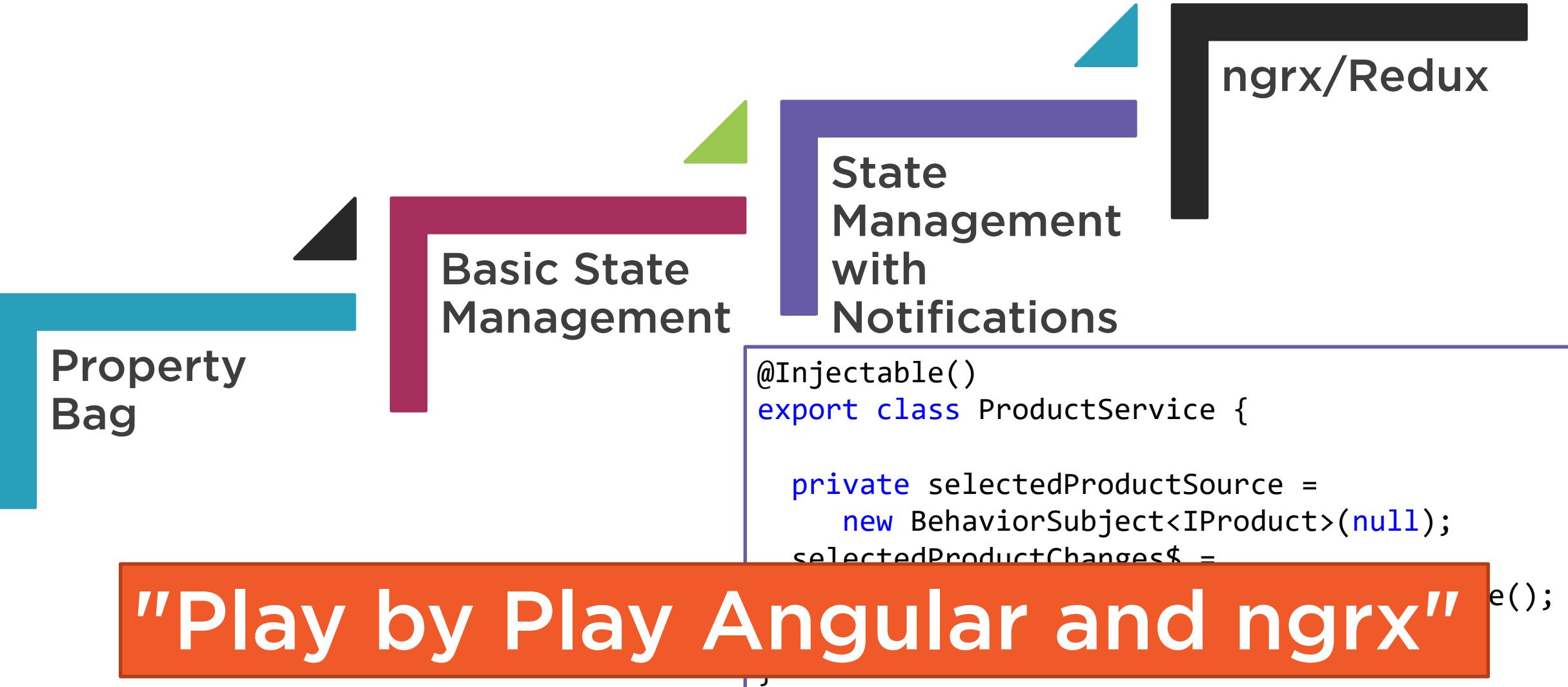


Broadcasting Service Notifications

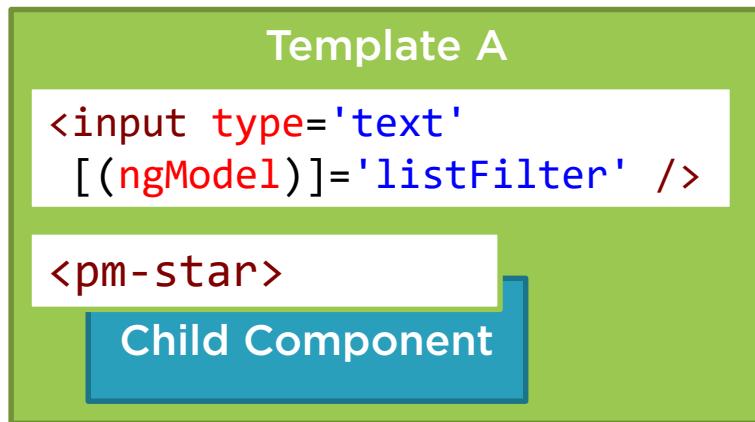
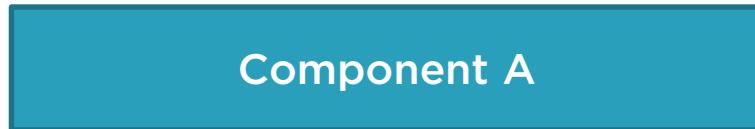
- EventEmitter
- Subject
- BehaviorSubject



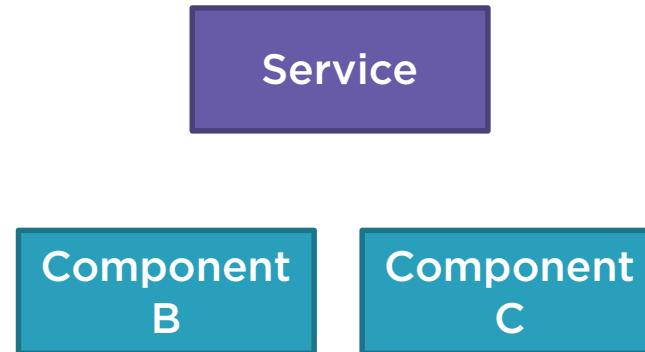
Managing State



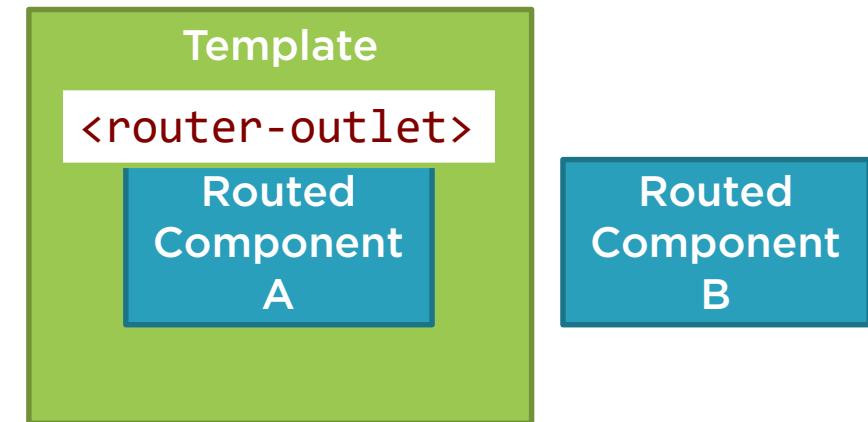
Component Communication



Template



Service



Routing



Communicating Using the Router



Deborah Kurata

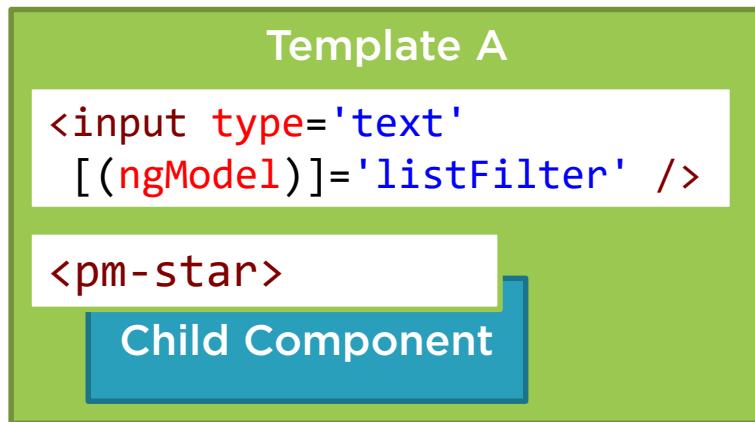
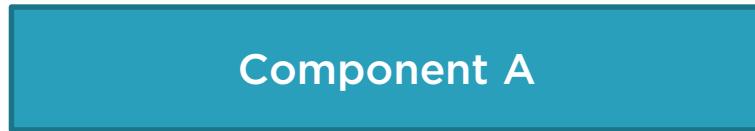
CONSULTANT | SPEAKER | AUTHOR | MVP | GDE

@deborahkurata | blogs.msmvps.com/deborahk/

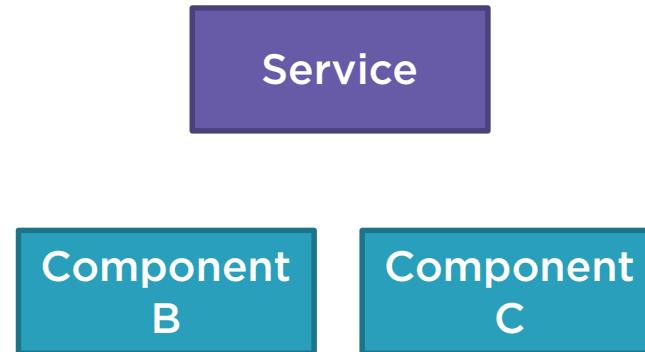




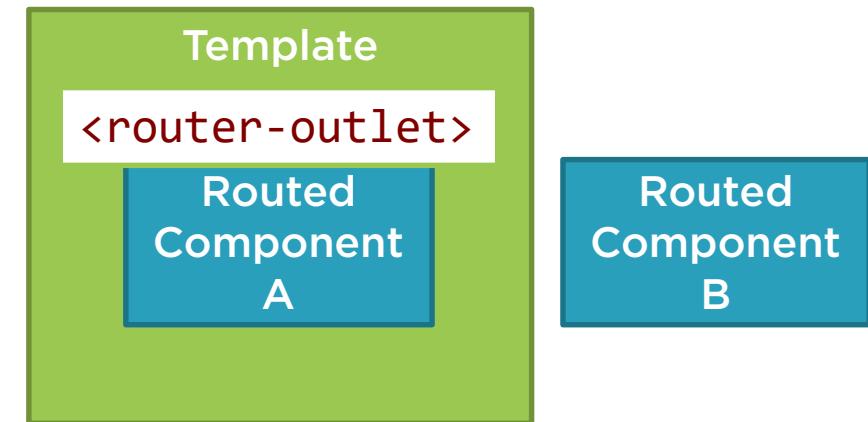
Component Communication



Template



Service



Routing



Module Overview

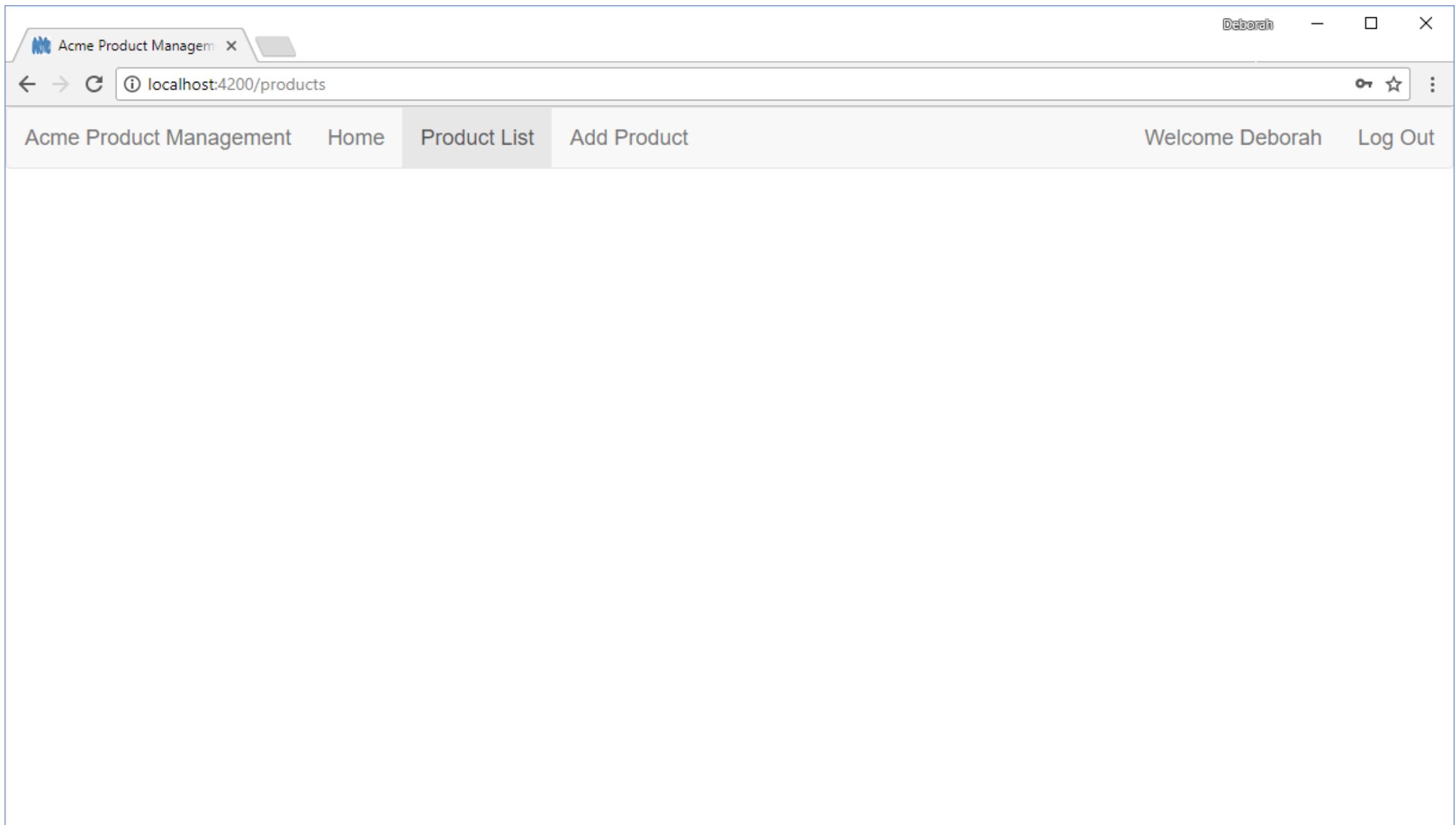


Routing Basics

Route Parameters

- Required
- Optional
- Query





How Routing Works

Acme Product Management ×

localhost:4200

Acme Product Management Home Product List

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

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

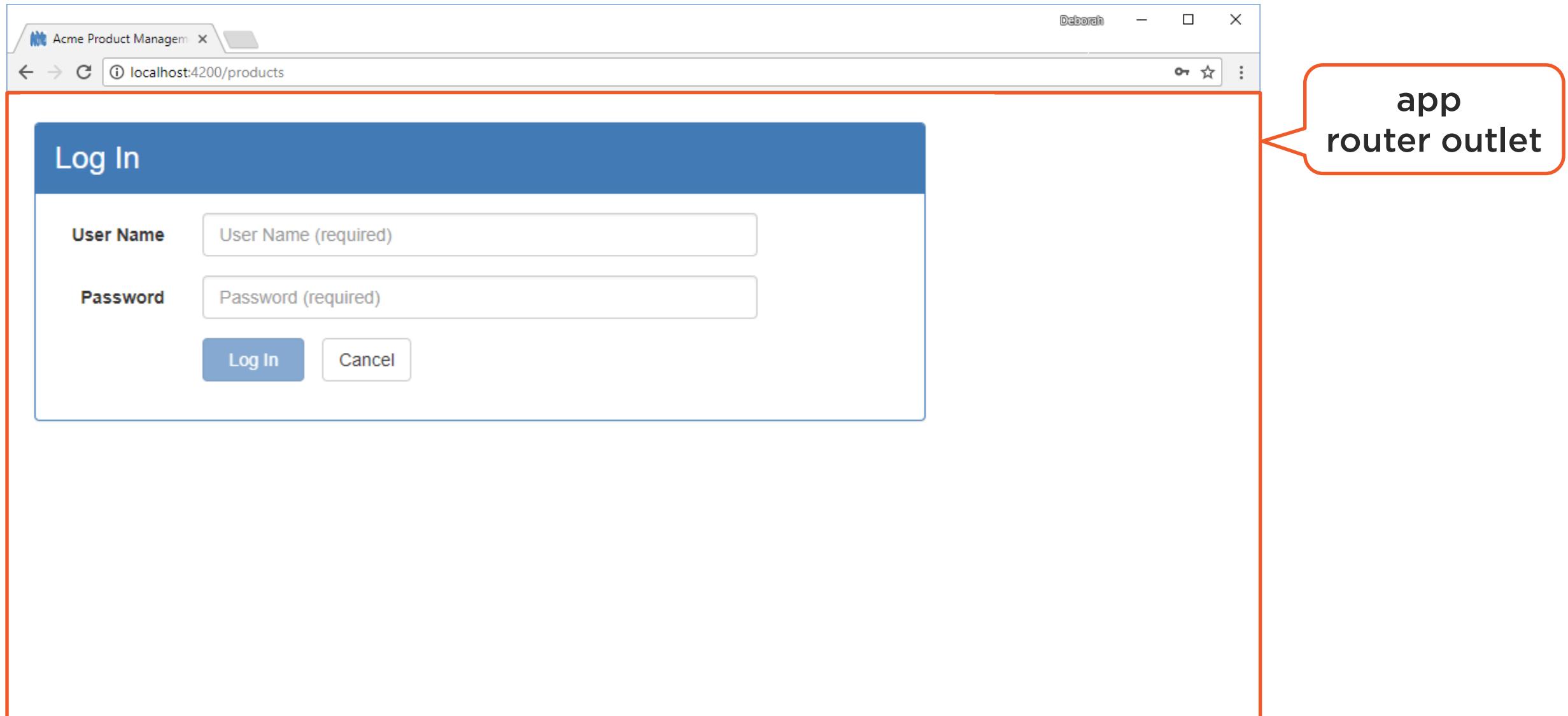
<router-outlet></router-outlet>

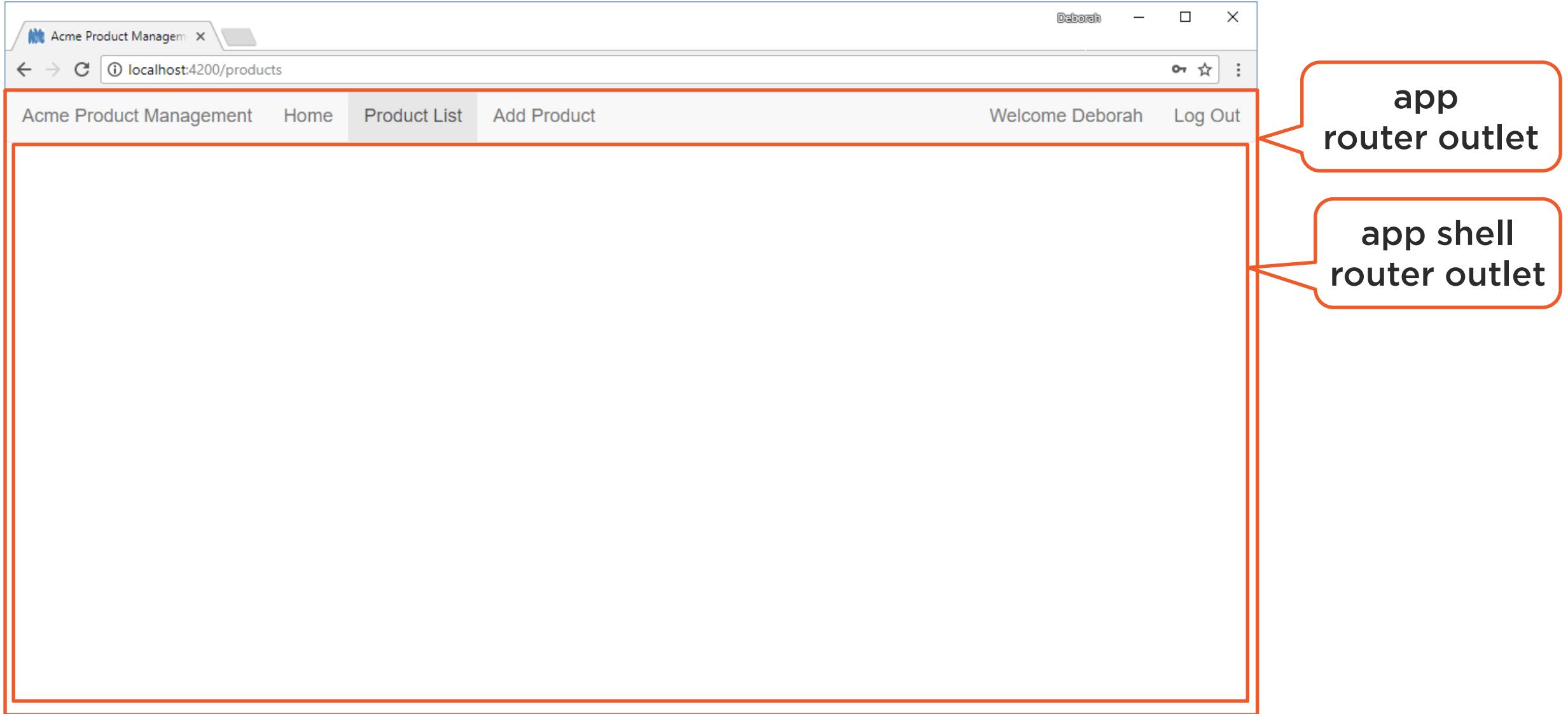
product-list.component.ts

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

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

Deborah





The screenshot shows a web browser window titled "Acme Product Management" with the URL "localhost:4200/products". The page is divided into several sections:

- Header:** "Acme Product Management", "Home", "Product List", "Add Product", "Welcome Deborah", and "Log Out".
- Left Sidebar (app router outlet):** A list of products: Leaf Rake (GDN-0011), Garden Cart (GDN-0023), Hammer (TBX-0048) (highlighted in blue), Saw (TBX-0022), and Video Game Controller (GMG-0042).
- Central Content Area (app shell router outlet):** A "Product Detail: Hammer" section. It displays the following information:
 - Name: Hammer
 - Code: TBX-0048
 - Description: Curved claw steel hammer
 - Availability: May 21, 2017
 - Price: \$8.90
 - Rating: ★★★★★A large image of a hammer is also present.
- Bottom Left:** An "Edit" button.

Three callout boxes with red outlines point to specific parts of the interface:

- An arrow points from the top right to the "app router outlet" area, which covers the header and the sidebar.
- An arrow points from the middle right to the "app shell router outlet" area, which covers the central product detail section.
- An arrow points from the bottom right to the "Product Shell component" area, which covers the entire main content area.



Edit Product: Hammer

Product Name

Hammer

Product Code

TBX-0048

Description

Curved claw steel hammer

[Save](#)[Cancel](#)[Delete](#)

product-module.ts

```
RouterModule.forChild([
  { path: '',
    component: ProductShellComponent },
  { path: ':id',
    component: ProductDetailComponent },
  { path: ':id/edit',
    canDeactivate: [ ProductEditGuard ],
    component: ProductEditComponent
  }
])
```

Number of months on the market: 7

Products	
Leaf Rake (GDN-0011)	
Garden Cart (GDN-0023)	
Hammer (TBX-0048)	
Saw (TBX-0022)	
Video Game Controller (GMG-0042)	

Product Detail: Hammer

Name:	Hammer
Code:	TBX-0048
Description:	Curved claw steel hammer
Availability:	May 21, 2017
Price:	\$8.90
5 Star Rating:	★★★★★



[Edit](#)

product-module.ts

```
RouterModule.forChild([
  { path: '',
    component: ProductListComponent },
  { path: ':id',
    component: ProductDetailComponent },
  { path: ':id/edit',
    canDeactivate: [ ProductEditGuard ],
    component: ProductEditComponent
  }
])
```

Product List					
Show Image	Product	Code	Available	Price	5 Star Rating
	Leaf Rake	GDN-0011	March 19, 2017	\$19.95	★★★
	Garden Cart	GDN-0023	March 18, 2017	\$32.99	★★★★
	Hammer	TBX-0048	May 21, 2017	\$8.90	★★★★★
	Saw	TBX-0022	May 15, 2017	\$11.55	★★★★
	Video Game Controller	GMG-0042	October 15, 2017	\$35.95	★★★★★



Learning More



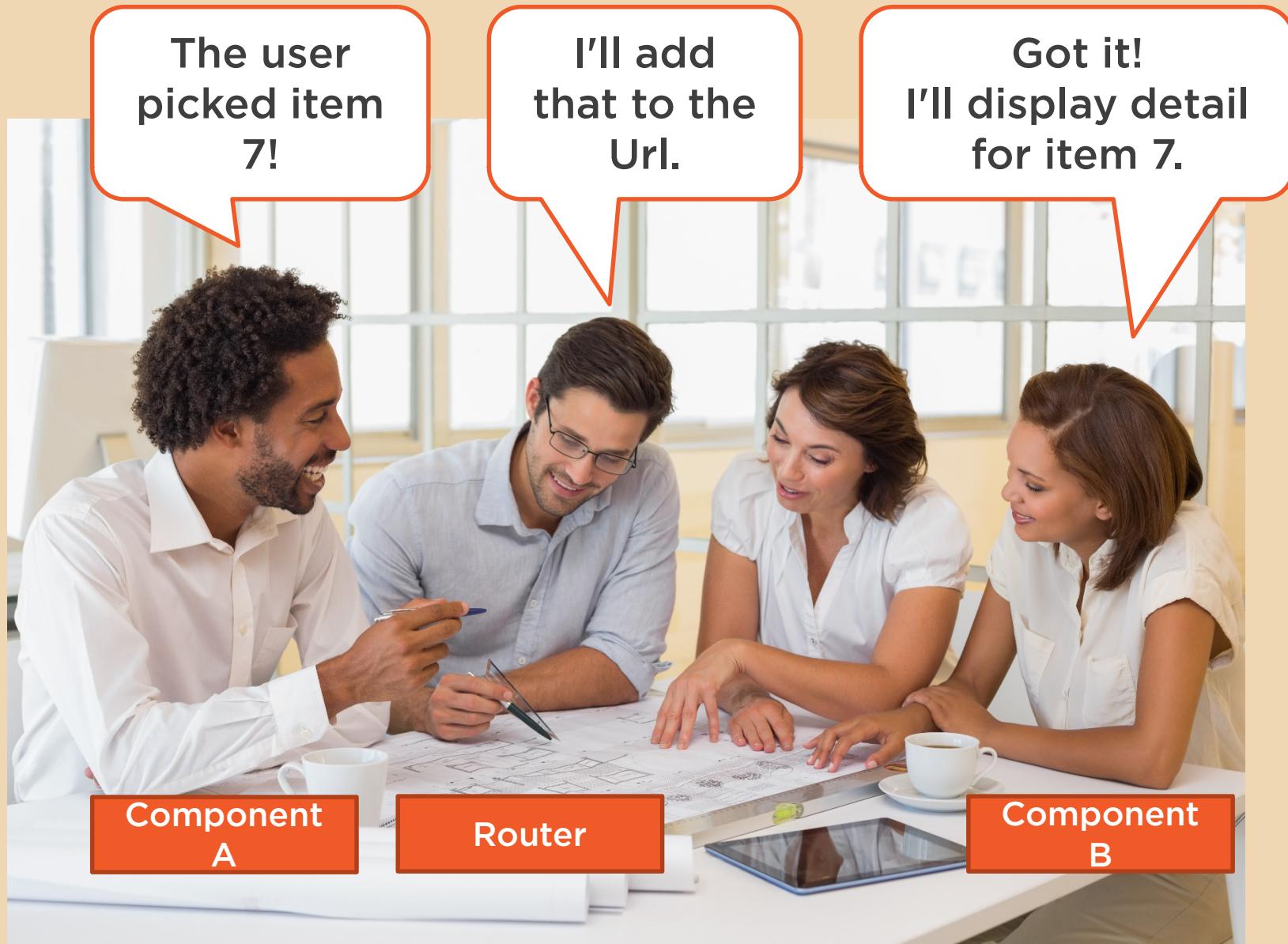
Pluralsight Angular Course

- "Angular Routing"

Book

- "Angular Router" by Victor Savkin
 - <https://leanpub.com/router>





Route Parameters

Product List

Filter by:

Filtered by: Hits:5

Show Image	Product	Code	Available	Price	5 Star Rating	
	Leaf Rake	GDN-0011	March 19, 2017	\$19.95	★★★	Edit
	Garden Cart	GDN-0023	March 18, 2017	\$32.99	★★★★	Edit
	Hammer	TBX-0048	May 21, 2017	\$8.90	★★★★★	Edit
	Saw	TBX-0022	May 15, 2017	\$11.55	★★★★	Edit
	Video Game Controller	GMG-0042	October 15, 2017	\$35.95	★★★★★	Edit

<http://localhost:4200/products/5>

Product Detail: Hammer

Name: Hammer
Code: tbx-0048
Description: Curved claw steel hammer
Availability: May 21, 2017
Price: \$8.90
5 Star Rating: ★★★★★



Back

Edit



Route Parameters



Configure

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

Activate

```
<a [routerLink]=["/products", product.id]>...</a>
```

```
this.router.navigate(['/products', this.product.id]);
```

Resulting Url

```
http://localhost:4200/products/5
```

Read

```
this.route.snapshot.paramMap.get('id');
```





I may have
search params
for you.

If you provide
them I'll pass
them along.

If I get
params, I'll
use them.

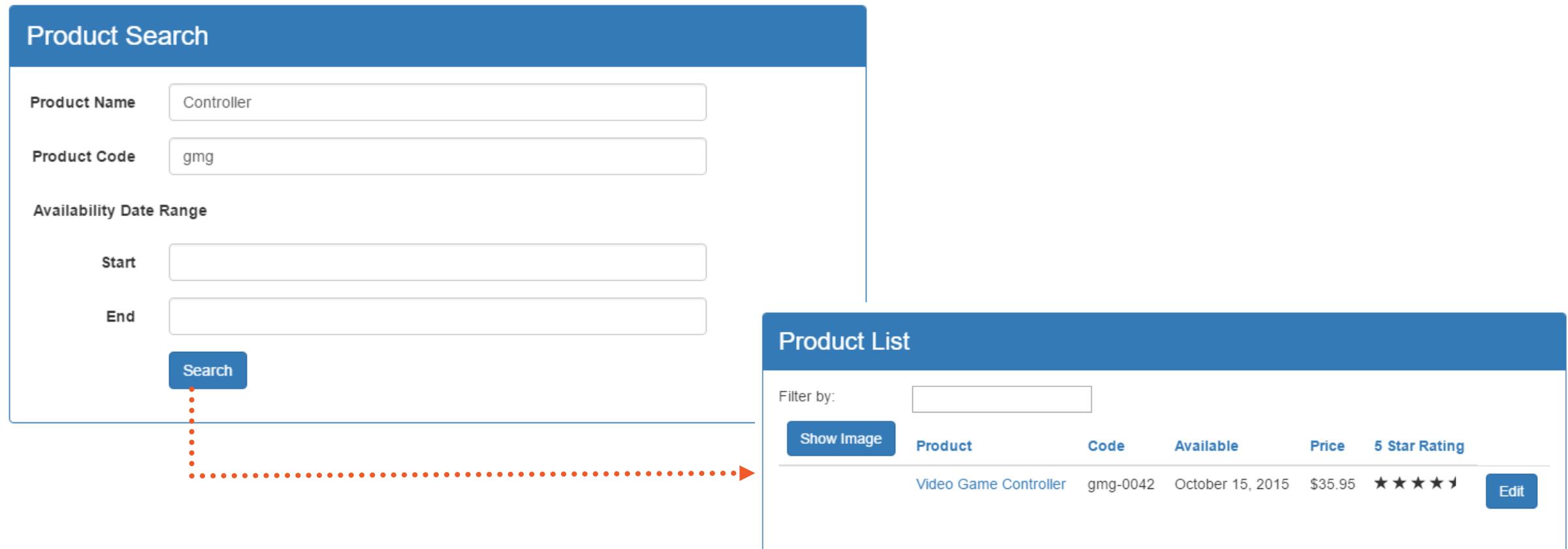
Component
A

Router

Component
B



Optional or Query Parameters



Configure

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

Activate

```
<a [routerLink]="['/products', {name: cart, code: g}]">...</a>  
this.router.navigate(['/products', {name: 'cart', code: 'g'}]);
```

Resulting Url

```
http://localhost:4200/products;name=cart;code=g
```

Read

```
this.route.snapshot.paramMap.get('name');
```



Configure

Query Parameters

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

Activate

```
<a [routerLink]=["/products"]  
  [queryParams]= "{name: cart, code: g}">...</a>  
  
this.router.navigate(['/products'],  
  {queryParams: {name: 'cart', code:'g'}});
```

Resulting Url

<http://localhost:4200/products?name=cart&code=g>

Read

```
this.route.snapshot.queryParamMap.get('name');
```



Query or Optional Parameters

Product Search

Product Name

Product Code

Availability Date Range

Start

End

Search

Product List

Filter by:

Show Image	Product	Code	Available	Price	5 Star Rating	Edit
	Video Game Controller	gmg-0042	October 15, 2015	\$35.95	★★★★★	

**http://localhost:4200/products?
name=Controller&code=gmg**

OR

**http://localhost:4200/products;
name=Controller;code=gmg**



Query Parameters

Product Search

Product Name	<input type="text" value="Controller"/>
Product Code	<input type="text" value="gmg"/>
Availability Date Range	
Start	<input type="text"/>
End	<input type="text"/>
Search	

http://localhost:4200/products?name=Controller&code=gmg

Product List

Show Image	Product	Code	Available	Price	5 Star Rating	Edit
	Video Game Controller	gmg-0042	October 15, 2015	\$35.95	★★★★★	Edit



Route Parameters



Guidelines & Summary: Communicating Using the Router



Route Parameters

Benefits:

- Simple
- Straightforward
- Resulting URLs are bookmarkable and sharable

Considerations:

- Parameters appear in the URL
- Not good for large amounts of data



Which Kind to Use?

Required parameters:

- When the data must be provided to the routed component

Optional parameters:

- When the data is optionally provided to the routed component

Query parameters:

- When the data is optionally provided to the routed component
- When a query style URL is desired
- When the data should be retained across routes



Summary



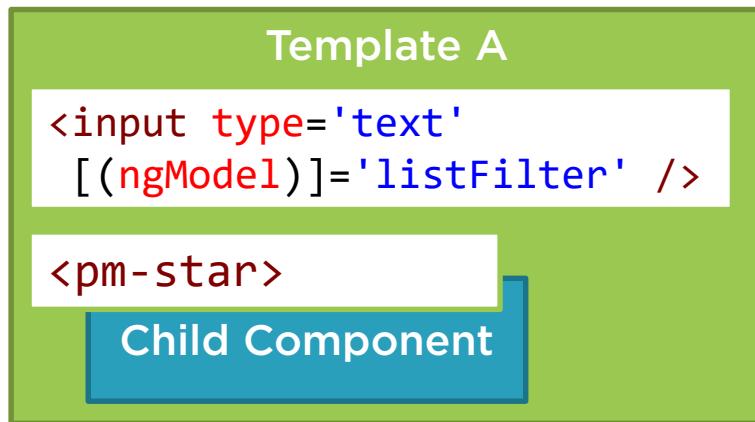
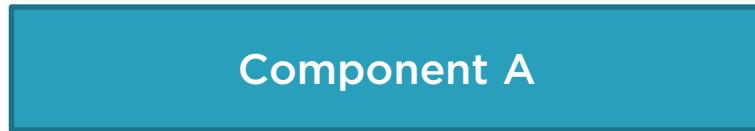
Routing Basics

Route Parameters

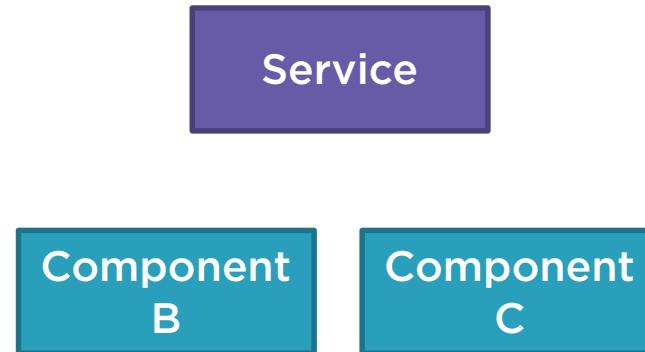
- Required
- Optional
- Query



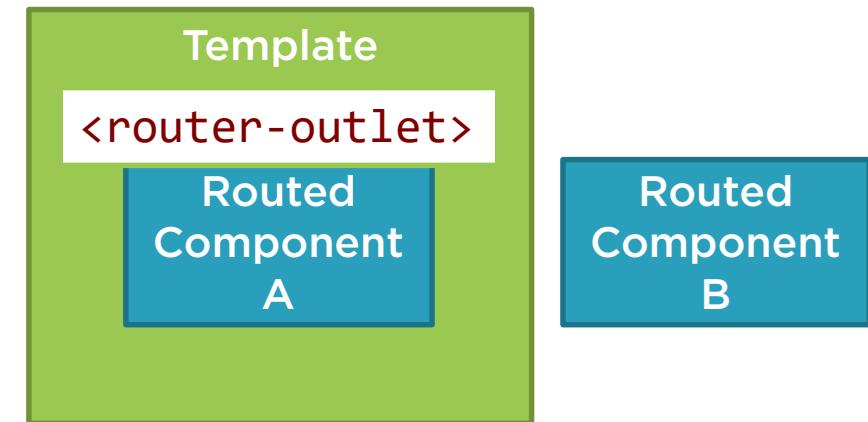
Component Communication



Template



Service



Routing



Final Words



Deborah Kurata

CONSULTANT | SPEAKER | AUTHOR

@deborahkurata | blogs.msmvps.com/deborahk/





Component
A

Service

Template
A

Component
B









Is your form
currently valid?

OK, I'll save.

Yep!







Parent
Component

Child
Component





Will you keep
track of my
current filter?

Sure!
Retaining your
current filter.

What was my
filter?

I have it. Here
you go ...



Component
A

Service

Component
B











I may have
search params
for you.

If you provide
them I'll pass
them along.

If I get
params, I'll
use them.

Component
A

Router

Component
B



Component to Template

Communication:

- Binding
- Structural Directives
- ViewChild/ViewChildren with template reference variable and nativeElement
- ViewChild/ViewChildren with NgForm or NgModel

Change Notification:

- Two-way binding, the long way
- Getters and setters
- ViewChild and valueChanges observable

Component to Child Component

Parent to Child:

- Input properties (@Input)
- Template reference variable
- ViewChild decorator

Change Notification:

- Getters and setters
- onChanges lifecycle hook

Child to Parent:

- Output properties (@Output)

Component to Component

Service:

- Simple properties
- Getters and setters
- State management
- Subject
- BehaviorSubject
- ngrx/Redux

Router:

- Required parameters
- Optional parameters
- Query parameters



Learning More

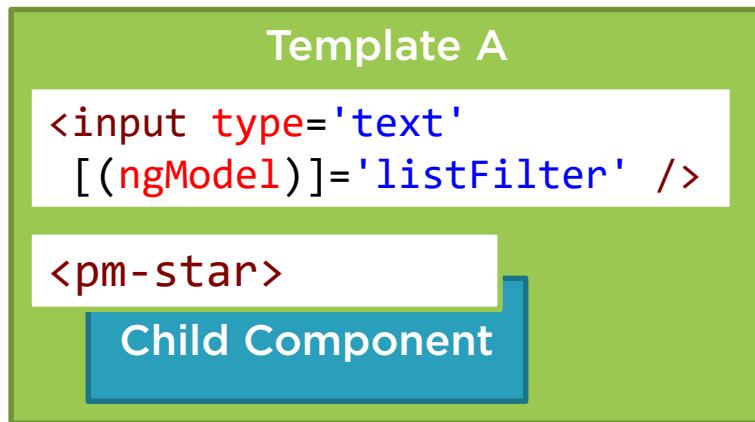
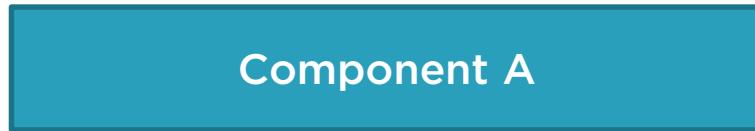


Angular Courses on Pluralsight

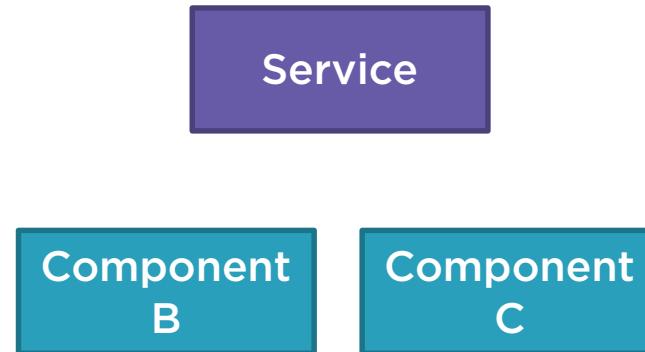
- "Play by Play Angular and ngrx"
- "Angular Forms"
- "Angular Reactive Forms"
- "Angular Routing"



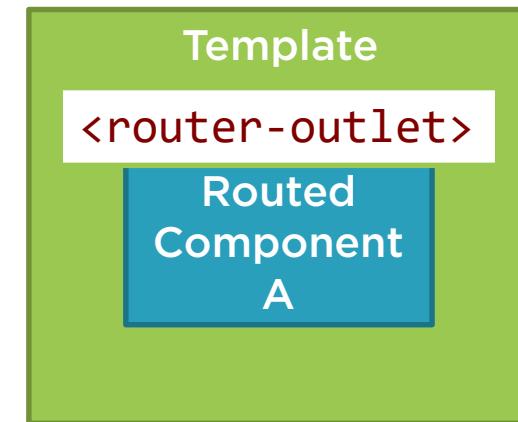
Component Communication



Template



Service



Routing

@deborahkurata

