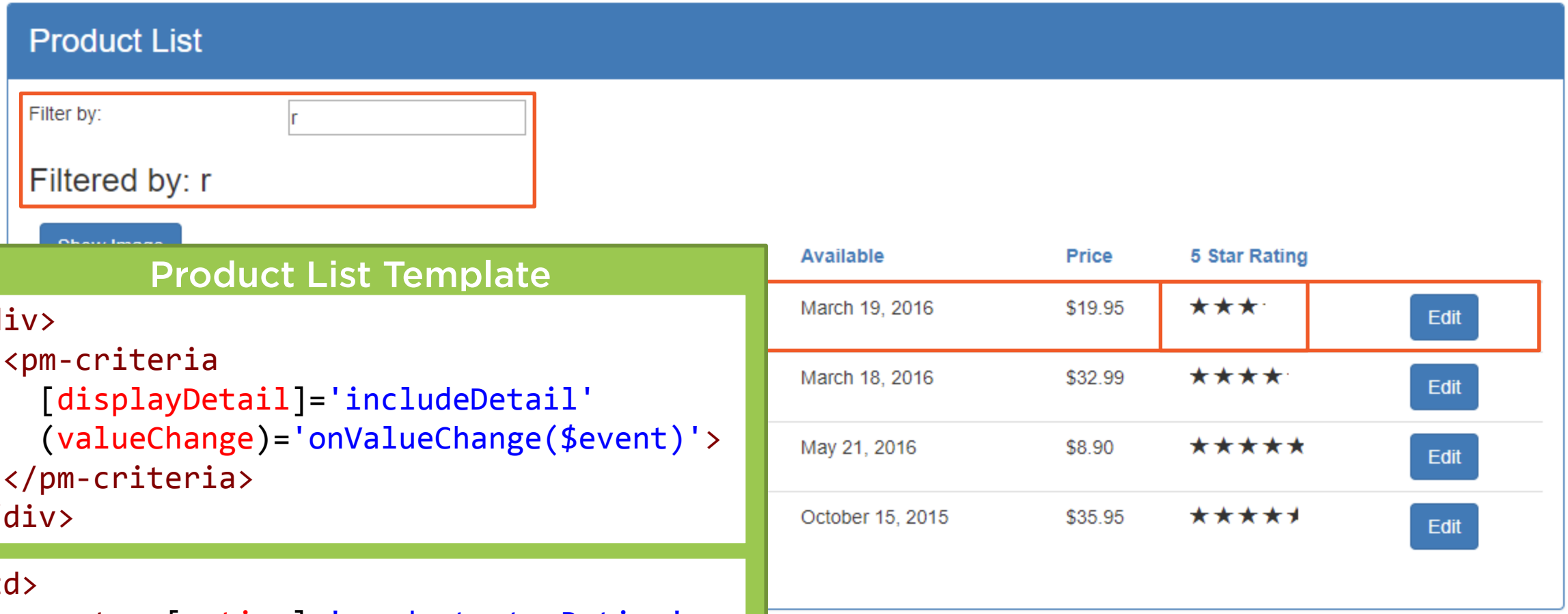
Communicating with a Child Component

Deborah Kurata
CONSULTANT | SPEAKER | AUTHOR | MVP | GDE

@deborahkurata | blogs.msmvps.com/deborahk/

Child Components

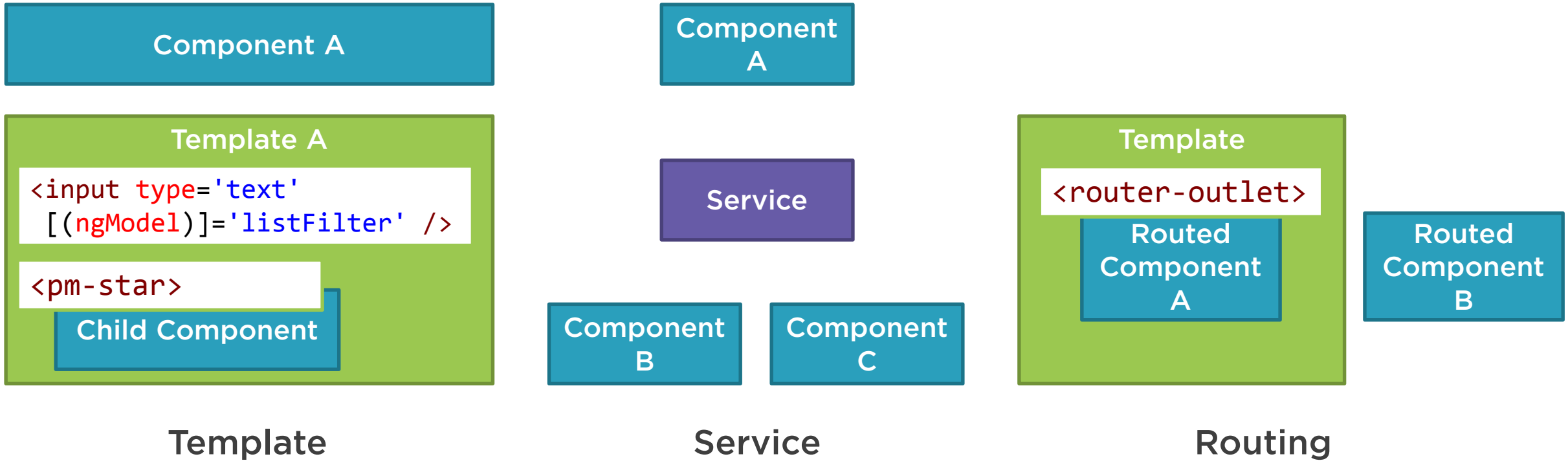


Product List Template

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

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

Component Communication



Module Overview

Child Components

Parent to Child Communication

Input Property

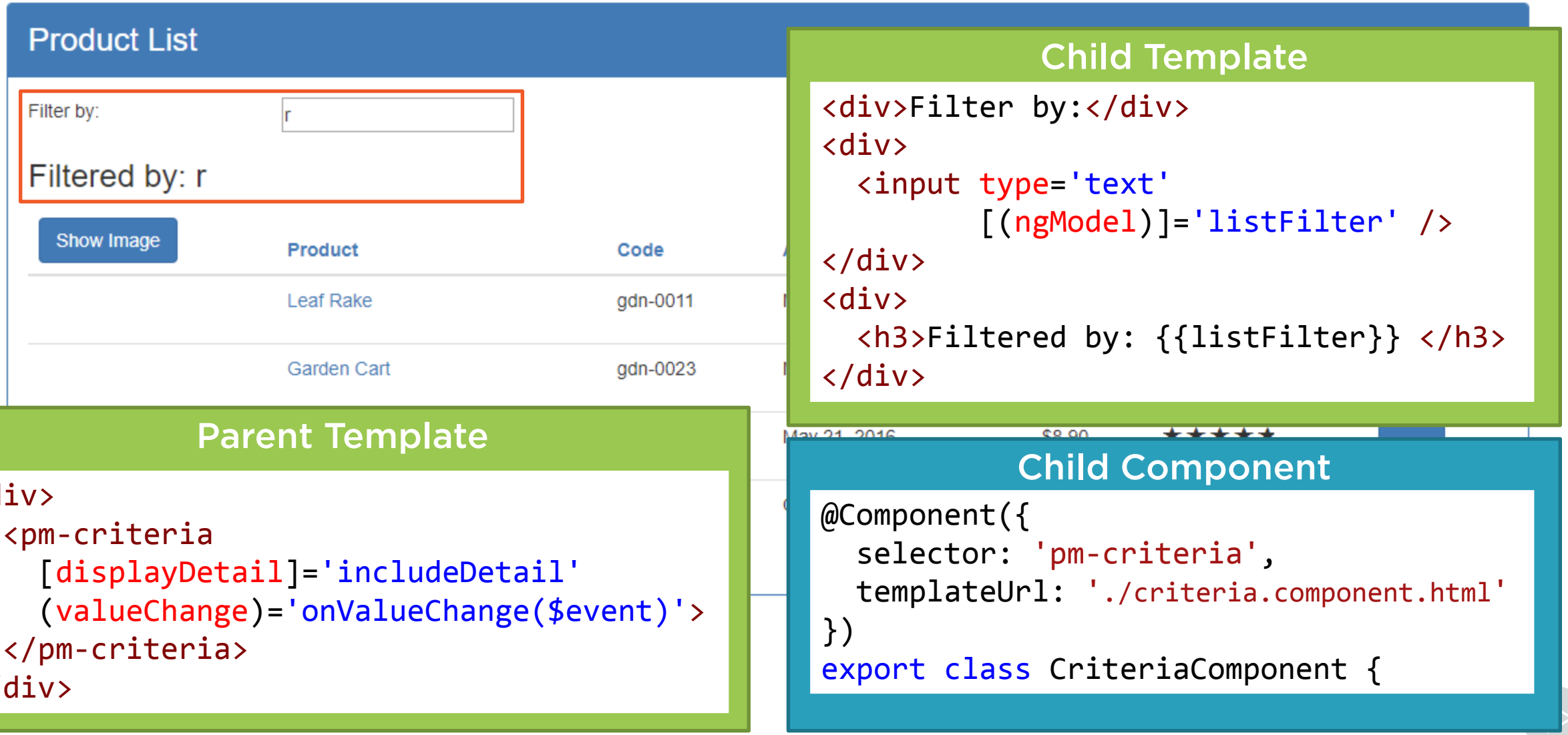
Watching for Changes

- Getter and Setter
- OnChanges Lifecycle Hook

Template Reference Variable

ViewChild Decorator

Building Child Components



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 {
```


Hey kiddo!
I have a new
value for you.
I'll update my
display.
Thanks, Dad!
Parent
Component
Child
Component

Parent to Child

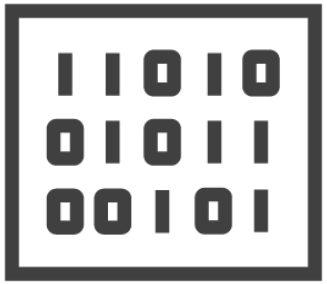
@Input

Getter/Setter

OnChanges

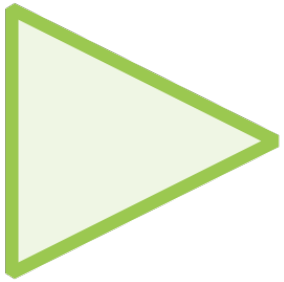
Configuration

Default value



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

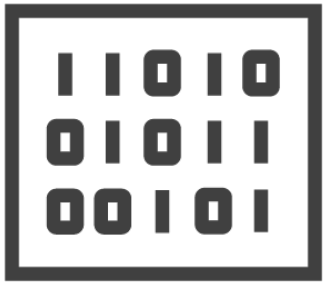
@Input

Getter/Setter

OnChanges

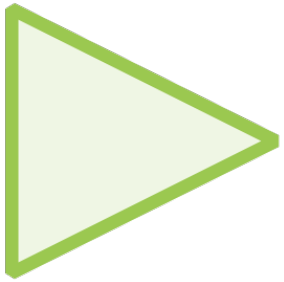
Configuration

Default value



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;
```

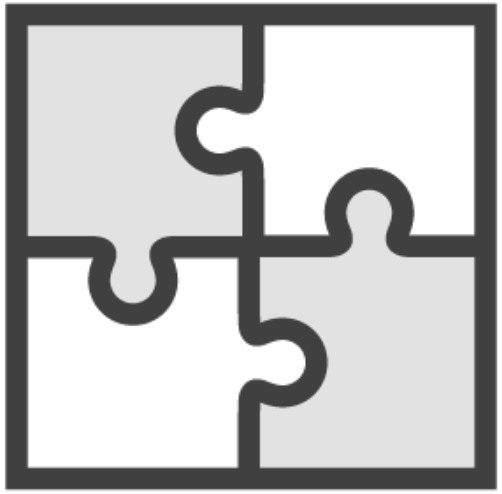
```
listFilter: string;
```

```
clear(): void { ...}
```

Guidelines & Summary: Communicating with a Child Component

Defining Child Components

Specific Task



Complex

Reusable

Parent to Child

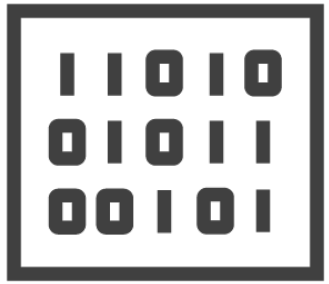
@Input

Getter/Setter

OnChanges

Configuration

Default value



Item data

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

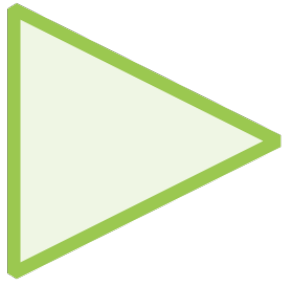
Template Reference Variable:

- Use from the parent's template

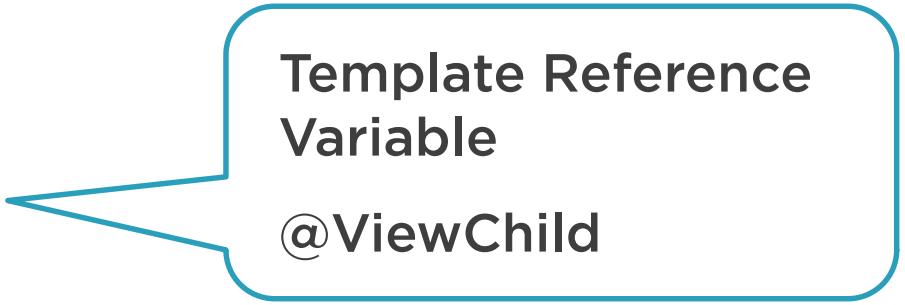
ViewChild:

- Use from the parent's class

Request information



Perform an Action



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