

#### Contents

- Interacting with Content
- Interacting with View
- Working with Handles
- Providers vs. ViewProviders



### Why is this Intersting?

- Reusable Components (Component Libraries)
- Better Understanding for Angular

# Interacting with a Component's Content



### Case Study #1: Tabbed Pane

| Upcoming Flights Operated Flights Cancelled Flights |         |           |                  |
|---|---------|-----------|------------------|
| Upcoming Flights                                    |         |           |                  |
| 1   | Hamburg | Berlin    | 01.02.2025 17:00 |
| 2   | Hamburg | Frankfurt | 01.02.2025 17:30 |
| 3   | Hamburg | Mallorca  | 01.02.2025 17:45 |



#### Tabbed Pane

```
<app-tabbed-pane>
   <app-tab title="Upcoming Flights">
       No upcoming flights!
   </app-tab>
   <app-tab title="Operated Flights">
       No operated flights!
   </app-tab>
   <app-tab title="Cancelled Flights">
       No cancelled flights!
   </app-tab>
</app-tabbed-pane>
```



# DEMO



#### View vs. Content



#### View vs. Content

```
@Component({
                                                    <tab title="Booked">
selector: 'tab',
                                                                                    Content
                                     View
                                                      > Sample Text ...
<u>template: `</u>
    <div *ngIf="visible">
                                                    </tab>
        <h1>{{title}}</h1>
        <div>
             <ng-content></ng-content>
        </div>
    </div>
})
export class TabComponent {
    @Input() title: string;
    visible: boolean = true;
```



#### Hooks

- 1) ngOnChanges
- 2) ngOnInit
- 3) ngDoCheck
- 4) ngAfterContentInit
- 5) ngAfterContentChecked
- 6) ngAfterViewInit
- 7) ngAfterViewChecked
- 8) ngOnDestroy



#### Hooks

- 1) ngOnChanges
- 2) ngOnInit
- 3) ngDoCheck
- 4) ngAfterContentInit
- 5) ngAfterContentChecked
- 6) ngAfterViewInit
- 7) ngAfterViewChecked
- 8) ngOnDestroy



# DEMO



#### Handles



#### Handles

```
<app-tabbed-pane #pane>
  [...]
</app-tabbed-pane>

Current Page: {{ pane.currentPage }}
```

# DEMO



#### ViewProviders



#### Providers

```
@Component({
    providers: [NavigatorService]
})
export class TabbedPaneComponent {
    [...]
}
```

#### Providers

```
@Component({
    providers: [NavigatorService] // Visible in View and Content
})
export class TabbedPaneComponent {
    [...]
}
```

#### View Providers

```
@Component({
    viewProviders: [NavigatorService] // Visible only in View
})
export class TabbedPaneComponent {
    [...]
}
```

# DEMO



# LAB



### Summary

- Content vs. View
- [Content | View] [Child | Children]
- Providers vs. ViewProviders
- Handles

