



ANGULAR  
**ARCHITECTS**  
INSIDE KNOWLEDGE

# Components Deep Dive

[angular-architects.io](https://angular-architects.io)

# Contents

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



ANGULAR  
**ARCHITECTS**  
INSIDE KNOWLEDGE



SOFTWARE  
**ARCHITECT**

# Why is this Interesting?

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



ANGULAR  
**ARCHITECTS**  
INSIDE KNOWLEDGE



SOFTWARE  
**ARCHITECT**

# 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



ANGULAR  
**ARCHITECTS**  
INSIDE KNOWLEDGE



SOFTWARE  
**ARCHITECT**

# Tabbed Pane

```
<app-tabbed-pane>
  <app-tab title="Upcoming Flights">
    <p>No upcoming flights!</p>
  </app-tab>

  <app-tab title="Operated Flights">
    <p>No operated flights!</p>
  </app-tab>

  <app-tab title="Cancelled Flights">
    <p>No cancelled flights!</p>
  </app-tab>
</app-tabbed-pane>
```



ANGULAR  
ARCHITECTS  
INSIDE KNOWLEDGE



SOFTWARE  
ARCHITECT

# DEMO



ANGULAR  
**ARCHITECTS**  
INSIDE KNOWLEDGE

# View vs. Content



ANGULAR  
**ARCHITECTS**  
INSIDE KNOWLEDGE



# View vs. Content

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

**View**

**Content**

Sample Text ...



ANGULAR  
**ARCHITECTS**  
INSIDE KNOWLEDGE



SOFTWARE  
**ARCHITECT**

# Hooks

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



ANGULAR  
**ARCHITECTS**  
INSIDE KNOWLEDGE



SOFTWARE  
**ARCHITECT**

# Hooks

- 1) ngOnChanges
- 2) ngOnInit
- 3) ngAfterContentInit**
- 4) ngAfterViewInit**
- 5) ngOnDestroy



# DEMO



ANGULAR  
**ARCHITECTS**  
INSIDE KNOWLEDGE

# Handles



# Handles

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

Current Page: {{ pane.currentPage }}



ANGULAR  
ARCHITECTS  
INSIDE KNOWLEDGE



SOFTWARE  
ARCHITECT

# DEMO



ANGULAR  
**ARCHITECTS**  
INSIDE KNOWLEDGE

# ViewProviders



ANGULAR  
**ARCHITECTS**  
INSIDE KNOWLEDGE



# Providers

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



ANGULAR  
**ARCHITECTS**  
INSIDE KNOWLEDGE



SOFTWARE  
**ARCHITECT**

# Providers

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



ANGULAR  
ARCHITECTS  
INSIDE KNOWLEDGE



SOFTWARE  
ARCHITECT

# View Providers

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



ANGULAR  
**ARCHITECTS**  
INSIDE KNOWLEDGE



SOFTWARE  
**ARCHITECT**

# LAB



ANGULAR  
**ARCHITECTS**  
INSIDE KNOWLEDGE

# Summary

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



ANGULAR  
**ARCHITECTS**  
INSIDE KNOWLEDGE



SOFTWARE  
**ARCHITECT**