

ANGULAR WORKSHOP

COMPONENTS & DATA BINDING

Naveen Pete

Friday, June 23, 2017

Agenda

- What is Angular?
- Angular Building Blocks
- Components
- Decorators
- Component Templates & Styles
- Data Binding
 - Interpolation
 - Property Binding
 - Event Binding
 - Two Way Binding
- Directives
- Q & A

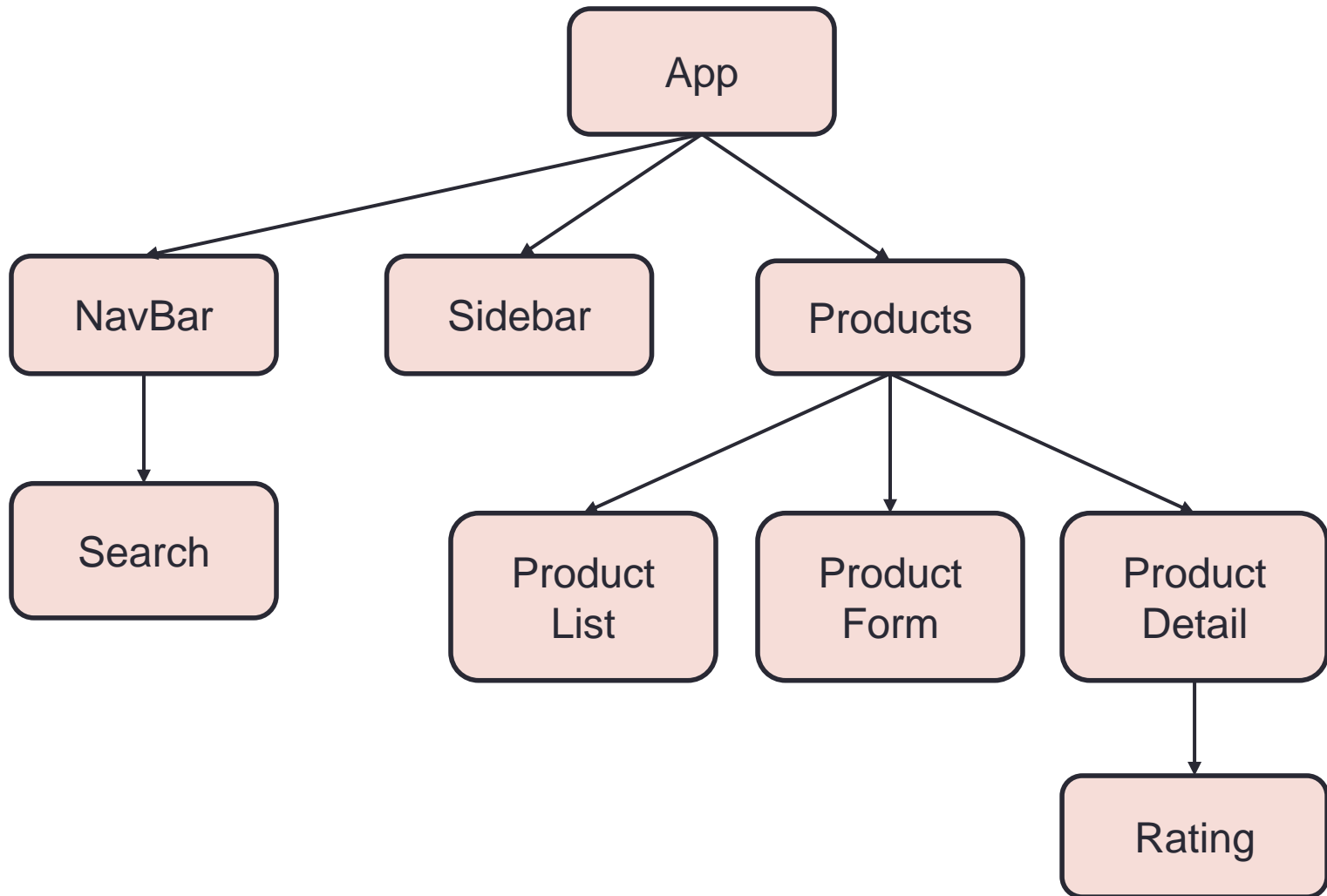
What is Angular?

- Framework for building dynamic apps for different platforms – Web, Mobile, Desktop
- Create apps that are modular, maintainable, testable

Angular Building Blocks

- Components
 - Encapsulates the template, data and the behavior of a view
 - Completely decoupled from DOM
- Services
 - Encapsulates any non UI logic
 - Http calls, logging, business logic, etc
 - Any logic not related to a view is delegated to a service
- Routers
 - Responsible for navigation from one view to another
- Directives
 - To modify DOM elements and/or extend their behavior
 - Built-in or custom
- Modules
 - A block of highly related classes

Components



Components

- Key feature of Angular
- Encapsulate the template, data and the behavior of a view
- Allows you to break a complex web page into smaller, manageable & reusable parts
- Plain TypeScript class
- App component
 - Root component
 - Holds our entire application
 - Other components are added to App component
- A Component has its own
 - Template – HTML markup
 - Style – CSS styles
 - Business logic (data and behavior) – TypeScript code
- Promotes
 - Reusability
 - Maintainability
 - Testability

Decorators

- Extends the behavior of a class / function without explicitly modifying it
- Attaches metadata to classes

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

@Component({
  selector: 'app-server',
  templateUrl: 'server.component.html'
})
export class ServerComponent {
}
```

Component Templates & Styles

- Templates
 - templateUrl property – external template file
 - template property – inline template
- Styles
 - styleUrls property – external stylesheet file(s)
 - styles property – inline styles

Exercise

- Create a component – Product
 - Add a reference to Product component in App component
 - Check the output
- Create another component – Products
 - Use Angular CLI command to create component
 - `ng generate component products`
 - Add 2 references of Product component within Products component
 - Add reference to Products component in App component
 - Check the output

Exercise

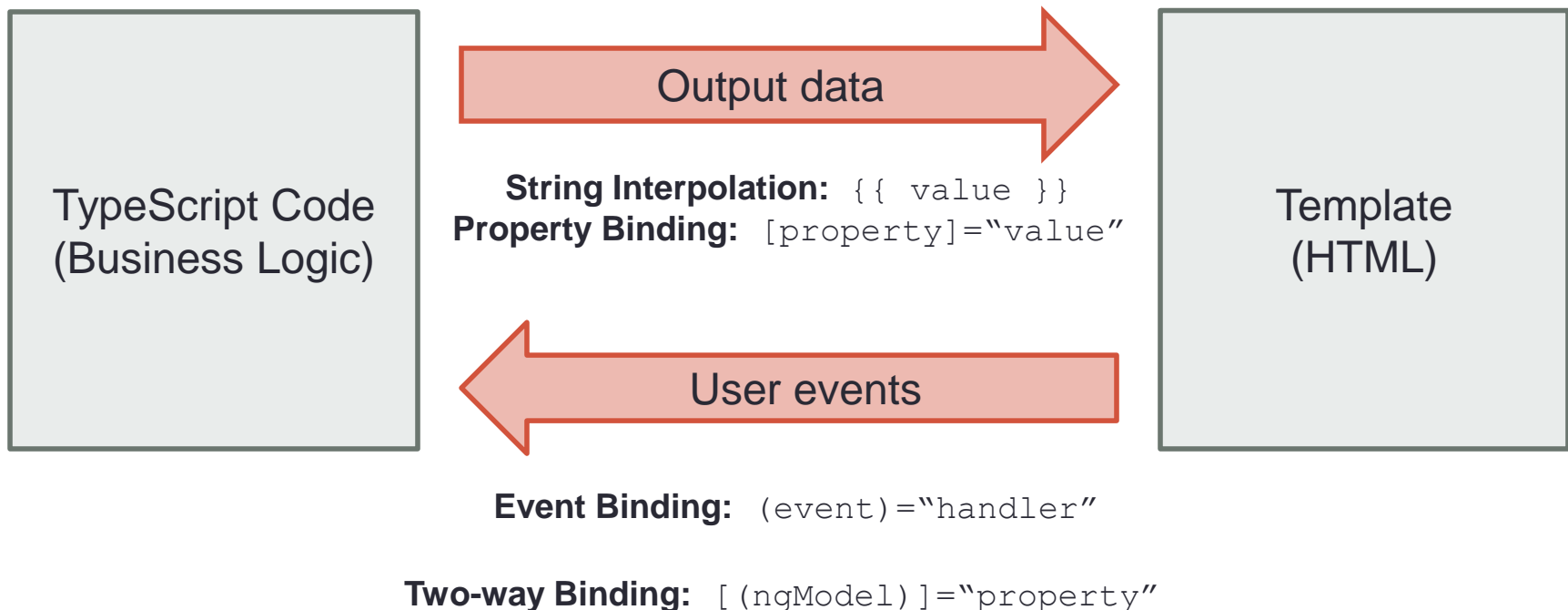
- Creating a new component
 - Create a new file, for e.g., product.component.ts
 - Create a class – ProductComponent
- Understanding Decorator
 - Add decorator - @Component()
 - import { Component } from '@angular/core';
 - Provide metadata within @Component decorator
 - selector, templateUrl
- Understanding AppModule
 - Register ProductComponent within 'declarations' array
 - Import ProductComponent into AppModule
- Using a component
 - Use the selector <app-product></app-product> within app component template

Exercise

- Creating a component with CLI
 - ng generate component products
 - ng g c products

Data Binding

- Communication between the TypeScript code and the HTML template



Data Binding

- String Interpolation
 - {{ }}
- Property Binding
 - []
- Event Binding
 - \$event – Passing event data
- Two-way Data Binding
 - [(ngModel)]
 - Note: FormsModule should be imported in AppModule (imports[] array) to use ngModel

Directives

- Instructions in the DOM
- Components are directives with template
- Can be built-in or custom
- Built-in directives
 - Structural directives – add / remove elements to / from the DOM
 - *ngIf
 - *ngFor
 - Attribute directives – make changes to the element
 - ngStyle
 - ngClass

Q & A

- Thank you!