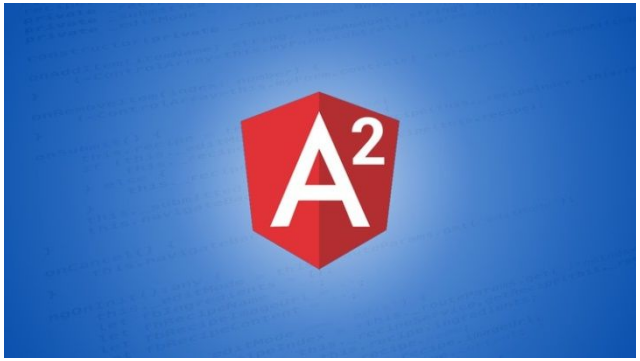
UNIVERSITAS SCIENTIARUM SZEGEDIENSIS
UNIVERSITY OF SZEGED
Department of Software Engineering

Program Systems Development practice

Practice 9

Angular 2



Angular

- JavaScript-based open-source front-end web-application framework
- Maintained by Google
- Uses the MVC architecture
- Initial release in 2010

Angular 2+

- Not an upgrade of Angular 1, but a complete rewrite
- Main differences:
  - Mobile development
  - Modularity
  - Recommends the TypeScript language
  - Improved dependency injection
  - Simpler routing

Modules

- Great way to organize the application and extend it with capabilities from external libraries
- Modules consolidate components, directives, etc… into cohesive blocks of functionality
- Created by the **`@NgModule`** decorator

Bootstrapping

- Imported via ***angular2/platform/browser*** module
- The **`bootstrap()`** here is a method, which starts the application by loading the main component into the component tree

Components

- In Angular 2+, ”everything is a component”
- They hold the logic of the page
- Defining Components with the `@Component` decorator

Parts of the component

- Defined by the `@Component({})` decorator
- `selector`: is used to access the template in other component's HTML code
- `template`: explicit HTML code of the component
- `templateUrl`: the path of the HTML file which contains the template of the component
- `export class ComponentName { }`: is used to make the component accessible for other components

Lifecycle Hooks

- constructor()
- ngOnChanges()
- ngOnInit()
- ngDoCheck()
  - ngAfterContentInit()
  - ngAfterContentChecked
  - ngAfterViewInit()
  - ngAfterViewChecked()
- ngOnDestroy()

`ngOnInit()` method

- Works very similar to the constructor
- Conventions:
  - Use the **`constructor()`** to initialize the class members
  - Use the **`ngOnInit()`** for all the initialization/declaration stuff instead of class members
    - `import {OnInit} from '@angular/core';`
    - `export class MyComponent implements OnInit { }`

Navigation among the components

- The different components are available on different URL parts

  (e.g.: **MainComponent** is available at `localhost:4200/main`
- Defined by the routes
- It uses the **Router** module of Angular
- Routes are defined with **path** and **component** pairs
- The root component has to traverse the routing tree first

Navigation among the components

- Navigation can be done in two ways
  - Implemented in the View part
    - Using the `<a routerLink="””></a>` tags.
  - Implemented in the Component
    - Use a DOM-event (e.g.: a click event)
      - » Define an event on one DOM-element and specify the method that should be triggered when the event fires
    - Define the method in the component
      - » Define the method, and use the Router component from the Router module of the Angular with its navigate method.
      - » With this solution, you can also pass parameters.

Working with route parameters

- Through the navigation, you can pass parameters (e.g.: a user ID)
- The sent parameters can be received in another component.
  - To work with a received parameter, use the ActivatedRoute component from the Router module of Angular.