KOENIG
step forward

ANGULAR 13

Table of Contents

- Two Way Data Binding
- Form Validation
- Angular Validators
- Displaying Validation State Using Classes
- Additional Input Types
- Checkboxes
- Select (Drop Down) Fields
- Rendering Options for Select (Drop Down)
- Date fields
- Radio Buttons

- **Chapter 7. Reactive Forms**
  - Reactive Forms Overview
  - The Building Blocks
  - Import ReactiveFormsModule
  - Construct a Form
  - Design the Template
  - FormControl Constructor
  - Getting Form Values
  - Setting Form Values
  - The Synchronous Nature
  - Subscribing to Input Changes
  - Validation
  - Built-In Validators
  - Showing Validation Error
  - Custom Validator
  - Using a Custom Validator

- **Chapter 8. Services and Dependency Injection**
  - What is a Service?
  - Creating a Basic Service
  - The Service Class
  - What is Dependency Injection?
  - Injecting a Service Instance
  - Injectors

- **Chapter 9. Pipes and Data Formatting**
  - What are Pipes?
  - Built-In Pipes
  - Using Pipes in HTML Template
  - Chaining Pipes
  - The number Pipe
  - Currency Pipe
  - Create a Custom Pipe
  - Custom Pipe Example
  - Using Custom Pipes
  - Using a Pipe with ngFor
  - A Filter Pipe

- **Chapter 10. HTTP Client**
  - The Angular HTTP Client
  - Using The HTTP Client - Overview
  - Importing HttpClientModule
  - Simple Example
  - Service Using HttpClient
  - ES6 Import Statements
  - Making a GET Request
  - What does an Observable Object do?
  - Using the Service in a Component
  - Error Handling
  - Customizing Error Object with. catch ()
  - Making a POST Request
  - Making a PUT Request
  - Making a DELETE Request

- **Chapter 11. The Angular Component Router**
  - The Component Router
  - View Navigation
  - The Angular Router API
  - Creating a Router Enabled Application
  - Hosting the Routed Components
  - Navigation Using Links and Buttons
  - Programmatic Navigation
  - Passing Route Parameters
  - Navigating with Route Parameters
  - Obtaining the Route Parameter Values
  - Retrieving a Route Parameter
  - Routing Enabled Feature Module
  - Using the Feature Module
  - Lazy Loading the Feature Module
  - Creating Links for the Feature Module
  - Components
  - More About Lazy Loading
  - Preloading Modules
  - routerLinkActive binding
  - Default Route
  - Wildcard Route Path
  - redirectTo
  - Child Routes
  - Defining Child Routes
  - for Child Routes
  - Links for Child Routes
  - Navigation Guards
  - Creating Guard Implementations
  - Using Guards in a Route

- **Chapter 12. Advanced HTTP Client**
    - Request Options
    - Returning an HttpResponse Object
    - Setting Request Headers
    - Creating New Observables
    - Creating a Simple Observable
    - The Observable.create() Method
    - Observable Operators
    - More About map
    - Piping Operators
    - The flatMap() Operator
    - The tap() Operator
    - The zip() Operator
    - Caching HTTP Response
    - Making Sequential HTTP Calls
    - Making Parallel Calls
    - Customizing Error Object with catchError ()

- **Chapter 16. Unit Testing Angular Applications**
    - Unit Testing Angular Artifacts
    - Testing Tools
    - Typical Testing Steps
    - Test Results
    - Jasmine Test Suites
    - Jasmine Specs (Unit Tests)
    - Expectations (Assertions)
    - Matchers
    - Examples of Using Matchers
    - Using the not Property
    - Setup and Teardown in Unit Test Suites
    - Example of before Each and after Each Functions
    - Angular Test Module
    - Example Angular Test Module
    - Testing a Service
    - Injecting a Service Instance
    - Test a Synchronous Method
    - Test an Asynchronous Method
    - Using Mock HTTP Client
    - Supplying Canned Response
    - Testing a Component
    - Component Test Module
    - Creating a Component Instance
    - The Component Fixture Class
    - Basic Component Tests
    - The Debug Element Class