



Certified Angular 8 Developer

Vskills Certifications

Vskills Brochure



Certified Angular 8 Developer

Angular 8 is the latest version of Angular, this course starts from scratch, from setup to deployment. This course will help you to learn all about components, directives, services, forms, HTTP access, authentication, optimizing an Angular app with modules and offline compilation, and much more.

This course will also show you how to use the Angular CLI and feature a complete project, which allows you to practice the things learned throughout the course.

Angular is one of the most modern, performance-efficient and powerful frontend frameworks you can learn today. It allows you to build great web apps which offer awesome user experiences

In this online course, you'll learn all the fundamentals about modules, directives, components, data binding, routing, HTTP access, and providing full CRUD operations.

Pre-requisite: Knowledge of JavaScript, HTML/HTML5, computer networking and internet.

Why should one take this certification?

The certification is suitable for web developers, programmers and graduates wanting to excel in web application development areas. It is also well suited for those who are already working in web development and using angular and would like to take certification for further career progression in web application development.

Earning Vskills Angular 8 Developer Certification can help candidate differentiate in today's competitive job market, broaden their employment opportunities by displaying their advanced skills, and result in higher earning potential.

Who will benefit from taking this certification?

Job seekers looking to find employment in IT or software development departments of various companies, students who want to learn angular 8 and implement the same in their web development journey.

Test Details

- **Duration:** 60 minutes
- **No. of questions:** 50
- **Maximum marks:** 50, Passing marks: 25 (50%)

Certified Angular 8 Developer

There is no negative marking in this module.

Fee Structure

Rs. 3,999/- (Excludes taxes) *

* Fees may change without prior notice, please refer <http://www.vskills.in> for updated fees

Companies that hire Vskills Certified Angular 8 Developer

Angular is one of the trending JavaScript framework web developers are using to create web applications, there is a huge demand for certified professionals all across the globe. Companies like Google, Flipkart, TCS & Cognizant, IBM, Infosys, IBA etc are looking for certified professionals.

Table of Contents

1. Getting Started

- 1.1 Course Introduction
- 1.2 What is Angular?
- 1.3 Angular vs Angular 2 vs Angular 8
- 1.4 Project Setup and First App
- 1.5 Editing the First App
- 1.6 The Course Structure
- 1.7 How to get the Most out of the Course
- 1.8 What is TypeScript?
- 1.9 A Basic Project Setup using Bootstrap for Styling

2. The Basics

- 2.1 Module Introduction
- 2.2 How an Angular App gets Loaded and Started
- 2.3 Components are Important!
- 2.4 Creating a New Component
- 2.5 Understanding the Role of AppModule and Component Declaration
- 2.6 Using Custom Components
- 2.7 Creating Components with the CLI & Nesting Components
- 2.8 Working with Component Templates
- 2.9 Working with Component Styles
- 2.10 Fully Understanding the Component Selector
- 2.11 Assignment Solution
- 2.12 What is Databinding?
- 2.13 String Interpolation
- 2.14 Property Binding
- 2.15 Property Binding vs String Interpolation
- 2.16 Event Binding
- 2.17 Passing and Using Data with Event Binding
- 2.18 Two-Way-Databinding
- 2.19 Combining all Forms of Databinding
- 2.20 Assignment Solution
- 2.21 Understanding Directives
- 2.22 Using ngIf to Output Data Conditionally
- 2.23 Enhancing ngIf with an Else Condition
- 2.24 Styling Elements Dynamically with ngStyle
- 2.25 Applying CSS Classes Dynamically with ngClass
- 2.26 Outputting Lists with ngFor
- 2.27 Assignment Solution
- 2.28 Getting the Index when using ngFor

3. Course Project - The Basics

- 3.1 Project Introduction
- 3.2 Planning the App
- 3.3 Setting up the Application
- 3.4 Creating the Components
- 3.5 Using the Components
- 3.6 Adding a Navigation Bar
- 3.7 Creating a "Recipe" Model
- 3.8 Adding Content to the Recipes Components
- 3.9 Outputting a List of Recipes with ngFor
- 3.10 Displaying Recipe Details
- 3.11 Working on the ShoppingListComponent
- 3.12 Creating an "Ingredient" Model
- 3.13 Creating and Outputting the Shopping List
- 3.14 Adding a Shopping List Edit Section
- 3.15 Wrap Up & Next Steps

4. Debugging

- 4.1 Understanding Angular Error Messages
- 4.2 Debugging Code in the Browser Using Sourcemaps
- 4.3 Using Augury to Dive into Angular Apps

5. Components & Databinding Deep Dive

- 5.1 Module Introduction
- 5.2 Splitting Apps into Components
- 5.3 Property & Event Binding Overview
- 5.4 Binding to Custom Properties
- 5.5 Assigning an Alias to Custom Properties
- 5.6 Binding to Custom Events
- 5.7 Assigning an Alias to Custom Events
- 5.8 Custom Property and Event Binding Summary
- 5.9 Understanding View Encapsulation
- 5.10 More on View Encapsulation
- 5.11 Using Local References in Templates
- 5.12 Getting Access to the Template & DOM with @ViewChild
- 5.13 Projecting Content into Components with ng-content
- 5.14 Understanding the Component Lifecycle
- 5.15 Seeing Lifecycle Hooks in Action
- 5.16 Lifecycle Hooks and Template Access
- 5.17 Getting Access to ng-content with @ContentChild
- 5.18 Wrap Up
- 5.19 Assignment Solution

6. Course Project - Components & Databinding

- 6.1 Introduction
- 6.2 Adding Navigation with Event Binding and ngIf
- 6.3 Passing Recipe Data with Property Binding
- 6.4 Passing Data with Event and Property Binding (Combined)
- 6.5 Allowing the User to Add Ingredients to the Shopping List

7. Directives Deep Dive

- 7.1 Module Introduction
- 7.2 ngFor and ngIf Recap
- 7.3 ngClass and ngStyle Recap
- 7.4 Creating a Basic Attribute Directive
- 7.5 Using the Renderer to build a Better Attribute Directive
- 7.6 Using HostListener to Listen to Host Events
- 7.7 Using HostBinding to Bind to Host Properties
- 7.8 Binding to Directive Properties
- 7.9 What Happens behind the Scenes on Structural Directives?
- 7.10 Building a Structural Directive
- 7.11 Understanding ngSwitch

8. Course Project - Directives

- 8.1 Building and Using a Dropdown Directive

9. Using Services & Dependency Injection

- 9.1 Module Introduction
- 9.2 Why would you Need Services?
- 9.3 Creating a Logging Service
- 9.4 Injecting the Logging Service into Components
- 9.5 Creating a Data Service
- 9.6 Understanding the Hierarchical Injector
- 9.7 How many Instances of Service Should It Be?
- 9.8 Injecting Services into Services
- 9.9 Using Services for Cross-Component Communication
- 9.10 Assignment Solution

10. Course Project - Services & Dependency Injection

- 10.1 Introduction
- 10.2 Setting up the Services
- 10.3 Managing Recipes in a Recipe Service
- 10.4 Using a Service for Cross-Component Communication
- 10.5 Adding the Shopping List Service
- 10.6 Using Services for Pushing Data from A to B
- 10.7 Adding Ingredients to Recipes

10.8 Passing Ingredients from Recipes to the Shopping List (via a Service)

11. Changing Pages with Routing

- 11.1 Module Introduction
- 11.2 Why do we need a Router?
- 11.3 Setting up and Loading Routes
- 11.4 Navigating with Router Links
- 11.5 Understanding Navigation Paths
- 11.6 Styling Active Router Links
- 11.7 Navigating Programmatically
- 11.8 Using Relative Paths in Programmatic Navigation
- 11.9 Passing Parameters to Routes
- 11.10 Fetching Route Parameters
- 11.11 Fetching Route Parameters Reactively
- 11.12 An Important Note about Route Observables
- 11.13 Passing Query Parameters and Fragments
- 11.14 Retrieving Query Parameters and Fragments
- 11.15 Practicing and some Common Gotchas
- 11.16 Setting up Child (Nested) Routes
- 11.17 Using Query Parameters - Practice
- 11.18 Configuring the Handling of Query Parameters
- 11.19 Redirecting and Wildcard Routes
- 11.20 Outsourcing the Route Configuration
- 11.21 An Introduction to Guards
- 11.22 Protecting Routes with canActivate
- 11.23 Protecting Child (Nested) Routes with canActivateChild
- 11.24 Using a Fake Auth Service
- 11.25 Controlling Navigation with canDeactivate
- 11.26 Passing Static Data to a Route
- 11.27 Resolving Dynamic Data with the resolve Guard
- 11.28 Understanding Location Strategies
- 11.29 Wrap Up

12. Course Project - Routing

- 12.1 Planning the General Structure
- 12.2 Setting up Routes
- 12.3 Adding Navigation to the App
- 12.4 Marking Active Routes
- 12.5 Fixing Page Reload Issues
- 12.6 Child Routes: Challenge
- 12.7 Adding Child Routing Together
- 12.8 Configuring Route Parameters
- 12.9 Passing Dynamic Parameters to Links
- 12.10 Styling Active Recipe Items
- 12.11 Adding Editing Routes

- 12.12 Retrieving Route Parameters
- 12.13 Programmatic Navigation to the Edit Page
- 12.14 One Note about Route Observables

13. Understanding Observables

- 13.1 Module Introduction
- 13.2 Analyzing Angular Observables
- 13.3 Getting Closer to the Core of Observables
- 13.4 Building a Custom Observable
- 13.5 Errors & Completion
- 13.6 Observables & You!
- 13.7 Understanding Operators
- 13.8 Subjects
- 13.9 Wrap Up

14. Course Project - Observables

- 14.1 Improving the Reactive Service with Observables (Subjects)

15. Handling Forms in Angular Apps

- 15.1 Module Introduction
- 15.2 Why do we Need Angular's Help?
- 15.3 Template-Driven (TD) vs Reactive Approach
- 15.4 An Example Form
- 15.5 TD: Creating the Form and Registering the Controls
- 15.6 TD: Submitting and Using the Form
- 15.7 TD: Understanding Form State
- 15.8 TD: Accessing the Form with @ViewChild
- 15.9 TD: Adding Validation to check User Input
- 15.10 TD: Using the Form State
- 15.11 TD: Outputting Validation Error Messages
- 15.12 TD: Set Default Values with ngModel Property Binding
- 15.13 TD: Using ngModel with Two-Way-Binding
- 15.14 TD: Grouping Form Controls
- 15.15 TD: Handling Radio Buttons
- 15.16 TD: Setting and Patching Form Values
- 15.17 TD: Using Form Data
- 15.18 TD: Resetting Forms
- 15.19 Introduction to the Reactive Approach
- 15.20 Reactive: Setup
- 15.21 Reactive: Creating a Form in Code
- 15.22 Reactive: Syncing HTML and Form
- 15.23 Reactive: Submitting the Form
- 15.24 Reactive: Adding Validation
- 15.25 Reactive: Getting Access to Controls
- 15.26 Reactive: Grouping Controls

- 15.27 Reactive: Arrays of Form Controls (FormArray)
- 15.28 Reactive: Creating Custom Validators
- 15.29 Reactive: Using Error Codes
- 15.30 Reactive: Creating a Custom Async Validator
- 15.31 Reactive: Reacting to Status or Value Changes
- 15.32 Reactive: Setting and Patching Values
- 15.33 Assignment Solution

16. Course Project - Forms

- 16.1 Introduction
- 16.2 TD: Adding the Shopping List Form
- 16.3 Adding Validation to the Form
- 16.4 Allowing the Selection of Items in the List
- 16.5 Loading the Shopping List Items into the Form
- 16.6 Updating existing Items
- 16.7 Resetting the Form
- 16.8 Allowing the the User to Clear (Cancel) the Form
- 16.9 Allowing the Deletion of Shopping List Items
- 16.10 Creating the Template for the (Reactive) Recipe Edit Form
- 16.11 Creating the Form for Editing Recipes
- 16.12 Syncing HTML with the Form
- 16.13 Adding Ingredient Controls to a Form Array
- 16.14 Adding new Ingredient Controls
- 16.15 Validating User Input
- 16.16 Submitting the Recipe Edit Form
- 16.17 Adding a Delete and Clear (Cancel) Functionality
- 16.18 Redirecting the User (after Deleting a Recipe)
- 16.19 Adding an Image Preview
- 16.20 Providing the Recipe Service Correctly
- 16.21 Deleting Ingredients and Some Finishing Touches

17. Using Pipes to Transform Output

- 17.1 Introduction & Why Pipes are Useful
- 17.2 Using Pipes
- 17.3 Parametrizing Pipes
- 17.4 Where to learn more about Pipes
- 17.5 Chaining Multiple Pipes
- 17.6 Creating a Custom Pipe
- 17.7 Parametrizing a Custom Pipe
- 17.8 Example: Creating a Filter Pipe
- 17.9 Pure and Impure Pipes (or: How to "fix" the Filter Pipe)
- 17.10 Understanding the "async" Pipe

18. Making Http Requests

- 18.1 A New IDE
- 18.2 Module Introduction
- 18.3 How Does Angular Interact with Backends?
- 18.4 The Anatomy of a Http Request
- 18.5 Backend (Firebase) Setup
- 18.6 Sending a POST Request
- 18.7 Getting Data
- 18.8 Using RxJS Operators to Transform Response Data
- 18.9 Using Types with the HttpClient
- 18.10 Outputting Posts
- 18.11 Showing a Loading Indicator
- 18.12 Using a Service for Http Requests
- 18.13 Services & Components Working Together
- 18.14 Sending a DELETE Request
- 18.15 Handling Errors
- 18.16 Using Subjects for Error Handling
- 18.17 Using the catchError Operator
- 18.18 Error Handling & UX
- 18.19 Setting Headers
- 18.20 Adding Query Params
- 18.21 Observing Different Types of Responses
- 18.22 Changing the Response Body Type
- 18.23 Introducing Interceptors
- 18.24 Manipulating Request Objects
- 18.25 Response Interceptors
- 18.26 Multiple Interceptors
- 18.27 Wrap Up

19. Course Project - Http

- 19.1 Module Introduction
- 19.2 Backend (Firebase) Setup
- 19.3 Setting Up the DataStorage Service
- 19.4 Storing Recipes
- 19.5 Fetching Recipes
- 19.6 Transforming Response Data
- 19.7 Resolving Data Before Loading
- 19.8 Fixing a Bug with the Resolver

20. Authentication & Route Protection in Angular Apps

- 20.1 Module Introduction
- 20.2 How Authentication Works
- 20.3 Adding the Auth Page
- 20.4 Switching Between Auth Modes
- 20.5 Handling Form Input

- 20.6 Preparing the Backend
- 20.7 Preparing the Signup Request
- 20.8 Sending the Signup Request
- 20.9 Adding a Loading Spinner & Error Handling Logic
- 20.10 Improving Error Handling
- 20.11 Sending Login Requests
- 20.12 Login Error Handling
- 20.13 Creating & Storing the User Data
- 20.14 Reflecting the Auth State in the UI
- 20.15 Adding the Token to Outgoing Requests
- 20.16 Attaching the Token with an Interceptor
- 20.17 Adding Logout
- 20.18 Adding Auto-Login
- 20.19 Adding Auto-Logout
- 20.20 Adding an Auth Guard
- 20.21 Wrap Up

21. Dynamic Components

- 21.1 Module Introduction
- 21.2 Adding an Alert Modal Component
- 21.3 Understanding the Different Approaches
- 21.4 Using ngIf
- 21.5 Preparing Programmatic Creation
- 21.6 Creating a Component Programmatically
- 21.7 Understanding entryComponents
- 21.8 Data Binding & Event Binding
- 21.9 Wrap Up

22. Angular Modules & Optimizing Angular Apps

- 22.1 Module Introduction
- 22.2 What are Modules?
- 22.3 Analyzing the AppModule
- 22.4 Getting Started with Feature Modules
- 22.5 Splitting Modules Correctly
- 22.6 Adding Routes to Feature Modules
- 22.7 Component Declarations
- 22.8 The ShoppingList Feature Module
- 22.9 Understanding Shared Modules
- 22.10 Understanding the Core Module
- 22.11 Adding an Auth Feature Module
- 22.12 Understanding Lazy Loading
- 22.13 Implementing Lazy Loading
- 22.14 More Lazy Loading
- 22.15 Preloading Lazy-Loaded Code
- 22.16 Modules & Services

- 22.17 Loading Services Differently
- 22.18 Ahead-of-Time Compilation
- 22.19 Wrap Up

23. Deploying an Angular App

- 23.1 Module Introduction
- 23.2 Deployment Preparations and Important Steps
- 23.3 Using Environment Variables
- 23.4 Deployment Example: Firebase Hosting

24. Bonus: Working with NgRx in our Project

- 24.1 Module Introduction
- 24.2 What is Application State?
- 24.3 What is NgRx?
- 24.4 Getting Started with Reducers
- 24.5 Adding Logic to the Reducer
- 24.6 Understanding & Adding Actions
- 24.7 Setting Up the NgRx Store
- 24.8 Selecting State
- 24.9 Dispatching Actions
- 24.10 Multiple Actions
- 24.11 Preparing Update & Delete Actions
- 24.12 Updating & Deleting Ingredients
- 24.13 Expanding the State
- 24.14 Managing More State via NgRx
- 24.15 Removing Redundant Component State Management
- 24.16 First Summary & Clean Up
- 24.17 One Root State
- 24.18 Setting Up Auth Reducer & Actions
- 24.19 Dispatching Auth Actions
- 24.20 Auth Finished (For Now...)
- 24.21 And Important Note on Actions
- 24.22 Exploring NgRx Effects
- 24.23 Defining the First Effect
- 24.24 Effects & Error Handling
- 24.25 Login via NgRx Effects
- 24.26 Managing UI State in NgRx
- 24.27 Finishing the Login Effect
- 24.28 Preparing Other Auth Actions
- 24.29 Adding Signup
- 24.30 Further Auth Effects
- 24.31 Adding Auto-Login with NgRx
- 24.32 Adding Auto-Logout
- 24.33 Finishing the Auth Effects
- 24.34 Using the Store Devtools

- 24.35 The Router Store
- 24.36 Getting Started with NgRx for Recipes
- 24.37 Fetching Recipe Detail Data
- 24.38 Fetching Recipes & Using the Resolver
- 24.39 Fixing the Auth Redirect
- 24.40 Update, Delete and Add Recipes
- 24.41 Storing Recipes via Effects
- 24.42 Cleanup Work
- 24.43 Wrap Up

25. Bonus: Angular Universal

- 25.1 Module Introduction
- 25.2 Getting Started with Angular Universal
- 25.3 Working on the App Module
- 25.4 Adding a Server-Side Build Workflow
- 25.5 Adding a NodeJS Server
- 25.6 Pre-Rendering the App on the Server
- 25.7 Next Steps

26. Angular Animations

- 26.1 Introduction
- 26.2 Setting up the Starting Project
- 26.3 Animations Triggers and State
- 26.4 Switching between States
- 26.5 Transitions
- 26.6 Advanced Transitions
- 26.7 Transition Phases
- 26.8 The "void" State
- 26.9 Using Keyframes for Animations
- 26.10 Grouping Transitions
- 26.11 Using Animation Callbacks

27. Adding Offline Capabilities with Service Workers

- 27.1 Module Introduction
- 27.2 Adding Service Workers
- 27.3 Caching Assets for Offline Use
- 27.4 Caching Dynamic Assets & URLs

28. A Basic Introduction to Unit Testing in Angular Apps

- 28.1 Introduction
- 28.2 Why Unit Tests?
- 28.3 Analysing the Testing Setup (as created by the CLI)
- 28.4 Running Tests (with the CLI)
- 28.5 Adding a Component and some fitting Tests
- 28.6 Testing Dependencies: Components and Services
- 28.7 Simulating Async Tasks
- 28.8 Using "fakeAsync" and "tick"
- 28.9 Isolated vs Non-Isolated Tests

29. Angular Changes & New Features

- 29.1 What's New with Angular 8?
- 29.2 A First Look At Angular Elements

Sample Questions

1. What is sent by server after approval of the client's request?
 - A. 200 OK
 - B. 400 OK
 - C. 600 OK
 - D. 800 OK

2. Which Angular component's lifecycle hook for the components children, is invoked when the component's view has been fully initialized?
 - A. ngAfterContentInit
 - B. ngAfterContentChecked
 - C. ngAfterViewInit
 - D. ngAfterViewChecked

3. What is the data type of first parameter of ngInclude directive of Angular?
 - A. string
 - B. object
 - C. list
 - D. character

4. What is not a benefit of dependency injection in Angular?
 - A. easier testability
 - B. better code organization
 - C. better code modularization
 - D. None of these

5. Which type of forms places most of the form handling logic within that form's template?
 - A. Template-Driven Forms
 - B. Reactive Forms
 - C. Display Forms
 - D. Permanent Forms

Answers: 1 (A), 2 (C), 3 (A), 4 (D), 5 (A)

Certifications

- ▶ **Accounting, Banking & Finance**
 - Certified GST Professional
 - Certified AML-KYC Compliance Officer
 - Certified Business Accountant
 - Certified BASEL III Professional
 - Certified GAAP Accounting Standards Professional
 - Certified Treasury Markets Professional
- ▶ **Big Data**
 - Certified Hadoop and Mapreduce Professional
- ▶ **Cloud Computing**
 - Certified Cloud Computing Professional
- ▶ **Design**
 - Certified Interior Designer
- ▶ **Digital Media**
 - Certified Social Media Marketing Professional
 - Certified Inbound Marketing Professional
 - Certified Digital Marketing Professional
- ▶ **Foreign Trade**
 - Certified Export Import (Foreign Trade) Professional
- ▶ **Health, Nutrition and Well Being**
 - Certified Fitness Instructor
- ▶ **Hospitality**
 - Certified Restaurant Team Member (Hospitality)
- ▶ **Human Resources**
 - Certified HR Compensation Manager
 - Certified HR Staffing Manager
 - Certified Human Resources Manager
 - Certified Performance Appraisal Manager
- ▶ **Office Skills**
 - Certified Data Entry Operator
 - Certified Office Administrator
- ▶ **Project Management**
 - Certified Master in Project Management
 - Certified Scrum Specialist
- ▶ **Real Estate**
 - Certified Real Estate Consultant
- ▶ **Marketing**
 - Certified Marketing Manager
- ▶ **Quality**
 - Certified Six Sigma Green Belt Professional
 - Certified Six Sigma Black Belt Professional
 - Certified TQM Professional
- ▶ **Logistics & Supply Chain Management**
 - Certified International Logistics Professional
 - Certified Logistics & SCM Professional
 - Certified Supply Chain Management Professional
- ▶ **Legal**
 - Certified IPR & Legal Manager
 - Certified Labour Law Analyst
 - Certified Business Law Analyst
 - Certified Corporate Law Analyst
- ▶ **Information Technology**
 - Certified Angular JS Professional
 - Certified Basic Network Support Professional
 - Certified Business Intelligence Professional
 - Certified Core Java Developer
 - Certified E-commerce Professional
 - Certified IT Support Professional
 - Certified PHP Professional
 - Certified Selenium Professional
- ▶ **Mobile Application Development**
 - Certified Android Apps Developer
 - Certified iPhone Apps Developer
- ▶ **Security**
 - Certified Ethical Hacking and Security Professional
 - Certified Network Security Professional
- ▶ **Management**
 - Certified Corporate Governance Professional
 - Certified Corporate Social Responsibility Professional
 - Certified Leadership Skills Professional
- ▶ **Life Skills**
 - Certified Business Communication Specialist
 - Certified Public Relations Officer
- ▶ **Media**
 - Certified Advertising Manager
 - Certified Advertising Sales Professional
- ▶ **Sales, BPO**
 - Certified Sales Manager
 - Certified Telesales Executive

& many more job related certifications

Contact us at:
V-Skills
011-473 44 723 or info@vskills.in
www.vskills.in