

Session 1

Topic 1 : Creating Our First Component in Angular

About Angular

Web components

Why Only TypeScript?

Angular CLI importance and Installation

First app

Testing

Angular Hello World

Creating the component

Selector

Template/templateUrl

Telling the module

Setting up a bootstrap file

Diving deeper into Angular components

Component methods and data updates

Going from static to actual data

Adding interactivity to the component

Improving the data output

IDEs and Plugins

IDEs

Atom

Sublime Text

Webstorm

Visual Studio Code

Session 2

Topic 2 : Introducing TypeScript

Understanding the case for TypeScript

The benefits of TypeScript

Introducing TypeScript resources in the wild

The TypeScript official site, The TypeScript official wiki

Types in TypeScript

String, The let keyword, Const, Number, Boolean, Array, Dynamic typing with the any type, Custom type, Enum, Void, Type inference

Functions, lambdas, and execution flow

- Annotating types in our functions
- Function parameters in Typescript
 - Optional parameters, Default parameters, Rest parameters,
 - Overloading the function signature

Better function syntax with and scope handling with lambdas

General features

- Spread parameter
- Template strings
- Generics

Classes, interfaces, and class inheritance

- Anatomy of a class – constructors, properties, methods, getters, and setters
- Constructor parameters with accessors
- Interfaces in TypeScript
- Extending classes with class inheritance

Decorators in TypeScript

- Class decorators
 - Extending the class decorator function signature
- Property decorators
- Method decorators
- Parameter decorator

Organizing our applications with modules

- Internal modules
- External modules
 - ES like modules per TypeScript > 1.5

Session 3

Topic 3 : Implementing Properties and Events in Our Components

A better template syntax

Data bindings with input properties

Some extra syntactic sugar when binding expressions

Event binding with output properties

Input and output properties in action

- Setting up custom values declaratively
- Communicating between components through custom events

Emitting data through custom events

Local references in templates

Alternative syntax for input and output properties

Configuring our template from our component class

- Internal and external templates
- Encapsulating CSS styling
 - The styles property
 - The styleUrls property
 - Inline style sheets
- Managing view encapsulation

Session 4:

Topic : Enhancing Our Components with Pipes and Directives

Directives in Angular

- Core directives
 - NgIf
 - NgFor
 - Advanced looping
 - Else
 - Applying style
 - Interpolation
 - NgStyle
 - NgClass
 - NgSwitch, ngSwitchCase, and ngSwitchDefault

Manipulating template bindings with pipes

- The uppercase/lowercase pipe
- The decimal, percent, and currency pipes
 - The decimal pipe
 - The percent pipe
 - The currency pipe
- The slice pipe
- The date pipe
- The JSON pipe
- The i18n pipes
 - The i18nPlural pipe
 - The i18nSelect pipe
- The async pipe

Putting it all together in a task list

- Setting up our main HTML container
- Building our task list table with Angular directives
- Toggling tasks in our task list

- Displaying state changes in our templates
- Embedding child components

Building our own custom pipes

- Anatomy of a custom pipe
 - Imports
 - Defining our pipe
 - Registering it
 - The pure property
- A custom pipe to better format time output
- Filtering out data with custom filters

Building our own custom directives

- Anatomy of a custom directive
 - Listening to events
 - Adding input data
 - Adding more than one input property
- A second example – error validation
- Building a task tooltip custom directive

A word about naming conventions for custom directives and pipes

Session 5

Topic : Building an Application with Angular Components

Introducing the component tree

Common conventions for scalable applications

- File and ES module naming conventions
- From facades/barrels to NgModule
- Using NgModule

How dependency injection works in Angular

- A note on providers

Injecting dependencies across the component tree

- Restricting dependency injection down the component tree
- Restricting provider lookup

Overriding providers in the injector hierarchy

Extending injector support to custom entities

Initializing applications with bootstrapModule()

- Switching between development and production modes
- Different modes in Angular CLI

Introducing the app directory structure

Refactoring our application the Angular way

- The shared context or store it all in a common module
- Services in the shared context

Configuring application settings from a central service

Putting it all together in our shared module

Creating our components

Introduction to life cycle hooks

OnInit - the beginning of it all

OnDestroy - called when a component is removed from a DOM tree

OnChanges - a change has occurred

The timer feature

The tasks feature

Defining the top root component

Session 6

Topic : Asynchronous Data Services with Angular

Strategies for handling asynchronous information

Asynchronous response - from callbacks to promises

Observables in a nutshell

Reactive functional programming in Angular

The RxJS library

Introducing the HTTP API

Working with headers

Handling errors when performing HTTP requests

Injecting the HttpClient service

A real case study – serving Observable data through HTTP

Leveraging HTTP – refactoring our TaskService to use HTTP service

Using the Angular HTTP service

A stateful TaskService for most scenarios

Further improvements – turning TaskService into a stateful, more robust service

Handling service calls when network connection intermittently goes offline

Introducing the store/feed pattern

Persisting data

Refreshing our service

HttpInterceptor

Creating a mock interceptor

Some best practices

Async operator

Being a good citizen – cleaning up after yourself

Session 7

Topic : Firebase

Three-way binding versus two-way binding

Some words on storage - the problem with a list

Objects in a list - solving the deletion problem

AngularFire

The core classes

The admin tools

Defining your database

Adding AngularFire to your app

Protecting our app

Authentication - allowing access to the app

Authorization - deciding who gets to access what data, and how

Validation

Working with data - CRUD

Reading data

Changing data

Removing data

Removing an object

Removing an item in a list

Responding to change

Adding authentication

Simple authentication with email/password

Session 8

Topic : Routing

Adding support for the Angular router

Specifying the base element

Importing and setting up the router module

Defining the routes

Defining a viewport

Building a practical example – setting up the router service

Building a new component for demonstration purposes

Cleaning up the routes

The router directives – RouterOutlet, RouterLink, and RouterLinkActive

Triggering routes imperatively

Handling parameters

- Building detail pages – using route parameters
- Filtering your data – using query parameters

Advanced features

- Child routes
- Absolute versus relative navigation
- Named outlets
- Debugging
- Fine-tuning our generated URLs with location strategies

Securing the routes with AuthGuard and CanActivate hook

Resolve<T> – fetching and resolving data before routing

The CanDeactivate – handling cancel and save

Async routing – improving response time

- Lazy loading
 - CanLoad – don't lazy load unless the user has access
- Preloading

Session 9

Topic : Forms in Angular

Template-driven forms

- Turning a simple form into a template-driven form
 - Input field error – from a CSS standpoint
 - Detecting an error on an input field with named references
- Improving the form
 - Showing errors at the right time

Reactive forms

- AbstractControl
- Programmatic and dynamic approach
- Turning a form into a dynamic form
- Adding controls with validation rules
- Refactoring – making the code more dynamic
- Updating our component form model – using setValue and patchValue
 - setValue
 - patchValue
- Cleaning up our form creation and introducing FormBuilder
- Building your own custom validator
- Watching state changes and being reactive

Session 10

Topic : Angular Material

Material Design

- Known implementations

Angular Material

- Installation

 - Our first Angular Material app

Component overview

- Buttons

- Form controls

 - Input

 - Autocomplete

 - Checkbox

 - Date picker

- Navigation

 - Menu

- Layout

 - List

 - Grid list

- Popups and modals

 - Dialog

 - A more advanced example – sending data to and from your dialog

- Data table

 - Table

 - Sorting

 - Pagination

Session 11

Topic : Animating Components with Angular

- Creating animations with plain vanilla CSS

- Introducing Angular animations

 - Our first trigger

 - Connecting the parts

The wildcard state
The void state
Animation callbacks
Animating components with the AnimationBuilder
Creating a reusable animation directive

Session 12

Topic : Unit Testing in Angular

Why do we need tests?

The anatomy of a unit test

Introduction to testing in Angular

Configuration and setting up

Angular testing utilities

Introduction to component testing

Component testing with dependencies

Using a stub to replace the dependency

Spying on the dependency method

Async services

Testing pipes

Mocking HTTP responses with HttpClientTestingModule

Input and outputs

Testing routing

Testing navigation

Testing routes by URL

Testing route parameters

Testing directives

The road ahead

Introducing code coverage reports in your test stack

Implementing E2E tests