

Angular Seminars

The following is a list of Angular seminars that can be presented at your user group or other live event by **Paul D. Sheriff**.

Essentials of Angular

Are you new to Angular? Want to learn how to build your first Angular application using Angular CLI and Visual Studio Code? This seminar walks you through the process step-by-step. A simple product list page will be built starting with HTML and gradually adding more functionality until everything works in Angular. You will learn the basics of data binding, how to apply pipes to transform data, and working with classes.

Learning Objectives

Best practices for starting an Angular application

Learn to create an Angular component

Build a controller from which to interact with your HTML view

Secure Your Angular Application using JSON Web Tokens

Most business applications require a user to login before they can use an application. In addition, menus and pages should only be available to certain users. This seminar shows you step-by-step how to secure menus using a security service and lock routes using an Angular Guard. You build a ASP.NET Core Web API and retrieve security data for a user. JSON Web Tokens are a simple and effective method of securing your Web API methods. You learn to integrate this tool into your Core Web API project and secure your methods.

Learning Objectives

Authenticate a user

Authorize users for different parts of the application

Secure routes with Guards

Get security data from SQL Server via ASP.NET Core Web API

Use JSON Web Tokens to secure Web API methods

Angular Routing Basics

To navigate from page to page in your Angular application, you will take advantage of routing. There are many things you can do with routing, including passing parameters, navigate using attributes and code, and checking data before allowing navigation. You also learn how to stop a user from navigating away from a page if data is not saved. The sample in this session is a list and a detail page that you learn to navigate between.

Learning Objectives

What is Angular routing and where to use it

Passing parameters

Navigating using code

Using Route Guards

Lazy Loading in Angular

In any large application, there are often areas of the program that user's do not go into much. Instead of loading the complete application when the user navigates to your Angular application, you can take advantage of lazy loading, so only those modules that are used most often are loaded immediately. Other modules are only loaded when a user navigates to them. In this seminar you learn how to break up your Angular application into modules that lend themselves to lazy loading techniques.

Learning Objectives

Break up your application into modules

Change your routing to support lazy loading

How to structure your application for lazy loading

Calling a Web API from Angular

This session will walk you step-by-step through building and calling a ASP.NET Core Web API project to get product data from the server and display on an Angular web page. We will start with a simple HTML page that uses mock data and add the appropriate Web API calls to receive data from a SQL Server table. You will also learn how to display a message while the data is retrieved. A generic exception handler is also shown.

Learning Objectives

Build Web API for use with Angular

Learn to call a Web API from your controller

Information message display

Generic exception handler

Handling Exceptions in Angular

Exceptions happen. This session shows you how to communicate from the Web API back to your Angular controller. You will learn how to use the various HTTP status codes and even throwing exceptions. On the client side you will write a generic exception handler to determine the type of error and display the appropriate error messages. You will also learn to handle model state validation errors as part of this seminar.

Learning Objectives

Write client-side exception handling

How to generate exceptions from the Web API

Upload Small Files

Sometimes you need to upload files to your server via an Angular application. There are a few different methods you may use to upload. In this seminar you learn a method that works well for small files, up to about one to two megabytes in size. In this seminar you watch as a page is built to allow you to select one or more small files using an `<input type="file">` element. The file selected is sent via a Web API call to a method on the server. Once the server has this file, you may choose to save it as a file on the server, or whatever you choose.

Learning Objectives

- Style the HTML file input control
- Create a thumbnail of an uploaded image
- Store the uploaded file on the server file system
- Store the uploaded file into an SQL Server table

Validating Data in Angular

There are so many options for validation when using Angular. This seminar explores a few different methods using client-side and server-side code. You learn to how to return validation errors from Web API calls, learn to use Angular client-side validation, and how to display validation to your user. You also learn to create custom validators.

Learning Objectives

- Server-side validation and message handling
- Client-side validation
- Building custom Angular validators

Message Handling in Angular

When working in an Angular application you need to display messages to the user. These can be informational messages to inform them something is loading. They can be exception messages received from a Web API call. They can also be validation messages. Developing a good message service will make working with messages easier in Angular.

Learning Objectives

Create a message service

Create custom selectors to display each type of messages

Avoid hard-coding messages

Create an Angular CRUD Page

CRUD pages are in almost every application. A CRUD page is created so a user can list, add, edit and delete data. Let's learn how to do this using Angular and ASP.NET Core Web API. In this seminar you see how to retrieve and modify data using Angular and Web API methods. Visual Studio Code and the latest version of Angular is used in this seminar.

Learning Objectives

Build a CRUD page

Add, edit and delete using Angular and Web API

Displaying messages to the user

Build a Reusable Angular Logging System

Professional programmers know that logging errors or being able to track what is going on within an Angular application is essential. But what is the best way to log errors, what should you log, and where should you log it to? As Angular is all about services, creating a logging service is the best way to start. You should log a message and log any additional data such as contents of variables

that make sense for a log entry. You should probably log to a couple of different locations to ensure that the entry is recorded. In this course, you build three classes to log to the console window, to local storage, and to a Web API.

Learning Objectives

Create an Angular Service to log messages and exceptions

Build a set of classes to log to console, local storage and Web API

Create a log publishing system

Configure which log classes to use

Build a Reusable Angular Configuration Management System

In the Angular application you are building, you would like to supply some default values, so your user does not have to fill in every field on your web pages. In addition, you want your user to be able to change these default values. This course shows you how to create a configuration management service allowing you to keep these default values in one place. This course guides the programmer on creating a reusable configuration management service to retrieve these values from different locations. You learn to retrieve settings from a JSON file, from local storage, and from a Web API call.

Learning Objectives

Create an Angular Service to read configuration settings

Read configuration settings from JSON file

Read and Store settings in local storage

Read settings from a Web API

Create an Address Entry System using Angular

Your boss has tasked you with creating an address entry system that can handle your growing international business. This means you need to be able to handle various address formats in a single screen. You decide to turn to Angular to help you develop this entry system. You are going to develop a set of database tables, Web API calls and a front-end system in Angular to help you meet the goals your boss has set for you.

Learning Objectives

- Build something “useful” using Angular
- Handle US and international addresses
- Data binding with Angular
- Calling a Web API from Angular
- Custom Validators

Create a Credit Card Entry System using Angular

Many websites need to ask a user to enter a credit card in order to purchase some goods. Building this page is not quite as simple as you may think. In fact you need a few APIs, some C# classes, and some business logic to ensure this page works correctly. In this seminar you are going to learn to build all the components necessary to create a page to accept credit card information using HTML, Bootstrap, Angular and the Web API.

Learning Objectives

- Build something “useful” using Angular
- Create a credit card entry system
- Data binding with Angular
- Calling a Web API from Angular