# Advanced MVC: Building Web Applications Using the ASP.NET Web API with C# and .NET6

**Duration:** 5 Days *(Face-to-Face & Remote-Live)*, or 35 Hours *(On-Demand)*

**Price:** $2495 *(Face-to-Face & Remote-Live)*, or $895 *(On-Demand)*

**Discounts:** We offer multiple discount options. Click here for more information.

**Delivery Options:** Attend face-to-face in the classroom, remote-live or via on-demand training.

## Description

This course provides students with the skills needed to create sophisticated web applications using advanced features of ASP.NET MVC, the Entity Framework, Web API, and the popular JavaScript libraries jQuery, jQuery UI, and Bootstrap. Students will build several ASP.NET MVC web applications using Visual Studio and .NET 6 during the week to reinforce the skills they learn.

The course begins with an overview of ASP.NET MVC for experienced programmers and then covers the fundamentals of responsive website design. Students then learn how to use Bootstrap with MVC to make a web site responsive on devices from large desktop displays to small mobile devices.

Students will learn how to integrate jQuery into an MVC application to provide dynamic client-side behavior. They will then learn how to use jQuery's Ajax features to build more interactive MVC applications. Students will also learn how to use jQuery UI widgets to provide a more sophisticated user interface for their MVC views.

The course covers use of the Entity Framework to provide a data access layer for an MVC application. The Database First strategy is briefly examined before thorough coverage of the more configurable Code First strategy. Students will also learn how to use the Web API to build APIs that expose services and data via HTTP. These APIs can be used by a diverse set of clients including browsers, desktop applications and mobile devices.

The course includes coverage of Single-Page Applications. In these types of applications, the entire page is loaded in the browser after the initial request. All other interactions with the server utilize Ajax requests to update the page. The service layer is implemented with

the Web API.

Additional topics include: using HTML helpers to design views, creating both inline and custom HTML helpers to encapsulate view markup, using convention-based and attribute routing to support custom routes, and use of the NuGet packages with MVC applications.

## Prerequisites

Prior experience building ASP.NET MVC web sites. Students who are not familiar with ASP.NET MVC or C# should take the ASP.NET MVC Programming Using C# and .NET6 course instead.

## Course Overview

### Quick Review of ASP.NET MVC

- What is the MVC Design Pattern?
- Controllers, Action Methods and Routes
- Using Razor for Views
- Working with Layouts
- Working with HTML Forms
- Untyped vs Typed Views
- HTML Helpers
- Data Annotations for Models
- Advanced Controller Action Methods

### Responsive Design using Bootstrap

- Installing Bootstrap in a Project
- Managing Layout with Bootstrap
- Bootstrap CSS Classes
- Navigation Bars & Menus
- Icons & Badges
- Tables, Forms & Images

### Using jQuery

- The jQuery Function
- jQuery Selectors
- Working with the Document Object Model
- Working with Forms
- Getting and Setting Dimensions

### Async Programming

- Legacy async with Ajax
- Ajax and the Callback Pattern
- JavaScript `fetch` API
- `fetch` and the Promise Pattern
- Using Partial Views with `fetch`
- `fetch` notifications

### Services and Middleware

- Service Lifetime
- Using a Service in an Action Method
- Investigating Built-In Middleware
- Middleware and Delegates
- Branching Logic with Middleware

### Routing

- Serving Non-Routed Requests for Static Files
- Creating Convention-Based Routes
- Creating Attribute Routes
- Route Constraints for Type Safety
- Custom Route Constraint Classes
- Using Route Names in HTML Helpers

### Entity Framework Database First

### Entity Framework Code First

- Entity Framework Basics
- Creating the Entity Data Model
- Using Stored Procedures
- Using LINQ to Work with Model Collections
- Controller Scaffolding with Models

- Creating Model Classes
- Seeding the New Database
- Changing Schema at Dev Time with Code First Migrations

## Working with Web API

- Designing for RESTful Services
- OpenAPI/Swagger for Documentation and Testing
- Using API Controllers
- Attributes for Model Binding
- Using HTTP GET, PUT, POST and DELETE Requests
- Consuming API Services in C# and JavaScript
- Customizing Data Response Types