

ASP.NET MVC 5

Prerequisites / Further Training

- [Introduction to Programming](#)
- [Beginner C#.NET](#) (C# Fundamentals and Object – Orientation)
- [Advanced C#.NET](#) (Building practical Apps)
- [SQL Querying](#) (Database design and querying)
- *ASP.NET MVC5 (Web App Development)*
- ASP.NET Web API (More Web App Development)
- Entity Framework

Also have a look at our [.NET Bootcamp](#) and save

Intended Audience

This course is useful for C# or ASP.NET Forms programmer who wants to learn the MVC (Model View Controller) methodology.

This course covers all the topics of the Microsoft Exam for ASP.NET MVC 5. After completing this course you should be able to prepare for the Microsoft exam and understand all the concepts to pass quickly, just by doing a few mock exams.

Further Training

You may consider doing the certification exam courses:

MCPD – Web Apps

Course Material

Included in the course price.

Course Contents

Day1:

A First MVC Application

- Adding the First Controller
- Understanding Routes
- Rendering Web Pages
- Creating and Rendering a View
- Adding Dynamic Output
- Creating a Simple Data-Entry Application
- Setting the Scene
- Designing a Data Model
- Linking Action Methods
- Building the Form
- Setting the Start URL
- Handling Forms
- Adding Validation
- Styling the Content

The MVC Pattern

- The History of MVC
- Understanding the MVC Pattern
- Understanding the Domain Model
- The ASP.NET Implementation of MVC
- Comparing MVC to Other Patterns
- Building Loosely Coupled Components
- Using Dependency Injection
- Using a Dependency Injection Container
- Getting Started with Automated Testing
- Understanding Unit Testing
- Understanding Integration Testing

Essential Language Features

- Using Automatically Implemented Properties
- Using Object and Collection Initializers
- Using Extension Methods
- Applying Extension Methods to an Interface
- Creating Filtering Extension Methods

- Using Lambda Expressions
- Using Automatic Type Inference
- Using Anonymous Types
- Performing Language Integrated Queries
- Understanding Deferred LINQ Queries
- Using Async Methods
- Applying the async and await Keywords

Working with Razor

- Preparing the Example Project
- Defining the Model
- Defining the Controller
- Creating the View
- Working with the Model Object
- Working with Layouts
- Creating the Layout
- Applying a Layout
- Using a View Start File
- Shared Layouts
- Using Razor Expressions
- Inserting Data Values
- Setting Attribute Values
- Using Conditional Statements
- Enumerating Arrays and Collections
- Dealing with Namespaces

Essential Tools for MVC

- Creating the Model Classes
- Adding the Controller
- Adding the View
- Using Ninject
- Understanding the Problem
- Adding Ninject to the Visual Studio Project
- Getting Started with Ninject
- Setting up MVC Dependency Injection
- Creating Chains of Dependency

- Specifying Property and Constructor Parameter Values
- Using Conditional Binding
- Setting the Object Scope
- Unit Testing with Visual Studio
- Creating the Unit Test Project
- Creating the Unit Tests
- Running the Unit Tests (and Failing)
- Implementing the Feature
- Testing and Fixing the Code
- Using Moq
- Understanding the Problem
- Adding Moq to the Visual Studio Project
- Adding a Mock Object to a Unit Test
- Creating a More Complex Mock Object

Day2:

A Real Application

- Creating the Visual Studio Solution and Projects
- Installing the Tool Packages
- Adding References Between Projects
- Setting Up the DI Container
- Starting the Domain Model
- Creating an Abstract Repository
- Making a Mock Repository
- Displaying a List of Products
- Adding a Controller
- Adding the Layout, View Start File and View
- Setting the Default Route
- Preparing a Database
- Creating the Database
- Defining the Database Schema
- Adding Data to the Database
- Creating the Entity Framework Context
- Creating the Product Repository
- Adding Pagination
- Displaying Page Links

- Improving the URLs
- Styling the Content
- Installing the Bootstrap Package
- Applying Bootstrap Styles to the Layout
- Creating a Partial View

Navigation

- Adding Navigation Controls
- Filtering the Product List
- Refining the URL Scheme
- Building a Category Navigation Menu
- Correcting the Page Count
- Building the Shopping Cart
- Defining the Cart Entity
- Adding the Add to Cart Buttons
- Implementing the Cart Controller
- Displaying the Contents of the Cart

The Shopping Cart

- Using Model Binding
- Creating a Custom Model Binder
- Completing the Cart
- Removing Items from the Cart
- Adding the Cart Summary
- Submitting Orders
- Extending the Domain Model
- Adding the Checkout Process
- Implementing the Order Processor
- Registering the Implementation
- Completing the Cart Controller
- Displaying Validation Errors
- Displaying a Summary Page

Mobile Interface

- Putting Mobile Web Development in Context
- Doing Nothing (Or As Little As Possible)

- Using Responsive Design
- Creating a Responsive Header
- Creating a Responsive Product List
- Creating Mobile Specific Content
- Creating a Mobile Layout
- Creating the Mobile Views

Administration Section

- Adding Catalog Management
- Creating a CRUD Controller
- Creating a New Layout
- Implementing the List View
- Editing Products
- Creating New Products
- Deleting Products

Day3:

Security & Finishing Touches

- Securing the Administration Controller
- Creating a Basic Security Policy
- Applying Authorization with Filters
- Creating the Authentication Provider
- Creating the Account Controller
- Creating the View
- Image Uploads
- Extending the Database
- Enhancing the Domain Model
- Creating the Upload User Interface Elements
- Saving Images to the Database
- Implementing the GetImage Action Method
- Displaying Product Images

Overview of MVC Projects

- Working with Visual Studio MVC Projects
- Creating the Project

- Understanding MVC Conventions
- Debugging MVC Applications
- Preparing the Example Project
- Launching the Visual Studio Debugger
- Causing the Visual Studio Debugger to Break
- Using Edit and Continue
- Using Browser Link

URL Routing

- Creating the Controllers
- Creating the View
- Setting the Start URL and Testing the Application
- Introducing URL Patterns
- Creating and Registering a Simple Route
- Using the Simple Route
- Defining Default Values
- Using Static URL Segments
- Defining Custom Segment Variables
- Using Custom Variables as Action Method Parameters
- Defining Optional URL Segments
- Defining Variable-Length Routes
- Prioritizing Controllers by Namespaces
- Constraining Routes
- Constraining a Route Using a Regular Expression
- Constraining a Route to a Set of Specific Values
- Constraining a Route Using HTTP Methods
- Using Type and Value Constraints
- Defining a Custom Constraint
- Using Attribute Routing
- Enabling and Applying Attribute Routing
- Creating Routes with Segment Variables
- Applying Route Constraints
- Using a Route Prefix

Advanced Routing Features

- Simplifying the Routes

- Adding the Optimization Package
- Updating the Unit Test Project
- Generating Outgoing URLs in Views
- Using the Routing System to Generate an Outgoing URL
- Targeting Other Controllers
- Passing Extra Values
- Specifying HTML Attributes
- Generating Fully Qualified URLs in Links
- Generating URLs (and Not Links)
- Generating Outgoing URLs in Action Methods
- Generating a URL from a Specific Route
- Customizing the Routing System
- Creating a Custom RouteBase Implementation
- Creating a Custom Route Handler
- Working with Areas
- Creating an Area
- Populating an Area
- Resolving the Ambiguous Controller Issue
- Creating Areas with Attributes
- Generating Links to Actions in Areas
- Routing Requests for Disk Files
- Configuring the Application Server
- Defining Routes for Disk Files
- Bypassing the Routing System
- URL Schema Best Practices
- Make Your URLs Clean and Human-Friendly
- GET and POST : Pick the Right One

Controllers and Actions

- Introducing the Controller
- Creating a Controller with IController
- Creating a Controller by Deriving from the Controller Class
- Receiving Request Data
- Getting Data from Context Objects
- Using Action Method Parameters

- Producing Output
- Understanding Action Results
- Returning HTML by Rendering a View
- Passing Data from an Action Method to a View
- Performing Redirections
- Returning Errors and HTTP Codes

Day4:

Filters

- Using Filters
- Introducing the Filter Types
- Applying Filters to Controllers and Action Methods
- Using Authorization Filters
- Applying the Custom Authorization Filter
- Using the Built-in Authorization Filter
- Using Authentication Filters
- Understanding the IAuthenticationFilter Interface
- Implementing the Authentication Check
- Combining Authentication and Authorization Filters
- Handling the Final Challenge Request
- Using Exception Filters
- Creating an Exception Filter
- Applying the Exception Filter
- Using a View to Respond to an Exception
- Using the Built-in Exception Filter
- Using Action Filters
- Implementing the OnActionExecuting Method
- Implementing the OnActionExecuted Method
- Using Result Filters
- Using the Built-in Action and Result Filter Class
- Using Other Filter Features
- Filtering Without Attributes
- Using Global Filters
 - Ordering Filter Execution
 - Overriding Filters

Controller Extensibility

- Creating a Custom Controller Factory
- Dealing with the Fallback Controller
- Instantiating Controller Classes
- Implementing the Other Interface Methods
- Registering a Custom Controller Factory
- Working with the Built-in Controller Factory
- Prioritizing Namespaces
- Customizing `DefaultControllerFactory` Controller Instantiation
- Creating a Custom Action Invoker
- Using the Built-in Action Invoker
- Using a Custom Action Name
- Using Action Method Selection
- Improving Performance with Specialized Controllers
- Using Sessionless Controllers
- Using Asynchronous Controllers

Views

- Creating a Custom View Engine
- Creating a Custom `IView`
- Creating an `IViewEngine` Implementation
- Registering a Custom View Engine
- Testing the View Engine
- Working with the Razor Engine
- Preparing the Example Project
- Understanding Razor View Rendering
- Configuring the View Search Locations
- Adding Dynamic Content to a Razor View
- Using Layout Sections
- Using Partial Views
- Using Child Actions

Helper Methods

- Creating Custom Helper Methods

- Creating an Inline Helper Method
- Creating an External Helper Method
- Managing String Encoding in a Helper Method
- Using the Built-In Form Helper Methods
- Creating Form Elements
- Specifying the Route Used by a Form
- Using Input Helpers
- Creating Select Elements

Templated Helper Methods

- Using Templated Helper Methods
- Generating Label and Display Elements
- Using Whole-Model Templated Helpers
- Using Model Metadata
- Using Metadata to Control Editing and Visibility
- Using Metadata for Labels
- Using Metadata for Data Values
- Using Metadata to Select a Display Template
- Applying Metadata to a Buddy Class
- Working with Complex Type Properties
- Customizing the Templated View Helper System
- Creating a Custom Editor Template
- Creating a Generic Template
- Replacing the Built-in Templates

Day5:

URL and Ajax Helper Methods

- Creating Basic Links and URLs
- Using MVC Unobtrusive Ajax
- Creating the Synchronous Form View
- Preparing the Project for Unobtrusive Ajax
- Creating an Unobtrusive Ajax Form
- Preparing the Controller
- Creating the Ajax Form
- Understanding How Unobtrusive Ajax Works

- Setting Ajax Options
- Ensuring Graceful Degradation
- Providing the User with Feedback While Making an Ajax Request
- Prompting the User Before Making a Request
- Creating Ajax Links
- Ensuring Graceful Degradation for Links
- Working with Ajax Callbacks
- Working with JSON
- Adding JSON Support to the Controller
- Processing JSON in the Browser
- Preparing Data for Encoding
- Detecting Ajax Requests in the Action Method

Model Binding

- Understanding Model Binding
- Using the Default Model Binder
- Binding to Simple Types
- Binding to Complex Types
- Binding to Arrays and Collections
- Manually Invoking Model Binding
- Dealing with Binding Errors
- Customizing the Model Binding System
- Creating a Custom Value Provider
- Creating a Custom Model Binder
- Registering the Custom Model Binder

Model Validation

- Explicitly Validating a Model
- Displaying Validation Errors to the User
- Displaying Validation Messages
- Displaying Property-Level Validation Messages
- Using Alternative Validation Techniques
- Performing Validation in the Model Binder
- Specifying Validation Rules Using Metadata
- Defining Self-Validating Models

- Performing Client-Side Validation
- Enabling Client-Side Validation
- Using Client-Side Validation
- Understanding How Client-Side Validation Works
- Performing Remote Validation

Bundles

- Profiling Script and Style Sheet Loading
- Using Script and Style Bundles
- Adding the NuGet Package
- Defining the Bundles
- Applying Bundles
- Optimizing the JavaScript and CSS Files

Web API and Single-page Applications

- Understanding Single-page Applications
 - Creating the Model
 - Adding the NuGet Packages
 - Adding the Controller
 - Adding the Layout and Views
 - Setting the Start Location and Testing the Example Application
 - Using Web API
 - Creating the Web API Controller
 - Testing the API Controller
 - Understanding How the API Controller Works
 - Understanding API Controller Action Selection
 - Mapping HTTP Methods to Action Methods
 - Using Knockout for Single-page Applications
 - Adding the JavaScript Libraries to the Layout
 - Implementing the Summary
 - Implementing the Create Feature
 - Completing the Application
 - Simplify the Home Controller
 - Manage Content Visibility
-

Duration and pricing

- *Full-time* over 5 days (R9995)
- *Part-time* over 4 weeks (2 nights per week, 3 hour sessions) (R11995)
- *Part-time* over 8 Saturdays, 3 hour sessions (R11995)
- Please note : For *part-time* courses we do not have a fixed schedule and you will be placed on a waiting list until we get a group of 4+ together. Please book with no dates on the bookings form. This will automatically put you on the waiting list. We will confirm with you as soon as we have a part-time group together.
 - [Distance-learning](#) over up to 3 months (R9995)
- International exams are not included in the course price.
 - Prices exclude Vat for Vat-registered companies

Certificate

1. Upon completion of this course we will issue you with attendance certificate to certify your attendance and / or completion of the prescribed minimum examples.
2. You may sit for our competency assessment test and on passing you will obtain our competency certificate.
3. Our competency assessment can be booked and taken by someone who has not attended the course at a cost of R2950 including the course material and guidance session.

Bookings

You can download the course registration form on our home page or by clicking [here](#)

Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions

Please [email us](#)

Schedule

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page.