



0208 942 5724

Email: [Info@aspecttraining.co.uk](mailto:Info@aspecttraining.co.uk)

Web: [www.aspecttraining.co.uk](http://www.aspecttraining.co.uk)

# ASP.NET MVC Fundamentals

**Duration: 3 Days**

## Overview:

The Model View Controller Framework in ASP.NET provides a new way to develop Web applications for the Microsoft .NET platform. Differing completely from traditional ASP.NET development, ASP.NET MVC facilitates a refined code structure, total control over content generation and full support for test-driven development. On Aspect Training's ASP.NET MVC Fundamentals training course, you gain the skills required to effectively use ASP.NET MVC to build Web applications.

This course can be presented for both VB and C# developers.

## Prerequisites:

Aspect Training's ASP.NET MVC Fundamentals is designed for anyone involved in the development of Web applications using Microsoft technology. Programming experience in C# or Visual Basic is required to our Introduction Course level, and a fundamental knowledge of HTML are assumed.

## Course Topics:

### The MVC Framework in ASP.NET

#### Architecture of ASP.NET MVC

- Identifying the components of MVC
- Benefits of an MVC architecture
- Describing the MVC request processing cycle

#### Preparing the environment

- Installing ASP.NET MVC
- Structuring the project layout
- Integrating test-driven development

#### Enhancing functionality

- Intercepting request processing through action filtering
- Generating advanced operations by extending core controllers
- Decorating actions with additional behaviours
- Testing controllers outside of the server

### Orchestrating Application

#### Requests with Controllers

#### Defining controller architecture

- Differentiating controller types
- Delegating request processing to actions
- Initiating the response with action results
- Handling special cases using viewless controllers



0208 942 5724

Email: [Info@aspecttraining.co.uk](mailto:Info@aspecttraining.co.uk)

Web: [www.aspecttraining.co.uk](http://www.aspecttraining.co.uk)

## Mapping URLs for Request Routing

### Formatting the MVC URL

Shortcomings of ASP.NET URLs  
Decoupling URLs from actions with routing  
Designing a URL schema to enforce consistency

### Applying practical URL solutions

Configuring static and dynamic routes  
Customising routing for search engine optimisation  
Verifying routing with test-driven development

## Defining the Model

### Navigating the model landscape

Analysing the role of the model in different scenarios  
Constructing the N-tier architecture  
Improving application structure with dependency injection

### Persisting the domain model

Generating entity classes with Visual Studio Object Relational Designer  
Querying with LINQ to SQL  
Filtering data using properties

## Generating the View

### Rendering the response with MVC views

Implementing dynamic views  
Streamlining the view structure with HTML helpers  
Creating custom HTML helpers  
Incorporating existing ASP.NET views into MVC applications

### Collecting data with forms

Accessing submitted data  
Assigning validation rules to input fields  
Client-side vs. server-side data validation  
Reporting errors

### Displaying complex data

Representing datasets in a tabular format  
Building custom view components to handle diverse data  
Customising the view engine

## Creating Responsive Interfaces with Ajax

### The role of Ajax

Building Web 2.0 functionality  
Benefiting from built-in support for Ajax  
**Leveraging the jQuery JavaScript library**

Bringing pages to life with jQuery  
Handling browser events asynchronously for faster responses

Boost interactivity with special effects and animation

### Communicating efficiently with JSON

Serialising application data using JSON structures  
Developing JSON in controllers for Ajax  
Consuming JSON client-side in the view

## Securing and Deploying the Application

### Controlling access with authentication

Managing Windows-based authentication  
Gathering user credentials with HTML-based forms

### Restricting application access

Shielding sensitive URLs  
Integrating role-based security  
Accessing user and role information in controllers

### Packaging and distributing options

Enhancing application performance with cache configuration  
Publishing the completed application