

Essential ASP.NET Core

Duration: 5 Days Course Code: GK3368

Overview:

There are major changes in the next version of ASP.NET and MVC. ASP.NET Core is the new modular hosting framework for web applications written in .NET. ASP.NET Core MVC is the next version of Microsoft's server side web framework that unifies MVC and Web API. This release "is a reboot", as quoted from a project manager on the ASP.NET team and Essential ASP.NET Core and MVC provides the essential information on these new web frameworks.

This course is a deep exploration of ASP.NET Core and MVC and is intended for developers familiar with C#, .NET and web development. It is designed for both developers that are new to the ASP.NET Core and MVC platforms, as well as developers coming from prior versions of the ASP.NET platform.

Target Audience:

It is designed for both developers that are new to the ASP.NET Core and MVC platforms, as well as developers coming from prior versions of the ASP.NET platform.

Objectives:

- There are major changes in the next version of ASP.NET and MVC. ASP.NET Core is the new modular hosting framework for web applications written in .NET. ASP.NET Core MVC is the next version of Microsoft's server side web framework that unifies MVC and Web API. This release "is a reboot", as quoted from a project manager on the ASP.NET team and Essential ASP.NET Core and MVC provides the essential information on these new web frameworks.
 - This course is a deep exploration of ASP.NET Core and MVC and is intended for developers familiar with C#, .NET and web development. It is designed for both developers that are new to the ASP.NET Core and MVC platforms, as well as developers coming from prior versions of the ASP.NET platform.
 - What the .NET Core system is, how it loads your application, and it dynamically compiles your code with Roslyn.
 - How the ASP.NET Core web pipeline is constructed with middleware and how to build your own middleware.
 - The new flexible configuration system based upon XML files, JSON files, or environment variables.
 - How dependency injection permeates the entire ASP.NET Core and MVC frameworks
 - How the new MVC architecture is similar yet different than prior versions of MVC
 - How razor is used to build dynamic HTML pages and what features are new in MVC
 - What are the new security features in ASP.NET Core and MVC
 - How web APIs are designed and built in MVC
 - How the new project system utilizes modern build tools such as Node, Bower, Grunt, and Gulp
-

Content:

Day 1

1. .NET Core

.NET Core is the new cross-platform version of .NET. This module will explore .NET Core, the new SDK to develop with .NET Core, and how to use it to load an application, define and load dependencies, and package for deployment.

2. HTTP pipeline and middleware

ASP.NET Core defines a brand new way to build a HTTP server. This is done by allowing an application developer to define a series of components called middleware that will process HTTP requests and responses. This orchestration of middleware defines a custom HTTP pipeline in order to satisfy the requirements of an application. This module will introduce the ASP.NET pipeline, and show how to configure and define middleware.

3. Configuration and dependency injection

Given that ASP.NET Core is host independent and cross platform, a new configuration system was needed that allows for more flexibility than the prior web.config design. In this module we will see the new configuration system is designed, how to use it for XML, JSON, and/or environment variables, and how to configure it for various environments. In addition, and related to the configuration system, we will see how dependency injection is designed into ASP.NET Core, how to configure dependencies and their lifetime, and how to resolve dependencies.

Day 2

4. ASP.NET Core MVC

ASP.NET Core MVC is the web application framework built on top of ASP.NET Core. It follows the popular model-view-controller architecture. This version of MVC is completely rewritten from scratch on top of ASP.NET Core, but it tries to maintain as much familiarity as prior versions. In this module we will provide an overview of the MVC architecture including how a controller processes requests, communicates with the model layer and chooses a view to generate a response. Part of this overview will include looking at URL routing (including

6. Razor 2

Razor provides many advanced features and this module covers those. This includes the new tag helpers mechanism for providing encapsulation of rendering and behavior in server side rendering code. We'll also look at the new view components feature in MVC which allows for a better re-use approach than partial views.

Day 3

7. Model binding ; validation

Model binding is an important part of the MVC framework to ease input processing for controllers. We will see how data is mapped from the request to parameters of action methods and how complex objects can also be model bound. In addition, we will examine how MVC provides a complete framework for validation. We'll see how it uses validation helpers, ModelState, and data annotations to provide server-side validation. Finally, we will see how MVC's validation framework can also be used to perform client-side validation.

8. Filters, error handling ; logging

Filters are a mechanism in MVC to provide re-usable functionality that address cross-cutting concerns (such as security, logging, and error handling). This module covers the filter architecture, describes the different kinds of filters (such as action filters, result filters, exception filters, and resource filters), and shows how to build and use custom filters. We will then show how to capture error information for logging and how to display user-friendly error pages. To achieve this, we will utilize exception filters in MVC, as well as the diagnostics middleware provided for ASP.NET. Finally, we will show the official logging framework in ASP.NET Core and how your applications should be using it for logging.

9. Entity Framework Core

Entity Framework Core (or EF Core) is the next generation ORM for .NET. It has been redesigned from scratch to target .NET Core as well as meet the more modern demands of an ORM. EF Core is a very easy transition for those coming from the prior EF6, using a

11. ASP.NET Identity

This module covers the ASP.NET Identity framework which provides an implementation of storing and managing user credentials. This also involves storing claims, roles, email, and any other identity information required by an application. We'll also see what's involved with email verification, password resets, and two factor authentication.

12. Authorization

Authorization is the process of verifying that the current caller is allowed to make the request into your application. This step is separate and necessarily comes after authentication. ASP.NET provides a familiar approach for performing role-based authorization, but also has introduced a new policy-based authorization system. This module will cover both approaches to authorizing requests into your applications.

Day 5

13. Web APIs

ASP.NET Core MVC is designed to support both HTML content as well as JSON content. This module focuses on the aspects of MVC that are geared towards building Web APIs and RESTful services. This includes input and output formatters, content negotiation, and object results.

14. API security

The standard approaches taken to secure web pages don't apply when dealing with Web APIs. This module covers how API security must be treated differently and discusses the topics of CORS and token based authentication using OAuth2. We will see the middleware provided for ASP.NET to enable both of these features in your Web API application.

15. Tooling

One of the major changes in ASP.NET Core is the approach to tooling that an application developer can use. Microsoft has acknowledged that there are many great

areas), and the basics of model binding and the Razor view engine.

5. Razor 1

Razor is the view engine used to dynamically generate HTML responses in MVC. This module covers the Razor syntax and the underlying compilation model. We will also see Razor's support for reusable templates with partial views and layout templates.

more code-first approach (rather than a design heavy approach with XML schema files). This module will cover how to use EF Core with .NET Core and ASP.NET Core and how it's has changed from EF6.

Day 4

10. Authentication

Security is crucial for online applications. This module on security discusses the basics of authentication using claims based identity and the cookie authentication middleware for ASP.NET. We'll then turn to external authentication which could be a corporate directory (such as Active Directory or Azure), or a social media identity provider (such as Google or Facebook). We'll then discuss the modern security architecture for centralizing authentication and account management.

innovations occurring in the ecosystem around node.js and they are encouraging that for ASP.NET Core. This module covers how tools such as bower, grunt, and gulp are being used within ASP.NET to augment the web developer's workflow for tasks such as magnification and bundling of web assets (among other things).

Further Information:

For More information, or to book your course, please call us on Head Office 01189 123456 / Northern Office 0113 242 5931

info@globalknowledge.co.uk

www.globalknowledge.co.uk

Global Knowledge, Mulberry Business Park, Fishponds Road, Wokingham Berkshire RG41 2GY UK