

Developing ASP.NET Core MVC Web Applications

20486

Detalles del Curso

Audiencia(s): Desarrolladores

Tecnología: Visual Studio

Duración: 40 Horas

Material Didáctico: Oficial (Ingles)

Sobre este Curso

En este curso los desarrolladores web profesionales aprenderán a desarrollar aplicaciones avanzadas de ASP.NET Core MVC utilizando las herramientas y tecnologías .NET Core. La atención se centrará en actividades de codificación que mejoren el rendimiento y la escalabilidad de la aplicación del sitio web. Este curso también preparará al estudiante para el examen 70-486.

Perfil de Audiencia

Este curso está destinado a desarrolladores web profesionales que usan Microsoft Visual Studio en un entorno de desarrollo individual o en equipo, de tamaño pequeño a grande. Los candidatos para este curso están interesados en desarrollar aplicaciones web avanzadas y desean administrar el HTML presentado de manera integral. Quieren crear sitios web que separen la interfaz de usuario, el acceso a datos y la lógica de la aplicación.

Contenido del Curso

Módulo 1: Exploring ASP.NET Core MVC

Microsoft ASP.NET Core MVC y las otras tecnologías web de ASP.NET Core pueden ayudarlo a crear y alojar aplicaciones web dinámicas, potentes y extensibles. ASP.NET Core, del cual ASP.NET Core MVC forma parte, es un marco de código abierto y multiplataforma que le permite crear aplicaciones web. Puede desarrollar y ejecutar aplicaciones web ASP.NET Core en Windows, macOS, Linux o cualquier otra plataforma que lo admita.

ASP.NET Core MVC admite un ciclo de desarrollo ágil y controlado por pruebas. También le permite usar el último estándar HTML y marcos de trabajo front-end como Angular, React y más.

Lecciones

- Descripción general de las tecnologías web de Microsoft
- Descripción general de ASP.NET 4.x
- Introducción a la ASP.NET Core MVC

Laboratorio : Explorar ASP.NET Core MVC

Módulo 2: Diseño de aplicaciones web ASP.NET Core MVC

Microsoft ASP.NET Core MVC es un modelo de programación que puede usar para crear aplicaciones web potentes y complejas. Sin embargo, todos los proyectos de desarrollo complejos, y los proyectos grandes en particular, pueden ser desafiantes e intrincados de comprender completamente. Sin una comprensión completa de los propósitos de un proyecto, no puede desarrollar una solución efectiva al problema del cliente. Debe saber cómo identificar un conjunto de necesidades comerciales y planificar la aplicación web Modelo-Vista-Controlador (MVC) para satisfacer esas necesidades. El plan del proyecto que cree asegura a los interesados que comprende sus requisitos y comunica a los desarrolladores la funcionalidad de la aplicación web, su interfaz de usuario, estructura y almacenamiento de datos. Al redactar un plan de proyecto detallado y preciso, puede asegurarse de que las potentes funciones de MVC se utilicen de manera efectiva para resolver los problemas comerciales del cliente.

Lecciones

- Planificación en la fase de diseño del proyecto
- Diseño de modelos, controladores y vistas

Laboratorio : Diseño de aplicaciones web ASP.NET Core MVC

Módulo 3: Configurar Middlewares y Servicios en ASP.NET Core

ASP.NET Core es un marco que nos permite crear muchas aplicaciones de tipos diferentes. En este módulo, aprenderá cómo aprovechar el marco ASP.NET Core para manejar solicitudes y respuestas a través de un middleware existente y personalizado, y cómo configurar servicios para usar en middleware y en otras partes de la aplicación, como los controladores. Un middleware es un segmento de código que se puede usar como parte de la canalización de solicitudes y respuestas que nos permite manejarlos de acuerdo con cualquier parámetro relevante. Esto permitiría manejar múltiples solicitudes separadas de una manera completamente diferente y que reciban respuestas separadas. Los servicios son clases que exponen la funcionalidad que puede usar después en diferentes partes de la aplicación, sin tener que hacer un seguimiento manual del alcance en cada ubicación individual e instanciar cualquier dependencia. Esto se hace utilizando inyección de dependencias. La inyección de dependencias es una técnica utilizada por ASP.NET Core que nos permite agregar dependencias al código sin tener que preocuparnos por crear instancias de objetos, mantenerlos en la memoria o transmitir las dependencias requeridas. Esto permite que la aplicación se vuelva más flexible y reduzca los posibles puntos de falla cada vez que cambie un servicio.

Lecciones

- Configuración de Middlewares
- Configuración de servicios

Laboratorio : Configuración de Middleware y Servicios en ASP.NET Core

Módulo 4: Controladores en desarrollo

ASP.NET Core MVC es un marco para crear aplicaciones web utilizando el patrón arquitectónico Modelo-Vista-Controlador (MVC). El controlador es esencialmente responsable de procesar una solicitud web al interactuar con el modelo y luego pasar los resultados a la vista. El modelo representa la capa empresarial, a veces denominada dominio, y puede incluir objetos de datos, lógica de aplicación y reglas comerciales. La vista utiliza los datos que recibe del controlador para producir el HTML u otra salida que se envía de vuelta al navegador. En este módulo, aprenderá a desarrollar controladores. Los controladores son fundamentales para las aplicaciones MVC. Comprender cómo funcionan los controladores es crucial para poder crear los objetos modelo apropiados, manipularlos y pasarlos a las vistas apropiadas. Un controlador es una clase. Contiene varios métodos. Estos métodos se llaman acciones. Cuando una aplicación MVC recibe una solicitud, encuentra qué controlador y acción debe manejar la solicitud. Determina esto mediante el enrutamiento del Localizador uniforme de recursos (URL). El enrutamiento de URL es otro concepto muy importante necesario para desarrollar aplicaciones MVC. El marco ASP.NET Core MVC incluye un sistema flexible de enrutamiento de URL que le permite definir reglas de mapeo de URL dentro de sus aplicaciones. Para maximizar la reutilización del código en los controladores, es importante saber cómo escribir filtros de acción. Puede usar filtros de

acción para ejecutar un código antes o después de cada acción en su aplicación web, en cada acción en un controlador o en otras combinaciones de acciones de controlador.

Lecciones

- Escritura de controladores y acciones
- Configuración de rutas
- Escritura de filtros de acciones

Laboratorio : Controladores en desarrollo

Módulo 5: Desarrollar vistas

Las vistas son uno de los tres componentes principales del modelo de programación Modelo-Vista-Controlador (MVC). Puede definir la interfaz de usuario para su aplicación web creando vistas. Una vista es una combinación de marcado HTML y código C# que se ejecuta en un servidor web. Por lo tanto, para crear una vista, debe saber cómo escribir el marcado HTML y el código C# y usar las diversas clases auxiliares que están integradas en MVC. También debe saber cómo crear vistas parciales y componentes de vista, que representan secciones de HTML que pueden reutilizarse en su aplicación web.

Lecciones

- Creación de vistas con sintaxis Razor
- Uso de auxiliares de HTML Helpers y auxiliares de etiqueta
- Reutilizar código en vistas

Laboratorio : Desarrollar vistas

Módulo 6: Modelos en desarrollo

La mayoría de las aplicaciones web interactúan con varios tipos de datos u objetos. Una aplicación de comercio electrónico, por ejemplo, administra productos, carros de compras, clientes y pedidos. Una aplicación de red social puede ayudar a administrar usuarios, actualizaciones de estado, comentarios, fotos y vídeos. Un blog se usa para administrar entradas de blog, comentarios, categorías y etiquetas. Cuando escribe una aplicación web Modelo-Vista-Controlador (MVC), crea un modelo MVC para modelar los datos de su aplicación web. Dentro de este modelo, crea una clase de modelo para cada tipo de objeto. La clase de modelo describe las propiedades de cada tipo de objeto y puede incluir lógica de negocios que coincida con los procesos comerciales. Por lo tanto, el modelo es un elemento fundamental en una aplicación MVC. En este módulo, aprenderá cómo crear el código para los modelos.

Lecciones

- Creación de modelos MVC
- Trabajar con formularios
- Validar la aplicación MVC

Laboratorio : Desarrollar modelos

Módulo 7: Uso de Entity Framework Core en ASP.NET Core

Las aplicaciones web a menudo usan información y generalmente requieren un almacén de datos para esa información. Al representar páginas web que usan datos de un almacén de datos, puede crear una aplicación web que cambie continuamente en respuesta a las aportaciones de los usuarios, las acciones administrativas y la publicación de eventos. El almacén de datos suele ser una base de datos, pero ocasionalmente se utilizan otros tipos de almacenes de datos. En las aplicaciones Modelo-Vista-Controlador (MVC), puede crear un modelo que implemente la lógica de acceso a datos y la lógica de negocios. Como alternativa, puede separar la lógica de negocios de la lógica de acceso a datos utilizando un repositorio. Un repositorio es una clase a la que un controlador puede llamar para leer datos de un almacén de datos y escribir datos en un almacén de datos. Cuando escribe una aplicación ASP.NET, puede usar las tecnologías Entity Framework Core (EF Core) y Language Integrated Query (LINQ), que hacen que el código de acceso a datos sea muy rápido de escribir y fácil de entender. En este módulo, verá cómo crear un sitio web basado en bases de datos en MVC.

Lecciones

- Introducción a Entity Framework Core
- Trabajar con Entity Framework Core
- Usar Entity Framework Core para conectarse a Microsoft SQL Server

Laboratorio : Usar Entity Framework Core en ASP.NET Core

Módulo 8: Uso de diseños, CSS y JavaScript en ASP.NET Core MVC

Al crear aplicaciones web, debe aplicar un aspecto coherente a la aplicación. Debe incluir secciones consistentes de encabezado y pie de página en todas las vistas. Microsoft ASP.NET Core MVC incluye características como diseños de hojas de estilo en cascada (CSS) y plantillas que mejoran la apariencia y la facilidad de uso de su aplicación web. En ASP.NET Core MVC, puede crear elementos HTML interactivos utilizando JavaScript. Necesita saber cómo usar JavaScript en su aplicación web. Para simplificar la adición de JavaScript a su aplicación web, necesita saber cómo usar bibliotecas como jQuery.

Lecciones

- Uso de diseños
- Uso de CSS y JavaScript
- Uso de jQuery

Laboratorio : Uso de diseños, CSS y JavaScript en ASP.NET Core

Módulo 9: Desarrollo del lado del cliente

Al crear una aplicación, es importante saber cómo desarrollar la aplicación tanto del lado del cliente como del lado del servidor. En este módulo, aprenderá herramientas del lado del cliente que le permitirán crear amplias aplicaciones web a cualquier escala. Estas herramientas se basan en los temas tratados en el Módulo 8, "Uso de diseños, CSS y JavaScript en ASP.NET Core MVC". En este módulo, aprenderá cómo usar el marco Bootstrap para diseñar su aplicación web. Luego, aprenderá a usar Sass y Less, dos preprocesadores comunes de hojas de estilo en cascada (CSS) que agregan características a CSS, como variables, reglas anidadas y funciones. Estos mejoran en gran medida la capacidad de mantenimiento de aplicaciones web complejas. A continuación, aprenderá cómo configurar corredores de tareas como Grunt y Gulp y cómo usarlos para compilar archivos Sass durante la compilación de Microsoft Visual Studio. Aprenderá cómo usar el corredor de tareas gulp para realizar la agrupación y la minificación de archivos CSS y JavaScript y cómo configurar una tarea de vigilancia para compilar archivos Sass mientras escribe su código, sin la necesidad de reconstruir la solución. Por último, aprenderá herramientas de diseño receptivo que le permiten personalizar la pantalla de su aplicación web según las capacidades y especificaciones de un navegador web o dispositivo. Aprenderá a escribir consultas de medios CSS, cómo usar el sistema de cuadrícula sensible Bootstrap y cómo aplicar el diseño CSS flexbox a sus vistas.

Lecciones

- Aplicación de estilos
- Uso de Task Runners
- Diseño de respuesta

Laboratorio : Desarrollo del lado del cliente

Módulo 10: Pruebas y solución de problemas

Los sistemas de software como las aplicaciones web son complejos y requieren múltiples componentes, que a menudo son escritos por diferentes desarrolladores, para trabajar en conjunto. Las suposiciones incorrectas, la comprensión inexacta, los errores de codificación y muchas otras fuentes pueden crear errores que provoquen excepciones o comportamientos inesperados. Para mejorar la calidad de su aplicación web y crear una

experiencia de usuario satisfactoria, debe identificar los errores de cualquier fuente y eliminarlos. Tradicionalmente, los evaluadores realizan la mayoría de las pruebas al final de un proyecto de desarrollo. Sin embargo, recientemente se ha aceptado ampliamente que las pruebas durante el ciclo de vida del proyecto mejoran la calidad y aseguran que no haya errores en el software de producción. Debe comprender cómo ejecutar pruebas en componentes pequeños de su aplicación web para asegurarse de que funcionen como se espera antes de ensamblarlos en una aplicación web completa. También es importante que sepa cómo manejar las excepciones mientras ocurren. Mientras se ejecuta una aplicación, puede encontrar eventos inesperados. Es importante que administre sus excepciones correctamente y proporcione una buena retroalimentación de los usuarios mientras evita la filtración de información sobre la estructura de la aplicación. Finalmente, utilizando registros en toda la aplicación, puede supervisar las actividades de los usuarios que pueden llevar a problemas inesperados y luego puede encontrar soluciones a los errores, que normalmente no estaría seguro de cómo reproducir, siguiendo los flujos que ocurrieron en el entorno de producción y encontrando errores adicionales.

Lecciones

- Prueba de aplicaciones MVC
- Implementación de una estrategia de manejo de excepciones
- Registro de aplicaciones MVC

Laboratorio : Pruebas y solución de problemas

Módulo 11: Gestionar la seguridad

Dado que las aplicaciones web normalmente están dirigidas a usuarios que utilizan solo un navegador para usar la aplicación, es probable que haya muchos más usuarios que en el caso de las aplicaciones instaladas. Sin embargo, el carácter abierto de una aplicación web significa que la seguridad siempre debe estar presente en primer lugar al crearla. Como parte de la seguridad, debe decidir qué usuarios pueden realizar determinadas funciones acciones, mientras protege a los usuarios y su aplicación de terceros maliciosos con posible acceso a su aplicación. La autenticación es el acto de utilizar varios parámetros para asegurarse de que un usuario es quien dice ser. Al implementar la autenticación, puede determinar quién es un usuario y proporcionarle el contenido apropiado mientras utiliza sus aplicaciones. La autorización es el proceso en el que un usuario ya autenticado en la aplicación puede tener acceso a acciones o recursos específicos. Al utilizar la autorización, puede evitar que los usuarios accedan a material confidencial no previsto para ellos o que realicen acciones que no deberían poder hacer. Por último, en algún momento de la vida útil de las aplicaciones, pueden ser atacadas por usuarios malintencionados. Pueden variar en medios e intención, pero el costo de estar desprotegido puede ser grande. Puede perder usuarios potenciales afectados, se pueden cambiar por error datos valiosos y, en el peor de los casos, la aplicación completa puede

quedar inutilizable. Las soluciones a algunos de los ataques más populares se revisarán en este módulo.

Lecciones

- Autenticación en ASP.NET Core
- Autorización en ASP.NET Core
- Defensa de los ataques

Laboratorio : Gestionar la seguridad

Módulo 12: Rendimiento y comunicación

Las aplicaciones web modernas requieren interacciones complejas con los usuarios. Los usuarios a menudo solicitarán una gran cantidad de datos en un corto período de tiempo, mientras esperan datos relevantes tan pronto como estén en línea. Esto puede causar fácilmente una cantidad significativa de carga en un servidor no preparado, lo que implica operaciones innecesariamente complejas o repetidas y una carga pesada en su servidor. Afortunadamente, hay varias formas de reducir la carga. El almacenamiento en caché le permite almacenar solicitudes que se repiten con frecuencia, evitando la necesidad de realizar la misma lógica repetidamente. Al utilizar el almacenamiento en caché, puede reutilizar los datos que ya se han cargado y presentarlos al usuario. Esto proporciona al usuario un tiempo de respuesta rápido y reduce los recursos del sistema utilizados para llevar a cabo la lógica de la acción. Mientras tanto, el estado permite lograr un estado de coherencia entre las diferentes solicitudes. Al utilizar diversas formas de administración de estado, puede transformar la experiencia web normalmente sin estado en una que se personaliza a la medida de los clientes individuales, con diferentes usuarios que disfrutan de una experiencia separada y relevante en la misma aplicación. Finalmente, SignalR es un marco que permite la abstracción de varios protocolos de comunicación diferentes en una API fácil de usar, que le permite crear fácilmente un conjunto único de herramientas en el servidor y el cliente para facilitar las comunicaciones bidireccionales. Esto le permite concentrarse en la lógica que desea implementar, al mismo tiempo que le permite no tener que atender a navegadores específicos.

Lecciones

- Implementación de una estrategia de almacenamiento en caché
- Estado de gestión
- Comunicación bidireccional

Laboratorio : Rendimiento y comunicación

Módulo 13: Implementación de API web

La mayoría de las aplicaciones web requieren integración con sistemas externos como las aplicaciones móviles. Debe saber cómo usar las API web para promover la interacción de la aplicación con sistemas externos. Puede usar la API web para implementar servicios de transferencia de estado representacional (REST) en su aplicación. Los servicios REST ayudan a reducir la sobrecarga de la aplicación y limitan los datos que se transmiten entre los sistemas cliente y servidor. Debe saber cómo llamar a una API web utilizando el código del lado del servidor y el código jQuery para implementar efectivamente las API web de estilo REST en su aplicación.

Lecciones

- Presentación de API web
- Desarrollo de una API web
- Llamar a una API web

Laboratorio : Implementación de API web

Módulo 14: Hospedaje e Implementación

Las aplicaciones ASP.NET Core MVC están diseñadas para proporcionar un servicio a múltiples usuarios simultáneamente mientras solo requieren que se instale el servidor y que los clientes usen navegadores para acceder a él. Esto da como resultado aplicaciones altamente deseables que no dependen de que el usuario instale un software dedicado y garanticen que los clientes puedan acceder a él en una amplia variedad de máquinas. Para configurar una aplicación ASP.NET Core para un entorno de producción, deberá compilar su código y comprimirlo, y luego configurarlo y ejecutarlo en un servidor dedicado. El alojamiento implica el uso de un servidor dedicado para contener la aplicación compilada y servirla a los usuarios como un servicio basado en la web. Hay muchas tecnologías diferentes que se pueden utilizar para alojar su aplicación y debe elegir una que sea adecuada para sus requisitos. La implementación es el proceso donde se compila el código del proyecto y luego se transfiere al servidor de alojamiento de una manera que funcione dentro de los requisitos del entorno de alojamiento. Microsoft Azure es un servicio en la nube proporcionado por Microsoft. Se puede usar para alojar aplicaciones ASP.NET Core y es una herramienta popular en el mercado de la tecnología en la nube. Proporciona servicios convenientes relacionados con aplicaciones web en múltiples opciones de facturación de acuerdo con los requisitos.

Lecciones

- Alojamiento e implementación local
- Implementación en Microsoft Azure
- Fundamentos de Microsoft Azure

Laboratorio : Hospedaje e Implementación

Al Finalizar este Curso

Después de completar este curso, los estudiantes podrán:

- Describir la pila de Microsoft Web Technologies y seleccionar una tecnología apropiada para desarrollar cualquier aplicación dada.
- Diseñar la arquitectura e implementación de una aplicación web que cumpla con una serie de requisitos funcionales, requisitos de interfaz de usuario y abordar modelos comerciales.
- Configurar la canalización de las aplicaciones web ASP.NET Core utilizando middleware y aprovechar la inyección de dependencia en la aplicación MVC.
- Agregar controladores a una aplicación MVC para administrar la interacción del usuario, actualizar modelos y seleccionar y devolver Vistas.
- Desarrollar una aplicación web que use el motor de enrutamiento ASP.NET Core para presentar URL amigables y una jerarquía de navegación lógica a los usuarios.
- Crear vistas en una aplicación MVC que muestre y edite datos e interactúe con modelos y controladores.
- Crear modelos MVC y escribir un código que implemente la lógica de negocios dentro de los métodos, propiedades y eventos del modelo.
- Conectar una aplicación ASP.NET Core a una base de datos usando Entity Framework Core.
- Implementar un aspecto coherente en toda una aplicación web MVC.
- Escribir un código JavaScript que se ejecute en el lado del cliente y utilice la biblioteca de scripts jQuery para optimizar la capacidad de respuesta de una aplicación web MVC.
- Agregar paquetes del lado del cliente y configurar Task Runners.
- Ejecutar pruebas unitarias y herramientas de depuración contra una aplicación web en Visual Studio 2017.
- Escribir una aplicación MVC que autentique y autorice a los usuarios a acceder a contenido de forma segura utilizando Identity.
- Crear una aplicación MVC que resista los ataques maliciosos.
- Utilizar el almacenamiento en caché para acelerar las respuestas a las solicitudes de los usuarios.
- Usar SignalR para habilitar la comunicación bidireccional entre el cliente y el servidor.
- Describir qué es una API web y por qué los desarrolladores pueden agregar una API web a una aplicación.
- Describir cómo empaquetar e implementar una aplicación web ASP.NET Core MVC desde un equipo de desarrollo a un servidor web.