

Curso ATDEVCORE - Programação | ASP.NET Core e JavaScript

30,00 Horas

Introdução

Este curso estrutura a base de conhecimentos para o desenvolvimento de aplicações Web usando a estrutura e os padrões ASP.NET MVC (Model-View-Controller) no .NET Core. Como criar e consumir serviços RESTful com ASP.NET Web API e JavaScript serão alguns dos temas abordados.

Público-alvo

Programadores que pretendem explorar e entender o Model-View-Controller (MVC) na .NET Core.

Quando completar o curso

No final do curso os participantes estarão aptos a desenvolver aplicações .net MVC

Pré-requisitos

Os participantes devem ter um bom conhecimento de uma das linguagens de programação .NET C# ou VB .NET. Paralelamente é recomendado um conhecimento básico de HTML. Este curso aborda o .NET Core Framework.

Exames

(não existem exames)

Certificado

Este curso confere certificado de frequência a todos os participantes que frequentem no mínimo 80% das horas do curso.

Conteúdo em detalhe

O protocolo HTTP

Este módulo descreve o protocolo HTTP usado para comunicar com um servidor. Exploraremos como uma aplicação cliente e um servidor conseguem comunicar de forma legítima.

- Verbos HTTP
- Cabeçalhos
- Códigos de estado
- Redirecionamento

- Cache

ASP.NET MVC: Introdução

Este módulo incide sobre a forma como o MVC se enquadra perfeitamente no contexto de desenvolvimento web.

- O padrão Model-View-Controller (MVC)
- MVC em ASP.NET
- Centrado no cliente versus centrado no servidor

Os 3 principais componentes ASP.NET MVC

Depois de conhecer o MVC, é necessário perceber como é implementado no ASP.NET. Este módulo explora os componentes mais importantes da estrutura.

- Lidar com a interação do utilizador com o controlador
- Formatar a resposta com visualizações MVC
- Criar modelos MVC e ViewModel
- Reencaminhamento

ViewModels

Este módulo abordar ViewModels, que permitem manter uma separação entre o modelo e as Views

- Passar dados para uma View
- ViewModels

- Mapeamento

Fundamentos da linguagem JavaScript

Neste módulo toca nos princípios básicos do JavaScript que pode ser uma linguagem complexa, por vezes.

- Variáveis
- Primitivas e objetos
- Funções e o padrão de função de auto-invocação
 - Âmbito
 - Modo estrito
 - Manipulação de erros

Pipeline ASP.NET Core

O ASP.NET Core supera outras soluções de hospedagem na web.

- Middleware em ASP.NET Core
- Compondo o pipeline

Reencaminhamento de “end-point”

Na web o URL encaminha para a ação que o servidor executará. No ASP.NET Core o End-point Routing Middleware será responsável por essa ação. Um conhecimento sólido deste middleware é necessário para construir com ASP.NET Core.

- Introdução aos conceitos de reencaminhamento

- Definição de modelos de reencaminhamento
- Restrições de parâmetros de rota
- Transformadores de parâmetros de rota
- Compreender a correspondência de rota
- Usando reencaminhamento por convenção vs. por atributo
- Atributos de reencaminhamento
- Tratamento de erros de reencaminhamento

Web Design Responsivo com Bootstrap

Um site precisa de ter uma boa aparência em qualquer dispositivo; seja um laptop, tablet ou telefone. Com recurso ao Bootstrap poderemos otimizar o desenvolvimento.

- Normalizando e Reinicializando
- O sistema de grelha
- Classes disponíveis

ASP.NET Framework Pipeline

Antes de podermos analisar em profundo o modelo MVC precisamos de perceber como um pedido se transforma numa resposta. Analisaremos neste módulo como o cenário se processa.

- Limpe URI's com reencaminhamento ASP
- Controladores e ações

- ActionResult e ViewEngines

Reencaminhamento ASP.NET MVC

Configurar um URL para despoletar uma ação num controlador e ter controle total sobre isso não é difícil, mas essencial para qualquer programador ASP.NET MVC.

- Formatar o URL de pedido
- Convenções base de reencaminhamento
- Atributos de reencaminhamento
- Uso de áreas para a construção de sites MVC complexos
- Depuração de reencaminhamento de URL

Controladores ASP.NET MVC

Neste módulo, examinaremos mais de perto o C no MVC. O Controlador tem uma função muito importante a cumprir: é o componente que descobre o que fazer ao receber uma solicitação.

- Model Binding
- Diferentes resultados de uma ação
- Trabalhar com GET e POST
- Filtros de ação
- Redirecionamentos

Vistas com Razor (.cshtml)

Neste módulo exploraremos o Razor, sua sintaxe., os seus fundamentos, bem como alguns cenários avançados.

- Visualizações e mecanismos de visualização
- Layout e Seções
- Helpers HTML

Blocos de construção reutilizáveis de navalha

Reutilizar Razor.

- Vistas parciais
- Ações “Child”
- Modelos de exibição e editor

Data de validade

- Validação explícita versus validação implícita
- Validar dados usando anotações de dados
- Modelos de autovalidação
- Validando usando o Model Binder
- Validação do lado do cliente com JavaScript discreto

Injeção e Teste de Dependência ASP.NET MVC

Injeção de dependência é a arte de desacoplar um objeto de suas dependências. Além de benefícios como facilidade de manutenção e separação de interesses, torna também os testes muito mais fáceis. Neste módulo usaremos injeção de dependência para criar seus primeiros testes de unidade.

- Métodos de injeção de dependência
- Inversão de Dependências, Dependências Explícitas e Inversão de Controle
- Teste de Unidade de seus Componentes
- Stubbing, mocking e faking

Construindo serviços RESTful com API da Web

Ao criar uma aplicação centrada no cliente, é importante enviar e receber dados entre o cliente e o servidor. A API da Web usa REST para enviar dados. É baseado nos mesmos princípios do MVC.

- O que é REST?
- Construir uma REST Web-API com ASP.NET Web API
- Suporte a várias representações: XML e JSON

Autenticação

A autenticação permite identificar utilizadores de um sistema. Neste capítulo, examinamos os diferentes mecanismos de autenticação e suas utilizações.

- Integração com Windows
- Utilizador e palavra passe
- Usar autenticação externa, como Google ou Facebook

- Estender a autenticação com seus próprios dados
- Tokens
- “Personificação” e delegação

Publicar uma aplicação

Como fazer a implementação da aplicação instalada? Como o .NET é multiplataforma, pode criar-se tudo no Windows e, em seguida, executar no Linux. Também discutiremos várias opções de implementação.

- Opções de implementação (“deployment”).
- Implementação “cross-platform”
- Implementação “self-contained”
- Opções de Implementação