Curso ATDEVBLZ - Programação | Aplicações “single page” (SPA) com Blazor

14,00 Horas

Introdução

Blazor é uma nova tecnologia Microsoft para criar aplicações de página única (SPA). Ao contrário de estruturas como o Angular e o React; o Blazor permite chegar a este objetivo com recurso apenas a C# e à Framework.NET. Os seus vários componentes como o Razor, o WebAssembly, o DOM e o .Net Core interagem em conjunto para simplificar o processo de desenvolvimento.

Público-alvo

Quem pretenda ser um “early-adopter” da tecnologia Blazor.

Quando completar o curso

No final desta ação os participantes vão conhecer o contexto de Blazor, os seus componentes em detalhe.

Pré-requisitos

Esta formação requer conhecimento sobre a construção de sites com HTML e CSS, e também algum conhecimento sobre C# e ASP.NET.

Exames

(não existem exames)

Conteúdo em detalhe

Introdução ao WebAssembly e ao Blazor

Nesta introdução, veremos como os navegadores são capazes de executar assemblies .NET usando WebAssembly, Mono e Blazor. Este curso prático inicia com a instalação dos pré-requisitos para desenvolver com Blazor, seguido de um primeiro projeto Blazor no Visual Studio.

- Programação web: passado, presente e futuro
- O que é o WebAssembly.
- WebAssembly e Mono.
- Introdução ao Blazor
- Blazor do lado do cliente vs. Blazor do lado do servidor.
- Visão geral dos recursos do Blazor.
- Pré-requisitos de instalação do Blazor.
- Gerar um projeto com Visual Studio ou Code.

Blazor Data Binding

As aplicações modernos da web usam a abordagem Model-View-Controller (MVC), que depende muito da ligação aos dados. O Blazor não é uma exceção e veremos as diferentes formar de fazer "binding" aos dados com o Blazor.

- Visão geral Razor
- Binding de dados unilateral
- Tratamento de eventos e binding de dados
- Binding de dados bidirecional
- Monitorizar alterações
- Formulários e validação

Componentes do Blazor

No desenvolvimento da web moderno, construímos aplicações a partir de componentes, que normalmente são desenvolvidos a partir de outros componentes. Um componente Blazor é uma parte independente da interface do utilizador. Os componentes do Blazor são classes construídas a partir de Razor e C# com um determinado objetivo e contexto e são mais fáceis de entender, depurar e manter. Desta forma será possível usar o mesmo componente em páginas diferentes.

- O que é um componente Blazor?
- Construir um componente Blazor simples
- Parâmetros do componente
- Renderização condicional e ChildContent
- Separar a View e o Model-View
- Ligação de dados a componentes
- Splatting de atributos
- Noções básicas sobre EventCallback
- Usar componentes
- Construir uma biblioteca de Componentes.
- Estilizar componentes.

Serviços e "Dependency Injection"

Inversão de dependência é um dos princípios básicos de um bom design orientado a objetos. Neste capítulo, avaliaremos a inversão e a injeção de dependência e por que é uma parte fundamental do Blazer.

- Noções básicas sobre inversão e injeção de dependência.
- Inversão de “Control Containers”
- Construtor e injeção de propriedade
- Configurar injeção de dependência
- Blazor e injeção de dependência
- Construir serviços Blazor.

Armazenamento de dados e microsserviços

Em geral as aplicações de navegador do lado do cliente precisam armazenar alguns dados. Em alguns casos, como em jogos, a aplicação pode armazenar dados usando o armazenamento local do navegador. Mas na maioria dos casos, o armazenamento acontecerá no servidor que tem acesso aos mecanismos de bases de dados. Neste capítulo, cobriremos os fundamentos do armazenamento de dados usando o Entity Framework Core e expondo esses dados usando REST e microsserviços criados com ASP.NET Core.

- O que é REST?
- Invocar a funcionalidades de servidor com REST.
- Construir microsserviços simples com ASP.NET Core.
- O que é Entity Framework Core?
- Gerar bases de dados com Code First.
- Testar microsserviços.

Comunicação com microsserviços

Como comunicar com um serviço REST com o Blazor? Usar a classe HttpClient que você provavelmente já conhece de outros projetos .NET, mas com uma diferença.

- Envio e receção de dados.
- Usar a classe HttpClient.
- Os métodos HttpClientJSONExtensions.
- Assumir o controle com HttpRequestMessage.
- Recuperar dados do servidor.
- Armazenar alterações.

Reencaminhamento em aplicações de página única

O Blazor é uma estrutura .NET usada para construir aplicações de página única, tal como frameworks JavaScript populares, como Angular, React e VueJs. Mas o que é uma aplicação de página única (SPA)? Veremos como usar o reencaminhamento para saltar entre as diferentes seções de um SPA e partilhar dados entre diferentes componentes.

- O que é uma aplicação de página única?
- Usar componentes de layout
- Noções básicas sobre reencaminhamento
- Definir a rota modelo
- Redireccionamento para outras páginas
- Partilha de Estado entre componentes.

Interoperabilidade JavaScript

O JavaScript continua a ser um componente importante, por exemplo, o próprio Blazor usa JavaScript para atualizar o Document Object Model (DOM) do navegador a partir dos componentes do Blazor. Neste capítulo, veremos a interoperabilidade com JavaScript e contruir uma biblioteca de componentes Blazor.

- Porque é ainda necessário o JavaScript no Blazor?
- Invocar JavaScript a partir de C#.
- Invocar métodos .NET a partir de JavaScript.