



Blazor WebAssembly with MongoDB



หลักสูตรนี้เนื้อหาจะเป็นการประยุกต์ใช้คำสั่งในการจัดการกับฐานข้อมูล NoSQL ที่ชื่อ MongoDB โดยใช้ภาษา c#.net และการพัฒนา WebAssembly โดยใช้ Blazor โดยใช้ฐานข้อมูลเป็น MongoDB

วัตถุประสงค์:

- สามารถพัฒนา WebAssembly โดยใช้ Blazor ได้

กลุ่มเป้าหมาย:

- Developer
- เจ้าหน้าที่ดูแลด้านไอทีของบริษัท

ความรู้พื้นฐาน:

- มีความรู้พื้นฐานในใช้ภาษา c#.net

ระยะเวลาในการอบรม:

- 18 ชั่วโมง (3 วัน)

ราคาคอร์สอบรม:

- 10,500 บาท / คน (ราคานี้ยังไม่ได้รวมภาษีมูลค่าเพิ่ม)

วิทยากรผู้สอน:

- อาจารย์สนิทวงศ์ กมลภากรณ์

คอร์สที่ควรอบรมก่อนหน้า:

- ASP.NET Core MVC



เนื้อหาการอบรม:

Module 1: Introduction MogoDB and C#.net

- Why NoSQL and Why MongoDB?
- MongoDB.Entities
- MongoDB and LINQ
- Indexing and Profiling
- Working Directly with Documents in C#
- GridFS
- Aggregation Framework
- Security and Permissions

Module 2: Introduction Blazor WebAssembly What is Blazor?

- Blazor Hosting Models
- What is Blazor Server vs. Blazor WebAssembly?
- What is WebAssembly?
- Browser Compatibility
- Web Assembly vs. JavaScript
- How does .NET Core / C# run in a web browser?

Module 3: Blazor WebAssembly Application

- Project Template
- Create a New Application
- Hosting Blazor WebAssembly with a ASP.NET Core MVC Server
- Configuration
- Dependency Injection
- Environments
- Logging
- Handling Errors
- Debugging WebAssembly

Module 4: Razor Components and Data Binding

- What is a Component?
- Creating a Data Model
- Binding the Data Model to the HTML



- Passing Arbitrary Attributes
- Handling Events
- Manually Trigger State Updates and Re-rendering

Module 5: Composing Razor Components

- Decompose a Component into Smaller Components
- One-Way Data Binding
- Two-Way Data Binding
- Pass Data from a Parent Component to a Child Component using Parameters
- Pass Data from a Child Component to a Parent Component using Event Callbacks
- Use Keys to Optimize Performance
- Use Refs to Access DOM Elements
- Razor Component Libraries
- Razor Component Design Patterns
- Parameters are Immutable
- Lift State Up
- Managing State in General

Module 6: Razor Component Forms

- What is the purpose of Form?
- Collecting Data using a Form, Input, Select and Textarea Elements
- Explore Form Element Two-Data Binding
- Build Forms with the Blazor Edit Form Razor Component
- Explore the Concept of the Edit Context
- Use the Specialized Edit Form Controls
- Input Text
- Input TextArea
- Input Select
- Input Number
- Input Checkbox
- Input Date
- Applying Validation to the Form
- Decorating the View Model with Validation Attributes
- Code Custom Validation Attributes



Module 7: Razor Component Pages

- What is the Page model?
- Differences between Razor Pages and Razor Components
- Using a Razor Component as a Page
- Explore the Router Component
- Configuring Page Routing
- Route to Components from Multiple Assemblies
- Using Route Parameters
- Using the Query String
- Applying Authorization to a Razor Component Page
- Using Authorization within the Component Tree

Module 8: Using Server Data

- ASP.NET Core MVC Web API
- What is ASP.NET Core MVC?
- What is a REST API?
- What is an API Controller?
- Injecting the Http Client
- Exploring the Http Client
- Calling a REST API from a Blazor Component using the HttpClient

Module 9: Interacting with JavaScript

- What is the JavaScript Interop?
- When is JavaScript needed?
- Synchronous vs. Asynchronous Calls
- How to call a JavaScript function from a Component
- How to call C# code from JavaScript
- Calling Static Methods
- Calling Instance Methods
- Organizing JavaScript Code within a Blazor WebAssembly App