Building Cloud based services using .NET Core 2.2 and Azure

Training Details:

|  |  |
| --- | --- |
| Duration: | 24 Days |
| Description: | Building Cloud based microservices applications using NoSQL databases, .NET Core 2.2 and Microsoft Azure |
| Objectives: | To learn ASP.Net Core in depth and apply its concept to build application. |
| Participants’ Entry Profile: | Participants attending this course must have development experience on:   * Model View Controller * Basic Concepts of ASP.Net MVC 5 * RESTful Services * Basic Knowledge on Cloud Platform |
| Training Methodology: | The workshop will follow Synergetics methodology of   * Concept Visualization * Active Experimentation * Application Development   The workshop will be 100% Hands-On with each participant having access to system during the session |

Setup Requirements:

|  |  |
| --- | --- |
| Hardware and Software Requirements: | Participant’s as well as Trainer’s Machine are required to have:  Hardware   * Intel Pentium 4 [2+ GHz recommended] * 4 GB RAM * 50 GB HDD space * LAN connectivity * Good Internet connectivity and bandwidth * Open Internet Connectivity Required   Software [Installed]   * Windows 10 with Hyper-V enabled * Visual Studio Code * Visual Studio 2017 Community Edition * MongoDB 3.6 or later * NoSQL Booster for MongoDB * Chrome and Mozilla /Opera latest * .NET Core 2.1 installed * Postman Desktop client * Azure SDK for .NET * Azure PowerShell * Azure CLI 2.x * Docker for Windows * Azure Subscription |
| Training Lab Requirements: | Whiteboard 6 feet by 4 feet (minimum)  Whiteboard markers – Red, Blue, Green, Black  Video Projector (1024 X 768 resolutions) |

Course Contents:

Day 1

Overview of HTML 5

* HTML5 DOCTYPE
* Page Encoding
* HTML5 Mark-up
  + New and Updated Elements
  + Structural Elements
  + New Attributes
* HTML5 and CSS3 together
* Browser Support

HTML 5 Forms

* Current Solutions
* New Input Types – email, date, time, url, number, range, color and more
* New form Attributes – novalidate, formaction, formmethod
* Form Validation attributes
* Browser Support

Using new Audio and Video elements

* The State of Web Audio and Video Based on Plugins
* New Audio/Video Mark up
* Attributes and Methods
* Understanding Audio/Video Events

Graphics using HTML5 Canvas

* Overview of Graphics in The Browser
  + Canvas Vs. SVG
* Using a Canvas
* Context and Coordinates
* Drawing Shapes and images

Styling web pages using CSS3

* Selective Styling
* Working with Pseudo-classes
* Working with Pseudo-elements

Working with Text, Color, Background and Table

* HTML5 Text formatting Tags
* CSS3 Font and Text properties
* CSS3 Color and Background properties
* CSS3 List and Table properties

Layout your Webpages

* CSS3 Box properties
* CSS3 display, float, margin, position properties
* Multi column layout

jQuery Functions

* Modifying HTML content
* Handling events

jQuery AJAX

* Loading content asynchronously
* Fetching JSON data
* Receiving HTML with AJAX
* Receiving JSON with AJAX

Day 2

Introduction to Bootstrap

* Designers and Developers
* What is Bootstrap 3
* The Mobile Web
* Responsive Web Design
* Mobile First Responsive Web Design
* Bootstrapping of the Web

Grid System

* Using the Grid System
* Multiple Grid Types
* Using Multiple Grids
* Offsets and Rows
* Images and Responsive Helpers
* Theming Bootstrap

Bootstrap 3 common metaphor

* Typography
* Buttons
* Icons
* Navs and NavBars
* Lists
* Tables
* Forms

Bootstrap Components

* Page Header and Breadcrumbs
* Dropdowns
* Button Dropdowns
* Using a Dropdown for a Select
* Input Groups
* Thumbnails

Day 3

Angular Modules

* Features of Angular 7
* CLI improvements
* Grammar conventions
* Angular 7 Project Structure
* Library projects
* @NgModule decorator
* Import and export modules
* Declarations in module
  + Registering Components
  + Registering Pipes
* Registering providers in module
* Bootstrapping component
* Ng-add and ng-update commands

Components and Directives

* Modules
* Components
* Built-in Directives
  + NgIf ..Else
    - Ng-template replaces template
  + NgSwitch
  + NgStyle
  + NgClass
  + NgFor with “as” support
  + Getting an index
  + NgNonBindable
* Custom Components
  + Component Decorator
  + Exporting components
  + Accessing elements using Reference (#ref)
  + Defining events
* Built-in Pipes and custom pipes

Forms

* Template Driven Forms
  + NgForm and NgModel
  + FormsModule
* Reactive Forms
  + FormControl and FormGroup
  + ReactiveFormsModule
* Using Validators
  + Custom validators
  + Automatic “novalidate” attribute for Forms
  + Email Form validator
  + CompareWith attribute for select control
* Watching form changes
* ngModelChange

Day 4

Routing and Navigation

* New and faster view engine
* Importing RouterModule
* Creating Routes
* Passing Route parameters
  + ActivatedRoute
  + Params.snapshot
  + ParamsMap
  + Subscribing parameters
  + Passing multiple parameters
* RouterOutlet using <router-outlet>
* RouterLink using [routerLink]
* Defining child routes

Lifecycle Hooks

* OnInit interface
* OnInit and OnDestroy
* OnChanges
* Constructor vs ngOnInit

Injectable Services

* Dependency Injection
* Built-in Services
* Creating Custom services
  + Registering services in module
  + Registering services in Component
* Injecting Services in Constructor
* Need for @Injectable
* Declaring the providers inside the service

Day 5

Working with RxJS

* Creating observables
* Promise

Http Services

* Importing HttpClientModule
* Ignoring map methods
* HttpHeaders and options
* Interceptors in Angular
* Requesting REST services
  + GET, POST, PUT and DELETE
  + Consuming Results

New **Features in Angular 7**

* Drag and drop in Angular 7
* Virtual Scrolling
* Updating from Angular 6

Day 6

NOSQL Basics

* Introduction to NOSQL
* Need for NOSQL Database
* Characteristics of NOSQL Databases

Overview of MongoDB

* What is MongoDB
* History of MongoDB
* Features of MongoDB.

Getting Started

* Documents
* Collections
* Databases
* Getting and Starting MongoDB
* MongoDB Shell

Creating, Updating and Deleting Documents

* Inserting and Saving Documents
* Removing Documents
* Updating Documents
  + Document Replacement
  + Using Modifiers
  + Upserts
  + Updating Multiple Documents

Querying

* Introduction to find
  + Specifying which keys to return
  + Limitations
* Query Criteria
  + Query Consitionals
  + OR Queries
  + $not
  + Rules for Conditionals
* Querying Array
  + $all
  + $size
  + $slice
* Querying Embedded Documents
* Cursors
  + Limits
  + Skips
  + Sorts

Aggregation

* Count
* distinct
* Group
* MapReduce.

Day 7

Introduction to ASP.NET Core

* New ASP.NET Core framework
* Characteristics of ASP.NET Core
* Cross platform development
  + Windows
  + Mac
  + Linux
* ASP.NET Core Architecture
* ASP.NET 4.6 vs ASP.NET Core

Developing with ASP.NET Core

* ASP.NET Core Project structure
* Installing packages
* Working with Visual Studio
* Working with Visual Studio Code

Middleware in ASP.NET Core

* What is Middleware?
* Middleware vs HttpModules and HttpHandlers
* New Http Request Pipeline
* Application startup class
* Built-in modules
  + Exception handlers
  + Static files server
  + Authentication
  + MVC
* Ordering Middleware
* Writing custom middleware
* Serving Static Files
  + Serving a default document
  + Enabling Directory browsing
  + Use FileServer

Dependency Injection

* Understanding DI system
* Constructor injection
* Service lifetime
  + Singleton
  + Transient
  + Scoped
* Register custom services
* RequestServices collection
* Need for Third party DI engines

Day 8

Configuration in ASP.NET Core

* Configuring with Files
* Command-line arguments
* Environment variables
* In-memory provider
* Binding to a POCO class
* Binding to an Object Graph

Hosting environment

* Self-Host vs Server
* Setting up host
* Using Kestral server

ASP.NET Core MVC Project

* Creating controllers, Views and models
* Working with Tag Helpers
  + Writing tag helpers
  + Tag helpers in forms

Routing in ASP.NET Core

* Understanding routing feature
* Configuring Routing middleware
* Multiple routes
* Attribute routing
* Route constraints
* Generating URLs by action name and route
* Defining areas

Day 9

Session and application state

* Session state
* Cookie based TempData provider
* Query Strings
* Post data and hidden fields
* Cookies
* HttpContext.Items
* Configuring session
* Setting and getting session values

ASP.NET Core MVC with EF Core

* EF Core NuGet packages
* Register Context with DI
* Configuring DBContext and model classes.
* CRUD operations using EF Core
* Paging, Sorting, Filtering, grouping
* Code migrations
* Creating and Registering repositories
* Model Binding
* Validations on models
* CRUD operations using MVC controllers

Day 10

ASP.NET Core Web API

* Creating API controller
* Configuring Routes
* Support with EF Core
* CRUD operations using Web API

Exception Handling

* Developer exception page
* Custom exception handling page
* Configuring status code pages
* Server exception handling
* Exception filters
* Handling model state errors

Day 11

EF Core in Web API

* Configuring EF Core in Startup
* Creating DbContext
* Creating Repository
* Performing CRUD operations
* Paging and Sorting data
* Testing with POSTMAN

Documenting Web APIs

* Adding Swagger to Web API
* NuGet Packages
* Configuring swagger to Middleware
* Customizing Swagger
* Creating Help Pages using Swagger
* Swagger UI
* Testing Swagger

Formatting Responses

* Format-Specific Action Results
* Content negotiation
* Configuring Formatters
* XML and JSON serialization
* Custom formatters

Day 12

Security on Web API

* Configuring Identity services
* Configuring authentication
* Preventing Cross Site Scripting
* Enabling Cross-Origin Requests (CORS)
* JWT Token Authentication

Day 13

Rabbit MQ

Day 14

Docker

* Images and containers
* Docker networking
* Docker storage
* Docker registry

Day 15

Containerizing .net core application

* Docker compose
* Building platform independent docker images
  + Multi stage building

Containerizing Services

* Creating docker file
* Building docker images
* Testing locally

Configuring authentication

Code Challenge

* Implement the Case Study and Test locally

Day 16

Introduction to Azure

* Overview Cloud platform
* Understanding Azure Services, Azure portal
* Creating Azure subscription
* Understanding different deployment models
* Managing Azure with Azure PowerShell and CLI
* Overview of Azure Resource Manager
* Creating ARM Templates using VS
* Deploying templates using PowerShell

Azure Storage service

* Understanding the Azure Storage Account
* Understanding Storage Redundancy Types
* Azure Storage Service abstractions (Azure Blobs, Tables, Queues, Files)
* Provisioning Azure Storage Account
* Working with Azure Storage Containers and Blobs
* Creating Shared Access Signature
* Creating Stored Access Policies
* Connecting Storage account services with C# application

Day 17

Introduction to Cosmos DB

* Overview of Cosmos DB Service
* Document DB vs Cosmos DB
* Multi-model storage
* Supported APIs
* Integration with Azure Services
* Working with CosmosDB Emulator
* Limitations of emulator
* Installing and configuring emulator
* Getting connection sting

Working with Documents

* Cosmos DB Database
* Collection and Documents, SQL API (Document DB API)
* MongoDB API
* Inserting documents
* Updating and Deleting Documents
* Querying documents
* Aggregation support, Search, Sort and Filter

Day 18

Designing a microservice-oriented application

* Application specifications
* Benefits of a microservice-based solution
* Downsides of a microservice-based solution

Creating a data-driven CRUD microservice

* Designing a CRUD microservice
* Implementing a simple CRUD microservice
* Generating Swagger description metadata
* Implementing event-based communication between microservices
* Using message brokers and services buses for production systems
* Tackling Business Complexity in a Microservice
* Distributed Transactions and Two-Phase commit model

Introduction to Containers and Docker

* What is Containerization
* Containers vs Virtual Machines
* What is Docker?
* Comparing Docker containers with virtual machines
* Understanding Docker terminology
* Docker architecture, Benefits of Docker
* Docker containers, images, and registries

Day 19

Creating Docker images

* Installing and Configuring Docker
* Docker layers
* Docker commands, Docker file
* Creating docker images
* Running docker images
* Stop, List and remove docker containers

Working with Docker Hub

* Registering in Docker Hub
* Creating repositories
* Login in to Docker Hub from Docker command line
* Tagging images
* Pushing images to Docker Hub
* Pulling images from Docker Hub
* Pushing images with different tags

Kubernetes introduction

* What is Kubernetes?
* Why to use Kubernetes.
* Kubernetes cluster architecture
* Containers and Pods, Cluster master and nodes
* Deployment State and Manifest files
* Services and Deployments
* Deploying containers from Docker Hub
* Deploying containers from ACR

Azure Container Services - AKS

* Provisioning Azure Container Services -AKS
* AKS architecture
* Deploying containers in to AKS
* Working with kubetel
* Deploying using YAML
* Creating and managing Services using Kubernetes UI

Day 20

Deploying Microservices using Azure Service Fabric

* Packaging Microservices applications
* Creating Service Fabric Clusters
* Service fabric architecture
* Services in Azure Service fabric
* Scaling Services – Scale up and Scale Down

Azure Container Instance (ACI)

* What is ACI? When to use ACI?
* Benefits of using ACI
* Running containers using ACI
* Container Groups
* Multi-container groups

Azure Event Grid

* What is Event Grid?
* When to use Event Grid?
* Benefits of using Event Grid
* Event Sources and Handlers
* Sending message to EventGrid
* Consuming messages from event grid

Day 21

Azure App Service API Apps

* Create and deploy API Apps
* Automate API discovery by using Swashbuckle
* Use Swagger API metadata to generate client code
* Monitor API Apps

API management Service

* Create managed APIs
* Configure API management policies
* Protect APIs with rate limits
* Configuring APIs, Import and Publish API
* Create and publish a product,
* Mock API responses
* Transform and protect your API
* Monitor published APIs
* Debug your APIs using request tracing
* Use revisions to make non-breaking changes safely
* Publish multiple versions of your API
* Customizing developer Portal

Day 22

Azure Functions

* What is Serverless computing
* Serverless computing options
* Create Azure Functions
* Implement a webhook function
* Create an event processing function
* Implement an Azure-connected function
* Custom binding and configuring proxies
* Integrating with storage
* Integrating with App Service Plan

Azure App Service Logic Apps

* Create a Logic App connecting SaaS services
* Understanding Triggers and actions
* Trigger a Logic App from another app
* Create custom and long-running actions
* Monitor Logic Apps

DevOps using VSTS

* Understanding DevOps CI/CD
* Introduction to VSTS
* Getting started with VSTS
* Git installation
* Continuous Integration with VSTS
* Publish Source code to Github
* Create a CI Pipeline
* Add task on VSTS: Build Image
* Add task on VSTS: Push Image
* Enable CI trigger in VSTS,
* Continuous Testing: Integration Testing
* Publish Test Results from CI Builds
* Continuous Testing Using DotNet Watch Tools
* Continuous Deployment with VSTS
* Configuring the artifacts
* Enabling Continuous Deployment trigger
* Configuring the environments: Dev, Staging & Prod
* Run the CI & CD pipeline

Day 23 & 24

Project Case study implementation