Node JS, Type Script and Express JS

Duration: 4 half days

Pre-requisites:

- Participants should have working knowledge on basic HTML, CSS and JavaScript [Functional style of programming and NodeJS]. [https://www.w3schools.com/html/]
- Complete the following JavaScript courses
  - https://github.com/timoxley/functional-javascript-workshop
  - https://github.com/workshopper/learnyounode
  - https://www.w3schools.com/js/js_es6.asp

Course Objective:

- Learn Node.js, Express & how to build REST APIs using TypeScript

Node.js

- Node JS Architecture – Single Threaded Event Loop
  - V8 engine
  - NodeJS Bindings [API]; Event Queue; Event loop; Libuv [Async I/O); Worker Threads
- When and When not to use is NodeJS?
- Writing asynchronous code
- Modularizing code
  - Understanding built-in modules
  - Techniques for modularizing JavaScirpt code
  - Package.json
    - Export and import modules
    - Using npm for third-party modules
- Handling Exceptions
- Events and Streams
  - Understanding Events
  - EventEmitter class
  - Understanding Streams
  - Reading and writing streams
  - Using pipe()
- HTTP Server with Node.js and Core http Module
  - Node.js, Web Apps and http Core Module
  - Node.js Hello World HTTP Server
  - Node.js Hello World HTTP Server Demo
- **Unit testing with Mocha and Chai**
- **Child Process**
  - Asynchronous Process creation
    - exec, execFile, fork and spawn
  - Synchronous Process creation
    - execSync, execFileSync, and spawnSync
  - Events:
    - close
    - disconnect
    - error
    - exit
    - message

        - Cluster
            - take advantage of multi-core systems
    - Worker Thread
        - How worker threads work?
        - Using Worker Threads
        - Creating and executing new Workers
    - Debug
        - Inspect
        - Node.js debugging in VS Code
        - V8 Inspector Integration for Node.js
        - Enabling remote debugging scenarios
        - Node inspector

TypeScript

- JavaScript types
- Typescript types
    - Numbers, string, Booleans
    - objects and arrays
    - Tuples, Enum
    - The any and unknown types
- Functions
- Next Generation JavaScript and TypeScript
    - The let and const keywords
    - Arrow Functions
    - Default Function parameters
    - The spread operators
    - Rest Parameters
    - Destructuring arrays and objects
    - Class
        - Private and public access modifiers
        - Inheritance
        - Protected access modifiers
        - Getters and setters
        - Static methods
        - Abstract classes
    - Interfaces
    - Generics
    - Generic Function
    - Generic classes
    - Modules and Namespaces
        - Using ES modules
        - Using import and export
- Decorators
    - Decorator factories
    - Building decorators
        - Class decorators
        - Method decorators
        - Property decorators
        - Parameter decorators

JavaScript Build Automation tools

- Rush
  - What is Monorepo?
  - Using Rush
    - Rush Commands
  - Rush Stack: Heft
    - Architecture: Action, plugin, hook, task, stage, rig package and rig profile
    - Getting started with Heft
    - Adding tasks
      - TypeScript, ESLint and JEST
    - Interfacing with Rush
    - Using

Express.JS: Building REST APIs

- Installing Express.js
- REST APIs and JSON
  - Using Express to Provide an API
  - CRUD operations
  - Routing
    - Routes and SEO
    - Route Handlers
    - Route Paths and Regular Expressions
    - Parameters and queries in routing
    - Declaring Routes in a module
- Error Handling
  - Catching errors
  - Default handlers
  - Writing error handlers
- Debugging Express
- Cookies and Sessions
  - Cookies in Express
    - cookie-parser, cookie-session
  - Using sessions
    - Implement Flash messages
  - Session Management
  - Secure use of cookies
    - Using Cookie flags “secure” – for HTTPS and “HTTPOnly” -- to help prevent attacks such as cross-site scripting
    - Cookie Scope