# Web API Development

* **Serialization in C#** 
  + **Serialization of Data:** It is the process of converting an object or data structure into a format that can be easily stored, transmitted, or reconstructed later.
  + Process use for converting object to JSON and back.
  + Use **System.Text.Json** for this process.
  + Why is serialization important?
    - Data Persistence
    - Data Transfer
    - API Communication
    - Data Backup
  + Common formats for serialization: JSON, XML, Binary
* **System.Text.Json**
  + Use to handle JSON data
  + It includes various JSON operations like creation, update, deletion or traversal.
  + **JsonValue**:
    - Represents simple JSON values: numbers, strings, booleans, or null.
    - Extremely lightweight
    - It represents only single values.
  + **JsonObject**:
    - A collection of key-value pairs in JSON.
    - Dynamical representation of data.
    - Use for dynamic JSON creation and manipulation.
  + **JsonArray**:
    - Represents sequential JSON arrays.
    - Efficient for sequential data access.
  + **JsonDocument**:
    - A read-only representation of a parsed JSON document.
    - Cannot modify the JSON structure.
  + **JsonNode:**
    - The abstract base class for JsonValue, JsonObject, and JsonArray.
    - Offers flexibility and dynamic JSON manipulation capabilities.
* **LINQ**:
  + It is powerful querying language.
  + It allows to perform data operations on various data sources like collection, databases, XML etc.
* **Lambda expressions:**
  + A **lambda expression** is an anonymous function (a function without a name) in C# that allows you to define a block of code that can be passed as a parameter or used inline.
  + Commonly used to define **delegates** and **expressions** in a functional style.
  + Concise and expressive way to handle temporary functionality.
  + **Syntax:**
    - (parameters) => expression\_or\_statement\_block
      * **parameters**: A comma-separated list of input parameters (can be zero or more).
      * **=>**: The lambda operator, which separates the parameters from the body.
      * **expression\_or\_statement\_block**: The body of the lambda expression, which can be a single expression or a block of code.