# **MINIMAL API - POC**

## **MINIMAL API:**

**Minimal API** is a feature introduced in .NET 6; a framework developed by Microsoft for building applications. It allows developers to create lightweight and concise APIs with minimal ceremony, reducing the amount of boilerplate code required.

The need for Minimal API arises from the desire to streamline the process of building APIs and making it more intuitive for developers. Traditional API development in .NET often involves creating controllers, routing configurations, and other infrastructure-related code. While this approach offers flexibility and extensibility, it can sometimes be overwhelming and time-consuming for simple APIs.

Minimal API simplifies the API development process by providing a more focused and straightforward approach. It enables developers to define endpoints and route handlers using a concise syntax, often in a single file.

**The key features and benefits of Minimal API include:**

1. ***Reduced boilerplate code***: Minimal API eliminates the need for writing extensive infrastructure-related code, such as controllers, routing attributes, and configuration files. It allows developers to focus on defining endpoints and implementing the core logic.

2. ***Improved readability***: With Minimal API, the code becomes more concise and easier to read. The reduced ceremony and simplified syntax make it simpler to understand the API's structure and behavior.

3. ***Rapid development***: By minimizing the required code and simplifying the development process, Minimal API enables faster development cycles. It accelerates the iteration and prototyping of APIs, especially for small or straightforward use cases.

4. **Improved *performance***: The reduced layers of abstraction and simplified routing can lead to improved performance compared to traditional API development approaches.

5. ***Seamless integration with existing infrastructure***: Minimal API can coexist with existing infrastructure and frameworks. It can be gradually adopted within an existing application or used to build new standalone APIs.

Overall, Minimal API provides a **lightweight** and **intuitive** way to develop APIs in .NET, making it easier for developers to create simple APIs quickly without sacrificing extensibility or performance. It offers a balance between simplicity and flexibility, allowing developers to choose the level of complexity that suits their specific use cases.

## **EMPLOYEE – CRUD:**

1. Get an Employee/Get an Employee details by ID
   1. Get an employee detail based on the ID ("/GetEmployeeByID/{id}")
   2. Also, Get employee lists (/EmployeesList)
2. Add an employee detail
   1. Add an employee detail such as Name, Age, Salary, Experience and Role ("/AddEmployeeDetails")
3. Update an employee detail
   1. Update an employee detail such as Name, Age, Salary, Experience and Role ("/UpdateEmployeeDetails/{id}")
4. Delete an employee detail
   1. Delete an employee detail from DB ("/DeleteAnEmployeeDetail/{id}")

***Note:*** *In this MinimalAPI application, I used DB as* ***‘UseInMemoryDatabase’.***

In EF Core, **UseInMemoryDatabase** is a method that allows developers to configure their application to use an in-memory database for testing or development purposes. An in-memory database is a lightweight, transient database that resides in the application's memory rather than persisting data to disk or a traditional database server.