To **compile and run** ASP.NET Core Web API project EmployeeCafeManagement.Api, (which includes the EmployeeController and CafeController and related MediatR-based CQRS commands/queries), follow these **step-by-step instructions**.

**✅ Prerequisites**

Make sure the following are installed:

* ✅ [.NET 8.0 or later](https://dotnet.microsoft.com/en-us/download)
* ✅ Visual Studio 2022+ **or** Visual Studio Code
* ✅ SQL Server running (if using EF Core with SQL Server)
* ✅ Your project structure is similar to:

EmployeeCafeManagement/

├── EmployeeCafeManagement.Api/ → Web API project (controllers, EmployeeCafeManagement.sln)

├── EmployeeCafeManagement.Application/ → CQRS commands, queries, middleware

├── EmployeeCafeManagement.Domain/ → Entities, DTOs, interfaces

├── EmployeeCafeManagement.Infrastructure/ → DbContext, repository implementations

├── EmployeeCafeManagement.Tests/ → xUnit tests

**🛠️ Step 1: Build the Solution**

**If you're using Visual Studio:**

1. Open the solution (.sln) file.
2. Right-click the solution in Solution Explorer and select **Build Solution** (Ctrl+Shift+B).

**If you're using CLI:**

dotnet build

**▶️ Step 2: Run the API Project**

**Visual Studio:**

* Right-click EmployeeCafeManagement.Api → **Set as Startup Project**
* Press F5 to run with debugger or Ctrl+F5 to run without debugger

**CLI:**

Navigate to the API project folder:

cd EmployeeCafeManagement.Api

dotnet run

By default, the API will launch at:

https://localhost:5001

http://localhost:5000

You can verify this via launchSettings.json or by checking the terminal output.

**🌐 Step 3: Test API Endpoints**

You can use:

* [Postman](https://www.postman.com/)
* [curl](https://curl.se/)
* [Swagger](https://github.com/domaindrivendev/Swashbuckle.AspNetCore) (enable it in your project)

Example request:

curl -X GET <https://localhost:5001/api/Cafe> - To run Cafe API

curl -X GET <https://localhost.5001/api/Employee> - To run Employee API

**Sample Requests:**

**✔️ Get Employees by Café:**

curl <https://localhost:5001/api/employee?cafe=Starbucks>

**✔️ Get Cafe by Location:**

curl https://localhost:5001/api/employee?location=Singapore

**✔️ Create Employee:**

POST https://localhost:5001/api/employee

Content-Type: application/json

{

"id": "E001",

"name": "Alice",

"emailAddress": "alice@example.com",

"phoneNumber": "12345678",

"gender": "F",

"cafeId": "GUID-HERE",

"startDate": "2025-04-19"

}

**✔️ Create Cafe:**

POST https://localhost:5001/api/cafe

Content-Type: application/json

{

"id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",

"name": "Butter Lussy",

"description": "A Lush Creamy Lussy drink with Chilled Flavour.",

"location": "Punja",

"logo": <https://example.com/logos/sunset-brew.png>

}

**✔️ Update Employee:**

PUT https://localhost:5001/api/employee

Content-Type: application/json

{

"id": "E001",

"name": "Alice Smith",

"emailAddress": "alice.smith@example.com",

"phoneNumber": "96442123",

"gender": "F",

"cafeId": "GUID-HERE",

"startDate": "2025-04-20"

}

**✔️ Update Cafe:**

PUT https://localhost:5001/api/cafe

Content-Type: application/json

{

"id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",

"name": "Butter Lussy",

"description": "A Lush Creamy Lussy drink with Chilled Flavour.",

"location": "Punjab",

"logo": <https://example.com/logos/sunset-brew.png>

}

**✔️ Delete Employee:**

DELETE <https://localhost:5001/api/employee?id=E001>

**✔️ Delete Cafe:**

DELETE https://localhost:5001/api/cafe?id=3fa85f64-5717-4562-b3fc-2c963f66afa6

**✅ Ensure These Are Properly Configured**

**1. Dependency Injection in Program.cs**

builder.Host.UseSerilog((context, configuration) =>

configuration.ReadFrom.Configuration(context.Configuration));

builder.Services.AddEmployeeCafeApiDI(builder.Configuration);

builder.Services.AddProblemDetails();

builder.Services.AddExceptionHandler<AppExceptionHandler>();

**2. Middleware in Program.cs**

app.UseExceptionHandler();

app.UseSerilogRequestLogging();

**3. Entity Framework Configuration In Program.cs:**

builder.Services.AddEmployeeCafeApiDI(builder.Configuration);

**In DependencyInjection.cs:**

services.AddEmployeeCafeApplicaionDI()

.AddEmployeeCafeInfrastructureDI(configuration);

**Add the appsettings.json connection string:**

"ConnectionStrings": {

"DefaultConnection": "Server=localhost;Database= EmployeeCafeManagementDB;Trusted\_Connection=True;TrustServerCertificate=True;"

}

**🧪 Optional: Run Tests (xUnit)**

cd EmployeeCafeManagement.Tests

dotnet test