**SQLite-Net-PCL**

Example of use:

using SQLite;

public class User

{

[PrimaryKey, AutoIncrement]

public int Id { get; set; }

public string Name { get; set; }

}

dotnet add package sqlite-net-pcl

**Newtonsoft.Json**

using Newtonsoft.Json;

var json = JsonConvert.SerializeObject(myObject);

var object = JsonConvert.DeserializeObject<MyClass>(json);

dotnet add package Newtonsoft.Json

**RestSharp**

For example, to make a GET request:

var client = new RestClient("https://api.example.com");

var request = new RestRequest("users", Method.Get);

var response = await client. ExecuteAsync(request);

RestSharp makes working with APIs much cleaner and more readable.

dotnet add package RestSharp

**MediatR**

A basic example of how a command is defined and handled in MediatR would be:

public class CreateUserCommand : IRequest<int>

{

public string Name { get; set; }

}

public class CreateUserHandler : IRequestHandler<CreateUserCommand, int>

{

public Task<int> Handle(CreateUserCommand request, CancellationToken cancellationToken)

{

Logic for creating a user

return Task.FromResult(1); Returns the Id of the created user

}

}

dotnet add package MediatR.Extensions.Microsoft.DependencyInjection

**Polly**

For example, to retry an operation up to 3 times:

var policy = Policy

. Handle<HttpRequestException>()

. WaitAndRetryAsync(3, retryAttempt => TimeSpan.FromSeconds(retryAttempt));

await policy. ExecuteAsync(() => client. GetAsync("<https://api.example.com>"));

**FFImageLoading.Maui**

dotnet add package FFImageLoading.Maui

**AutoMapper**

var config = new MapperConfiguration(cfg = > {

cfg. CreateMap<Source, Destination>();

});

IMapper mapper = config. CreateMapper();

Destination dest = mapper. Map<Source, Destination>(sourceObj);

dotnet add package AutoMapper.Extensions.Microsoft.DependencyInjection