| | |
|---|---|
| **ENTITIES** | *class ContaCorrente*<br>*public string Idcontacorrente {get; set }*<br>*public int Numero {get; set;}*<br>*public string Nome {get; set }*<br>*public bool Ativo {get; set;}*<br>*public bool Check {get; set}*<br><br>*class Movimentacao : IRequest<bool>*<br>*{*<br>*public string Idmovimiento { get; set}*<br>*public string IdContacorrente { get; set}*<br>*public date Datamovimento { get; set}*<br>*public decimal Valor { get; set}*<br>*public TipoMovimentacao Tipo { get; set}*<br>*}* |
| **HANDLERS** | *public class MovimentacaoHandler :*<br>*IRequestHandler<MovimentacaoContaCommand, bool>*<br>*{*<br>*private readonly IDbConnection _connection;*<br><br>*public MovimentacaoContaHandler(IDbConnection connection)*<br>*{*<br>*_connection = connection;*<br>*}*<br><br>*public async Task<bool> Handle(MovimentacaoConta request,*<br>*CancellationToken cancellationToken)*<br>*{*<br>*var conta = await _connection. ----*<br><br>*if (conta == null)*<br>*{*<br>*throw new Exception("INVALID_ACCOUNT ");*<br>*}*<br>*}*<br><br>*public class ConsultaSaldoHandler :*<br>*IRequestHandler<ConsultaSaldoQuery, decimal>*<br>*{*<br>*private readonly IDbConnection _connection;*<br><br>*public ConsultaSaldoHandler(IDbConnection connection)*<br>*{*<br>*_connection = connection;*<br>*}*<br><br>*public async Task<decimal> Handle(ConsultaSaldoQuery request,*<br>*CancellationToken cancellationToken)*<br>*{*<br>*var conta = await _connection. ---- |

| | |
|---|---|
| | *if (conta == null)*<br>*{*<br>*throw new Exception("INVALID_ACCOUNT.");*<br>*}*<br>*return conta.Saldo;*<br>*}*<br>*}* |
| **SERVIÇO MEDIATOR SWAGGER** | *public void ConfigureServices(IServiceCollection services)*<br>*{*<br>*services.AddControllers();*<br>*services.AddSwaggerGen(c =>*<br>*{*<br>*c.SwaggerDoc("v1", new OpenApiInfo { Title = "ContaCorrenteApi", Version = "v1" });*<br>*});*<br>*services.AddSingleton<ContaCorrente>();*<br>*services.AddMediatR(Assembly.GetExecutingAssembly());*<br>*}* |
| **REQUEST HTTP** | *[ApiController]*<br>*[Route("[controller]")]*<br>*public class ContaCorrenteController : ControllerBase*<br>*{*<br>*private readonly IMediator mediator;*<br><br>*public ContaCorrenteController(IMediator mediator)*<br>*{*<br>*this.mediator = mediator;*<br>*}*<br><br>*[HttpPost("movimentacao")]*<br>*public async Task<IActionResult> MovimentacaoConta([FromBody] MovimentacaoContaCommand command)*<br>*{*<br>*var resultado = await mediator.Send(command);*<br>*return Ok(resultado);*<br>*}*<br><br>*[HttpGet("saldo")]*<br>*public IActionResult ConsultarSaldo([FromServices] ContaCorrente conta)*<br>*{*<br>*return Ok(conta.Saldo);*<br>*}*<br>*}* |
| **TESTES UNITÁRIOS COM MOQ - FLUENTASSERTIONS** | *public class ContaCorrenteTests*<br>*{*<br>*[Fact]*<br>*public async Task MovimentacaoConta_Saque ()* |