

ORM-Handler

Handler

1

```
func AmountGreaterThan1000(m *gdb.Model) *gdb.Model {
    return m.WhereGT("amount", 1000)
}

func PaidWithCreditCard(m *gdb.Model) *gdb.Model {
    return m.Where("pay_mode_sign", "credit_card")
}

func PaidWithCod(m *gdb.Model) *gdb.Model {
    return m.Where("pay_mode_sign", "cod")
}

func OrderStatus(statuses []string) func(m *gdb.Model) *gdb.Model {
    return func(m *gdb.Model) *gdb.Model {
        return m.Where("status", statuses)
    }
}

var (
    m = g.Model("product_order")
)

m.Handler(AmountGreaterThan1000, PaidWithCreditCard).Scan(&orders)
// SELECT * FROM `product_order` WHERE `amount`>1000 AND
// `pay_mode_sign`='credit_card'
// 1000

m.Handler(AmountGreaterThan1000, PaidWithCod).Scan(&orders)
// SELECT * FROM `product_order` WHERE `amount`>1000 AND
// `pay_mode_sign`='cod'
// 1000 COD

m.Handler(AmountGreaterThan1000, OrderStatus([]string{"paid", "shipped"})).
Scan(&orders)
// SELECT * FROM `product_order` WHERE `amount`>1000 AND `status` IN
// ('paid','shipped')
// 1000
```

Content Menu

- [1](#)
- [2](#)

2

```
func Paginate(r *ghttp.Request) func(m *gdb.Model) *gdb.Model {
    return func(m *gdb.Model) *gdb.Model {
        type Pagination struct {
            Page int
            Size int
        }
        var pagination Pagination
        _ = r.Parse(&pagination)
        switch {
        case pagination.Size > 100:
            pagination.Size = 100

        case pagination.Size <= 0:
            pagination.Size = 10
        }
        return m.Page(pagination.Page, pagination.Size)
    }
}

m.Handler(Paginate(r)).Scan(&users)
m.Handler(Paginate(r)).Scan(&articles)
```