

ORM-

ModelTXModelDBModelModelTXCommit/Rollback

[ORM](#)

1Transaction

gdbTransaction

```
func (db DB) Transaction(ctx context.Context, f func(ctx context.Context,
tx TX) error) (err error)
```

errornilCommitRollback

 panic

```
func Register() error {
    return g.DB().Transaction(ctx, func(ctx context.Context, tx gdb.
TX) error {
        var (
            result sql.Result
            err    error
        )
        //
        result, err = tx.Table("user").Insert(g.Map{
            "name": "john",
            "score": 100,
            //...
        })
        if err != nil {
            return err
        }
        // uid
        result, err = tx.Table("user_detail").Insert(g.Map{
            "uid":    result.LastInsertId(),
            "phone": "18010576258",
            //...
        })
        return err
    })
}
```

Content Menu

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

2TX

TX

```

func Register() error {
    var (
        uid int64
        err error
    )
    tx, err := g.DB().Begin()
    if err != nil {
        return err
    }
    //
    // tx.Commit(),
    // tx.Rollback()
    defer func() {
        if err != nil {
            tx.Rollback()
        } else {
            tx.Commit()
        }
    }()
    //
    uid, err = AddUserInfo(tx, g.Map{
        "name": "john",
        "score": 100,
        //...
    })
    if err != nil {
        return err
    }
    // uid
    err = AddUserDetail(tx, g.Map{
        "uid": uid,
        "phone": "18010576259",
        //...
    })
    return err
}

func AddUserInfo(tx gdb.TX, data g.Map) (int64, error) {
    result, err := g.Model("user").TX(tx).Data(data).Insert()
    if err != nil {
        return 0, err
    }
    uid, err := result.LastInsertId()
    if err != nil {
        return 0, err
    }
    return uid, nil
}

func AddUserDetail(tx gdb.TX, data g.Map) error {
    _, err := g.Model("user_detail").TX(tx).Data(data).Insert()
    return err
}

```