

# ORM-

GoFrame ORMcontextORMTransaction Save PointTransaction

## SQL

SQLidname

```
CREATE TABLE `user` (  
  `id` int(10) unsigned NOT NULL COMMENT 'ID',  
  `name` varchar(45) NOT NULL COMMENT '',  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

### Content Menu

- [SQL](#)
- - [1db.BeginTx.Begin](#)
  - [2](#)
- [\(\)](#)
- [SavePoint/RollbackTo](#)
- - [controller](#)
  - [service](#)
  - [dao](#)

```
db := g.DB()  
  
tx, err := db.Begin()  
if err != nil {  
    panic(err)  
}  
if err = tx.Begin(); err != nil {  
    panic(err)  
}  
_, err = tx.Model(table).Data(g.Map{"id": 1, "name": "john"}).Insert()  
if err = tx.Rollback(); err != nil {  
    panic(err)  
}  
_, err = tx.Model(table).Data(g.Map{"id": 2, "name": "smith"}).Insert()  
if err = tx.Commit(); err != nil {  
    panic(err)  
}
```

## 1db.BeginTx.Begin

db.BeginTx.Begindb.BeginTxctxtx.BeginSavePointtransactionNNSAVEPOINT  
`transaction1`20

## 2

goframeORMSQL

```
2021-05-22 21:12:10.776 [DEBU] [ 4 ms] [default] [txid:1] BEGIN  
2021-05-22 21:12:10.776 [DEBU] [ 0 ms] [default] [txid:1] SAVEPOINT  
`transaction0`  
2021-05-22 21:12:10.789 [DEBU] [ 13 ms] [default] [txid:1] SHOW FULL  
COLUMNS FROM `user`  
2021-05-22 21:12:10.790 [DEBU] [ 1 ms] [default] [txid:1] INSERT INTO  
`user`(`id`,`name`) VALUES(1,'john')  
2021-05-22 21:12:10.791 [DEBU] [ 1 ms] [default] [txid:1] ROLLBACK TO  
SAVEPOINT `transaction0`  
2021-05-22 21:12:10.791 [DEBU] [ 0 ms] [default] [txid:1] INSERT INTO  
`user`(`id`,`name`) VALUES(2,'smith')  
2021-05-22 21:12:10.792 [DEBU] [ 1 ms] [default] [txid:1] COMMIT
```

[txid:1]ORMIDIDID

```
mysql> select * from `user`;
+-----+-----+
| id | name |
+-----+-----+
| 2 | smith |
+-----+-----+
1 row in set (0.00 sec)
```

()

Transaction

```
db.Transaction(ctx, func(ctx context.Context, tx gdb.Tx) error {
    // Nested transaction 1.
    if err := tx.Transaction(ctx, func(ctx context.Context, tx gdb.Tx)
error {
    _, err := tx.Model(table).Ctx(ctx).Data(g.Map{"id": 1,
"name": "john"}).Insert()
    return err
    }); err != nil {
    return err
    }
    // Nested transaction 2, panic.
    if err := tx.Transaction(ctx, func(ctx context.Context, tx gdb.Tx)
error {
    _, err := tx.Model(table).Ctx(ctx).Data(g.Map{"id": 2,
"name": "smith"}).Insert()
    // Create a panic that can make this transaction rollback
    automatically.
    panic("error")
    return err
    }); err != nil {
    return err
    }
    return nil
})
```

txdbdaotctx

```

db.Transaction(ctx, func(ctx context.Context, tx gdb.Tx) error {
    // Nested transaction 1.
    if err := db.Transaction(ctx, func(ctx context.Context, tx gdb.Tx)
error {
    _, err := db.Model(table).Ctx(ctx).Data(g.Map{"id": 1,
"name": "john"}).Insert()
    return err
    }); err != nil {
    return err
    }
    // Nested transaction 2, panic.
    if err := db.Transaction(ctx, func(ctx context.Context, tx gdb.Tx)
error {
    _, err := db.Model(table).Ctx(ctx).Data(g.Map{"id": 2,
"name": "smith"}).Insert()
    // Create a panic that can make this transaction rollback
automatically.
    panic("error")
    return err
    }); err != nil {
    return err
    }
    return nil
})

```

## SQL

```

2021-05-22 21:18:46.672 [DEBU] [ 2 ms] [default] [txid:1] BEGIN
2021-05-22 21:18:46.672 [DEBU] [ 0 ms] [default] [txid:1] SAVEPOINT
`transaction0`
2021-05-22 21:18:46.673 [DEBU] [ 0 ms] [default] [txid:1] SHOW FULL
COLUMNS FROM `user`
2021-05-22 21:18:46.674 [DEBU] [ 0 ms] [default] [txid:1] INSERT INTO
`user`(`id`,`name`) VALUES(1,'john')
2021-05-22 21:18:46.674 [DEBU] [ 0 ms] [default] [txid:1] RELEASE
SAVEPOINT `transaction0`
2021-05-22 21:18:46.675 [DEBU] [ 1 ms] [default] [txid:1] SAVEPOINT
`transaction0`
2021-05-22 21:18:46.675 [DEBU] [ 0 ms] [default] [txid:1] INSERT INTO
`user`(`name`,`id`) VALUES('smith',2)
2021-05-22 21:18:46.675 [DEBU] [ 0 ms] [default] [txid:1] ROLLBACK TO
SAVEPOINT `transaction0`
2021-05-22 21:18:46.676 [DEBU] [ 1 ms] [default] [txid:1] ROLLBACK

```



ctx

```
db.Transaction(ctx, func(ctx context.Context, tx gdb.Tx) error {
    // Nested transaction 1.
    if err := db.Transaction(ctx, func(ctx context.Context, tx
gdb.Tx) error {
        _, err := db.Model(table).Ctx(ctx).Data(g.Map
{"id": 1, "name": "john"}).Insert()
        return err
    }); err != nil {
        return err
    }
    // Nested transaction 2, panic.
    if err := db.Transaction(ctx, func(ctx context.Context, tx
gdb.Tx) error {
        _, err := db.Model(table).Data(g.Map{"id": 2,
"name": "smith"}).Insert()
        // Create a panic that can make this transaction
rollback automatically.
        panic("error")
        return err
    }); err != nil {
        return err
    }
    return nil
})
```

SQL

```
2021-05-22 21:29:38.841 [DEBU] [ 3 ms] [default] [txid:1] BEGIN
2021-05-22 21:29:38.842 [DEBU] [ 1 ms] [default] [txid:1]
SAVEPOINT `transaction0`
2021-05-22 21:29:38.843 [DEBU] [ 1 ms] [default] [txid:1] SHOW
FULL COLUMNS FROM `user`
2021-05-22 21:29:38.845 [DEBU] [ 2 ms] [default] [txid:1] INSERT
INTO `user`(`id`,`name`) VALUES(1,'john')
2021-05-22 21:29:38.845 [DEBU] [ 0 ms] [default] [txid:1] RELEASE
SAVEPOINT `transaction0`
2021-05-22 21:29:38.846 [DEBU] [ 1 ms] [default] [txid:1]
SAVEPOINT `transaction0`
2021-05-22 21:29:38.847 [DEBU] [ 1 ms] [default] INSERT INTO
`user`(`id`,`name`) VALUES(2,'smith')
2021-05-22 21:29:38.848 [DEBU] [ 0 ms] [default] [txid:1]
ROLLBACK TO SAVEPOINT `transaction0`
2021-05-22 21:29:38.848 [DEBU] [ 0 ms] [default] [txid:1] ROLLBACK
```

INSERTINSERT INTO `user`(`id`,`name`) VALUES(2,'smith') ID

## SavePoint/RollbackTo

Transaction Save PointSavePointPoint

```

tx, err := db.Begin()
if err != nil {
    panic(err)
}
defer func() {
    if err := recover(); err != nil {
        _ = tx.Rollback()
    }
}()
if _, err = tx.Model(table).Data(g.Map{"id": 1, "name": "john"}).Insert();
err != nil {
    panic(err)
}
if err = tx.SavePoint("MyPoint"); err != nil {
    panic(err)
}
if _, err = tx.Model(table).Data(g.Map{"id": 2, "name": "smith"}).Insert();
err != nil {
    panic(err)
}
if _, err = tx.Model(table).Data(g.Map{"id": 3, "name": "green"}).Insert();
err != nil {
    panic(err)
}
if err = tx.RollbackTo("MyPoint"); err != nil {
    panic(err)
}
if err = tx.Commit(); err != nil {
    panic(err)
}

```

## SQL

```

2021-05-22 21:38:51.992 [DEBU] [ 3 ms] [default] [txid:1] BEGIN
2021-05-22 21:38:52.002 [DEBU] [ 9 ms] [default] [txid:1] SHOW FULL
COLUMNS FROM `user`
2021-05-22 21:38:52.002 [DEBU] [ 0 ms] [default] [txid:1] INSERT INTO
`user`(`id`,`name`) VALUES(1,'john')
2021-05-22 21:38:52.003 [DEBU] [ 1 ms] [default] [txid:1] SAVEPOINT
`MyPoint`
2021-05-22 21:38:52.004 [DEBU] [ 1 ms] [default] [txid:1] INSERT INTO
`user`(`id`,`name`) VALUES(2,'smith')
2021-05-22 21:38:52.005 [DEBU] [ 1 ms] [default] [txid:1] INSERT INTO
`user`(`id`,`name`) VALUES(3,'green')
2021-05-22 21:38:52.006 [DEBU] [ 0 ms] [default] [txid:1] ROLLBACK TO
SAVEPOINT `MyPoint`
2021-05-22 21:38:52.006 [DEBU] [ 0 ms] [default] [txid:1] COMMIT

```

```

mysql> select * from `user`;
+----+-----+
| id | name |
+----+-----+
| 1  | john |
+----+-----+
1 row in set (0.00 sec)

```

InsertSavePointMyPointRollbackToInsert

(user)(user\_detail)dao

goframeapi-service-dao

## controller

```
// HTTP
func (*cUser) Signup(r *ghttp.Request) {
    // ....
    service.User().Signup(r.Context(), userServiceSignupReq)
    // ...
}
```

HTTPContext

## service

```
//
func (*userService) Signup(ctx context.Context, r *model.
UserServiceSignupReq) {
    // ....
    dao.User.Transaction(ctx, func(ctx context.Context, tx gdb.Tx)
error {
    err := dao.User.Ctx(ctx).Save(r.UserInfo)
    if err != nil {
        return err
    }
    err := dao.UserDetail.Ctx(ctx).Save(r.UserDetail)
    if err != nil {
        return err
    }
    return nil
})
    // ...
}
```

userServiceDetailCtxserviceCtx

```
func (*userService) Signup(ctx context.Context, r *model.
UserServiceSignupReq) {
    // ....
    dao.User.Transaction(ctx, func(ctx context.Context, tx gdb.Tx)
(err error) {
    if err = dao.User.Ctx(ctx).Save(r.UserInfo); err != nil {
        return err
    }
    if err = dao.UserDetail.Ctx(ctx).Save(r.UserDetail); err !
= nil {
        return err
    }
    if err = service.XXXA().Call(ctx, ...); err != nil {
        return err
    }
    if err = service.XXXB().Call(ctx, ...); err != nil {
        return err
    }
    if err = service.XXXC().Call(ctx, ...); err != nil {
        return err
    }
    // ...
    return nil
})
    // ...
}
```

## dao

daogoframe cli