

demo

封装 DB

参照 [📖 xfx学习](#)，我们可以知道在 newBackingServices 实例化 MySQLDB，所以在这之后我们用 Db 取出这个实例化的 MySQLDB 供我们调用

Go

```
1 type ModuleContext struct {
2     AppContext xfx.App
3     *backingServices
4     Db *dao.MySQLDB
5 }
6
7 func (c *ModuleContext) Init(app xfx.App) {
8     c.AppContext = app
9     c.backingServices = newBackingServices(c)
10    c.Db = c.backingServices.MySQLDB
11 }
```

DB 示例

创建 DB 实例

Go

```
1 func NewCreateDbService(moduleCtx *ctx.ModuleContext) CreateDbFunc {
2     return func(ctx context.Context, request *CreateDbRequest) (*Create
    DbResponse, error) {
3         err := validator.Validate(request)
4         if err != nil {
5             return nil, err
6         }
7         database := fmt.Sprintf("ecis_%s", request.Dbname)
8
9         // 创建数据库的 SQL 语句
10        createSQL := fmt.Sprintf("CREATE DATABASE IF NOT EXISTS %s;", d
    atabase)
11
12        // 执行 SQL 语句创建数据库
13        _, err = moduleCtx.Db.Exec(ctx, createSQL)
14        if err != nil {
15            return nil, err
16        }
17        out := NewCreateDbResponse()
18        return &out, nil
19    }
20 }
21
```

create_db

POST http://{{test_addr}}/api/db/v1/createDb

Params Authorization Headers (9) Body Pre-request Script Tests

none form-data x-www-form-urlencoded

```
1 {
2   "dbname": "std"
3 }
```

Database

```
db_tool
ecis_std
information_schema
mysql
performance_schema
std
sys
test
```

8 rows in set (0.00 sec)

```
mysql> show databases;
```

Database

```
db_tool
ecis_std
information_schema
mysql
performance_schema
std
sys
test
```

8 rows in set (0.00 sec)

mysql>

Body Cookies Headers (3) Test Results

Pretty Raw Preview JSON

```
1 {
2   "code": 0,
3   "msg": ""
4 }
```

删除DB

Go

```
1
2 func NewDeleteDbService(moduleCtx *ctx.ModuleContext) DeleteDbFunc {
3     return func(ctx context.Context, request *DeleteDbRequest) (*Delete
    DbResponse, error) {
4         err := validator.Validate(request)
5         if err != nil {
6             return nil, err
7         }
8         database := fmt.Sprintf("ecis_%s", request.Dbname)
9         //database := request.Dbname
10        deleteSQL := fmt.Sprintf("DROP DATABASE IF EXISTS %s;", databas
    e)
11
12        _, err = moduleCtx.Db.Exec(ctx, deleteSQL)
13        if err != nil {
14            return nil, err
15        }
16        out := NewDeleteDbResponse()
17        return &out, nil
18    }
19 }
20
```

delete_db

POST http://{{test_addr}}/api/db/v1/deleteDb

Params Authorization Headers (9) Body ● Pre-request Script Tests

● none ● form-data ● x-www-form-urlencoded

1 {
2 "dbname": "std"
3 }

Body Cookies Headers (3) Test Results

Pretty Raw Preview JSON

1 {
2 "code": 0,
3 "msg": ""
4 }

命令提示符 - mysql -uroot -p

```
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| db_tool |  
| ecis_std |  
| information_schema |  
| mysql |  
| performance_schema |  
| std |  
| sys |  
| test |  
+-----+  
8 rows in set (0.00 sec)  
  
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| db_tool |  
| information_schema |  
| mysql |  
| performance_schema |  
| std |  
| sys |  
| test |  
+-----+  
7 rows in set (0.00 sec)  
  
mysql>
```

查询 DB

Go

```
1 func NewSelectDbService(moduleCtx *ctx.ModuleContext) SelectDbFunc {
2     return func(ctx context.Context, request *SelectDbRequest) (*Select
    DbResponse, error) {
3         err := validator.Validate(request)
4         if err != nil {
5             return nil, err
6         }
7         database := fmt.Sprintf("ecis_%s", request.Dbname)
8         showTablesSQL := fmt.Sprintf("SHOW TABLES FROM %s", database)
9         // 执行 SQL 语句查询数据库列表
10        rows, err := moduleCtx.Db.Query(ctx, showTablesSQL)
11        if err != nil {
12            return nil, err
13        }
14        defer rows.Close()
15        out := NewSelectDbResponse()
16        for rows.Next() {
17            var tableName string
18            if err := rows.Scan(&tableName); err != nil {
19                return nil, err
20            }
21            out.Data.TableNames = append(out.Data.TableNames, tableName)
22        }
23        if err != nil {
24            return nil, err
25        }
26        out.Data.DbName = request.Dbname
27        return &out, nil
28    }
29 }
```

POST

http://{{test_addr}}/api/db/v1/showDb

none

form-data

x-www-form-urlencoded

raw

binary

JSON

1 {

2 "dbname":"db_tool"

3 }

Body

Cookies

Headers (3)

Test Results

Pretty

Raw

Preview

JSON

1 {

2 "code": 0,

3 "data": {

4 "dbName": "db_tool",

5 "tableNames": [

6 "tb_account",

7 "tb_schema_checkpoint_base_info",

8 "tb_schema_hook_task_info",

9 "tb_schema_opt_log",

10 "tb_schema_table_core_info",

11 "tb_schema_table_ext_info",

12 "tb_schema_table_partition_info"

13]

14 },

15 "msg": ""

16 }

查询 DB 实例表结构


```
34         fmt.Println(tableName)
35         if tableName == "" {
36             continue
37         }
38         query := fmt.Sprintf("SHOW CREATE TABLE `%s`.`%s`;", db
Name, tableName)
39         row, _ := moduleCtx.Db.QueryRow(ctx, query)
40         var tableInfo schemas.TableInfo
41         err = row.Scan(&tableInfo.Name, &tableInfo.Ddl)
42         if err != nil {
43             return nil, err
44         }
45         tableInfo.Name = tableName
46         out.Data.Total += 1
47         out.Data.Items = append(out.Data.Items, tableInfo)
48     }
49 }
50 return &out, nil
51 }
52 }
53
```

selectDbTables

GET

http://{{test_addr}}/api/db/v1/selectDbTables?dbname=db_tool

Params

Authorization

Headers (10)

Body

Pre-request Script

Tests

Query Params

	KEY	VALUE
<input checked="" type="checkbox"/>	dbname	db_tool
<input type="checkbox"/>	tablename	tb_schema_table_core_info
	Key	Value

Body

Cookies

Headers (3)

Test Results

Pretty

Raw

Preview

JSON

1

{

2

"code": 0,

3

"data": {

4

"items": [

5

{

6

"ddl": "CREATE TABLE `tb_account` (\n `created_at` bigint NOT NULL,\n `deleted_at` bigint NOT NULL DEFAULT

7

`login_name` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL,\n `password` varch

8

`services` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL,\n `updated_at` bigint

9

DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci",

10

"extra": "",

11

"name": "tb_account"

12

},

13

{

14

"ddl": "CREATE TABLE `tb_schema_checkpoint_base_info` (\n `id` bigint NOT NULL AUTO_INCREMENT,\n `check_poi

15

NULL DEFAULT '',\n `service` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci DEFAULT NULL,

16

utf8mb4_unicode_ci NOT NULL DEFAULT '',\n `base_check_point` varchar(255) CHARACTER SET utf8mb4 COLLATE

17

CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL DEFAULT '',\n `db_name` varchar(255) CHARACTER

18

`wps_yundoc_mini`,\n `alias` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL DEFA

19

utf8mb4_unicode_ci NOT NULL DEFAULT '',\n `version_name` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8

20

DEFAULT '0',\n `updated_at` bigint NOT NULL DEFAULT '0',\n `deleted_at` bigint NOT NULL DEFAULT '0',\n

21

DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci ROW_FORMAT=DYNAMIC",

22

"extra": "",

23

"name": "tb_schema_checkpoint_base_info"

24

}

25

]

26

}

27

}

用户相关

创建 USER

Go

```
1 func NewCreateUserService(moduleCtx *ctx.ModuleContext) CreateUserFunc
2 {
3     return func(ctx context.Context, request *CreateUserRequest) (*CreateUserResponse, error) {
4         err := validator.Validate(request)
5         if err != nil {
6             return nil, err
7         }
8         userStr := fmt.Sprintf("%s_%s", "ecis", request.UserId)
9         row, err := moduleCtx.Db.Query(ctx, "SELECT User, Host from mysql.user where User=?", userStr)
10        if err != nil {
11            return nil, err
12        }
13        isExist := false
14        if row.Next() {
15            isExist = true
16        }
17        out := NewCreateUserResponse()
18        if isExist {
19            out.Msg = "User is exist"
20            return &out, nil
21        }
22        // 创建新用户
23        create_sql := fmt.Sprintf("CREATE USER '%s'@'#' IDENTIFIED BY '%s';", userStr, request.UserPasswd)
24        _, err = moduleCtx.Db.Exec(ctx, strings.Replace(create_sql, "#", "%", -1))
25        if err != nil {
26            return nil, err
27        }
28        return &out, nil
29    }
30 }
31
```

► createUser

POST

http://{{test_addr}}/api/user/v1/createUser

Params

Authorization

Headers (9)

Body ●

Pre-request Script

Tests

● none

● form-data

● x-www-form-urlencoded

```
1 {  
2   "user_id": "whl",  
3   "user_passwd": "123123"  
4 }
```

命令提示符 - mysql -uroot -p

```
mysql> show users  
-> ;  
ERROR 1064 (42000): You have an error in your SQL  
statement; check the manual that corresponds to your MySQL  
server version for the right syntax to use near 'users' at line 1  
mysql> SELECT user FROM mysql.user;  
+-----+  
| user |  
+-----+  
| ecis_whl |  
| mysql.infoschema |  
| mysql.session |  
| mysql.sys |  
| root |  
+-----+  
5 rows in set (0.00 sec)  
  
mysql> SELECT user FROM mysql.user;  
+-----+  
| user |  
+-----+  
| ecis_whl |  
| mysql.infoschema |  
| mysql.session |  
| mysql.sys |  
| root |  
+-----+  
5 rows in set (0.00 sec)  
  
mysql>
```

Body

Cookies

Headers (3)

Test Results

Pretty

Raw

Preview

JSON ▼

```
1 {  
2   "code": 0,  
3   "msg": ""  
4 }
```

createUser

POST http://{{test_addr}}/api/user/v1/createUser

Params Authorization Headers (9) Body ● Pre-request Script Tests

● none ● form-data ● x-www-form-urlencoded

```
1 {
2   "user_id": "whl",
3   "user_passwd": "123123"
4 }
```

Body Cookies Headers (3) Test Results

Pretty Raw Preview JSON ▼

```
1 {
2   "code": 0,
3   "msg": "User is exist"
4 }
```

命令提示符 - mysql -uroot -p

```
ecis_whl
mysql.infoschema
mysql.session
mysql.sys
root
+-----+
5 rows in set (0.00 sec)

mysql> SELECT      user FROM      mysql.user;
+-----+
| user |
+-----+
| ecis_whl |
| mysql.infoschema |
| mysql.session |
| mysql.sys |
| root |
+-----+
5 rows in set (0.00 sec)

mysql>
```

删除USER

Go

```
1
2 func NewDeleteUserService(moduleCtx *ctx.ModuleContext) DeleteUserFunc
3 {
4     return func(ctx context.Context, request *DeleteUserRequest) (*DeleteUserResponse, error) {
5         err := validator.Validate(request)
6         if err != nil {
7             return nil, err
8         }
9         // 验证用户密码
10        if !isValidUser(request.UserId, request.UserPasswd) {
11            return nil, fmt.Errorf("invalid credentials, user %s is not authorized to delete users", request.UserId)
12        }
13        // 拼接要删除的用户名
14        userStr := fmt.Sprintf("%s_%s", "ecis", request.UserId)
15
16        // 查询该用户是否存在
17        rows, err := moduleCtx.Db.Query(ctx, "SELECT User, Host from mysql.user where User=?", userStr)
18        if err != nil {
19            return nil, err
20        }
21
22        if !rows.Next() {
23            out := NewDeleteUserResponse()
24            return &out, nil
25        }
26
27        // 删除用户
28        delSQL := fmt.Sprintf("DROP USER '%s';", userStr)
29        _, err = moduleCtx.Db.Exec(ctx, delSQL)
30        if err != nil {
31            return nil, err
32        }
33        // 刷新MySQL权限
34        _, err = moduleCtx.Db.Exec(ctx, "FLUSH PRIVILEGES")
```

```
34         if err != nil {
35             return nil, err
36         }
37         out := NewDeleteUserResponse()
38         return &out, nil
39     }
40 }
41
42 // isValidUser 校验用户的用户名和密码是否有效
43 func isValidUser(username, password string) bool {
44
45     // 检查用户名和密码是否匹配、是否具有删除权限等
46
47     return true
48 }
```

deleteUser

POST http://{{test_addr}}/api/user/v1/deleteUser

Params Authorization Headers (9) Body Pre-request Script Tests

none form-data x-www-form-urlencoded raw

```
1 {
2   "user_id": "whl",
3   "user_passwd": "123123"
4 }
```

Body Cookies Headers (3) Test Results

Pretty Raw Preview JSON

```
1 {
2   "code": 0,
3   "msg": ""
4 }
```

命令提示符 - mysql -uroot -p

```
mysql.session
mysql.sys
root
4 rows in set (0.00 sec)

mysql> SELECT user FROM mysql.user;
+-----+
| user |
+-----+
| ecis_whl |
| mysql.infoschema |
| mysql.session |
| mysql.sys |
| root |
+-----+
5 rows in set (0.00 sec)

mysql> SELECT user FROM mysql.user;
+-----+
| user |
+-----+
| mysql.infoschema |
| mysql.session |
| mysql.sys |
| root |
+-----+
4 rows in set (0.00 sec)

mysql>
```

授权（绑定对应 DB 实例）

Go

```
1 func NewGrantService(moduleCtx *ctx.ModuleContext) GrantFunc {
2     return func(ctx context.Context, request *GrantRequest) (*GrantResponse, error) {
3         err := validator.Validate(request)
4         if err != nil {
5             return nil, err
6         }
7         prisql := ""
8         for _, pri := range request.Privileges {
9             prisql += pri
10            prisql += ","
11        }
12        prisql = rmLast(prisql, ',')
13
14        userStr := fmt.Sprintf("%s_%s", "ecis", request.UserId)
15        grant_sql := fmt.Sprintf("GRANT %s on %s.* to %s;", prisql, request.Dbname, userStr)
16        fmt.Println(grant_sql)
17        _, err = moduleCtx.Db.Exec(ctx, grant_sql)
18        if err != nil {
19            return nil, err
20        }
21        // 刷新MySQL权限
22        _, err = moduleCtx.Db.Exec(ctx, "FLUSH PRIVILEGES")
23        if err != nil {
24            return nil, err
25        }
26        out := NewGrantResponse()
27        return &out, nil
28    }
29 }
30 func rmLast(source string, rm uint8) string {
31     for i := len(source) - 1; i >= 0; i-- {
32         if source[i] == rm {
33             if i == len(source)-1 {
34                 return source[:i]
35             } else {
```

```

36         return source[:i] + source[i+1:]
37     }
38 }
39 }
40 return source
41 }

```

grant

POST http://{{test_addr}}/api/user/v1/grant

Params Authorization Headers (9) Body Pre-request Script Tests

none form-data x-www-form-urlencoded raw

```

1 {
2   "dbname": "db_tool",
3   "privilegeType": 0,
4   "privileges": [
5     "delete",
6     "update"
7   ],
8   "table_name": "string",
9   "user_passwd": "whl",
10  "user_id": "whl"
11 }

```

Body Cookies Headers (3) Test Results

Pretty Raw Preview JSON

```

1 {
2   "code": 0,
3   "msg": ""
4 }

```

命令提示符 - mysql -uroot -p

```

mysql> GRANT create on ecis_whl.* to db_tool;
ERROR 1410 (42000): You are not allowed to create a user with GRANT
mysql> SHOW GRANTS FOR ecis_whl;

+-----+
| Grants for ecis_whl@% |
+-----+
| GRANT USAGE ON *.* TO `ecis_whl`@`%` |
| GRANT CREATE ON `db_tool`.* TO `ecis_whl`@`%` |
+-----+
2 rows in set (0.00 sec)

mysql> SHOW GRANTS FOR ecis_whl;

+-----+
| Grants for ecis_whl@% |
+-----+
| GRANT USAGE ON *.* TO `ecis_whl`@`%` |
| GRANT CREATE ON `db_tool`.* TO `ecis_whl`@`%` |
+-----+
2 rows in set (0.00 sec)

mysql> SHOW GRANTS FOR ecis_whl;

+-----+
| Grants for ecis_whl@% |
+-----+
| GRANT USAGE ON *.* TO `ecis_whl`@`%` |
| GRANT UPDATE, DELETE, CREATE ON `db_tool`.* TO `ecis_whl`@`%` |
+-----+
2 rows in set (0.00 sec)

mysql>

```

sql 相关

获取表结构(类似 show table)

Go

[illegible]

```

31             filteredSqlsDstCoreInfosSet = append(fi
lteredSqlsDstCoreInfosSet, coreInfo)
32         }
33     }
34     sqlsDstCoreInfosSet = filteredSqlsDstCoreInfosS
et
35     case AlterColumn:
36         for _, column := range stmtItem.CoreInfos {
37             for _, coreInfo := range sqlsDstCoreInfosSe
t {
38                 if *coreInfo.TableName == *column.Table
Name && *coreInfo.ColumnName == *column.ColumnName {
39                     coreInfo.ColumnType = column.Column
Type
40                     break
41                 }
42             }
43         }
44     case AddConstraint:
45         sqlsDstExtInfosSet = append(sqlsDstExtInfosSe
t, stmtItem.ExtInfos...)
46     case DropIndex:
47         var filteredSqlsDstExtInfosSet []*dao.TableExtI
nfoSet
48
49         // 遍历 sqlsDstCoreInfosSet , 并检查每个元素是否与 s
tmtItem.CoreInfos 中的任意元素匹配
50         for _, extInfo := range sqlsDstExtInfosSet {
51             found := false
52             for _, option := range stmtItem.ExtInfos {
53                 if extInfo.TableName == option.TableNam
e && extInfo.ItemType == option.ItemType && extInfo.ItemName == option.
ItemName {
54                     found = true
55                     break
56                 }
57             }
58             // 如果没有找到匹配项, 则将该元素添加到过滤后的切片
中
59             if !found {

```

```

60             filteredSqlsDstExtInfosSet = append(fil
teredSqlsDstExtInfosSet, extInfo)
61         }
62     }
63     sqlsDstExtInfosSet = filteredSqlsDstExtInfosSet
64     case RenameIndex:
65         for _, extInfo := range sqlsDstExtInfosSet {
66             for _, option := range stmtItem.ExtInfos {
67                 if extInfo.TableName == option.TableNam
e && extInfo.ItemType == option.ItemType && extInfo.ItemName == option.
ItemName {
68                     extInfo.ItemName = option.ItemName
69                     break
70                 }
71             }
72         }
73     case RenameTable:
74         oldName := ""
75         newName := ""
76         for _, option := range stmtItem.ExtInfos {
77             if option.OldTableName != nil && len(*opti
on.OldTableName) > 0 {
78                 oldName = *option.OldTableName
79                 newName = *option.TableName
80                 break
81             }
82         }
83         for _, coreInfo := range sqlsDstCoreInfosSet {
84             if *coreInfo.TableName == oldName {
85                 *coreInfo.TableName = newName
86                 *coreInfo.OldTableName = oldName
87             }
88         }
89         for _, extInfo := range sqlsDstExtInfosSet {
90             if *extInfo.TableName == oldName {
91                 *extInfo.TableName = newName
92                 *extInfo.OldTableName = oldName
93             }
94         }
95     default:

```

```

96             el.Error("unsupport alter type.", el.String("type",
fmt.Sprintf("%d", alterType)))
97         }
98     }
99 }
100 }
101     tableInfos := make([]business.TableInfo, 0)
102     sqlsDstCoreInfos := business.TableCoreInfoSetToTableCoreInfo(sqlsDstCoreInfosSet)
103     sqlsDstExtInfos := business.TableExtInfoSetToTableExtInfo(sqlsDstExtInfosSet)
104     anss, _ := business.GetUpgradeSqls(tableInfos, sqlsDstCoreInfos, sqlsDstExtInfos)
105     out.Data.Total = 0
106     for _, sql := range anss {
107         // p := parser.New()
108         // stmtNodes, _, _ := p.ParseSQL(sql)
109         // if len(stmtNodes) == 0 {
110
111             //     fmt.Println(sql)
112             //     fmt.Println(sql)
113             //     fmt.Println(sql)
114             // }
115             // var stmt ast.StmtNode
116             // stmt = stmtNodes[0]
117             // createStmt, _ := stmt.(*ast.CreateTableStmt)
118             // tableName := createStmt.Table.Name.String()
119             pattern := `(?(i)CREATE\s+TABLE\s+(?:IF\s+NOT\s+EXISTS\s+)?
\s+ + "`([^\`]+)`" + "\s+(\`
120
121             re := regexp.MustCompile(pattern)
122             match := re.FindStringSubmatch(sql)
123             tableName := ""
124             if len(match) > 0 {
125                 tableName = match[1]
126             } else {
127                 fmt.Println("No match found.")
128             }
129             tableInfo := schemas.TableInfo{
130                 Name: tableName,

```

```

131         Ddl: sql,
132     }
133     out.Data.Items = append(out.Data.Items, tableInfo)
134     out.Data.Total++
135 }
136
137     return &out, nil
138 }
139 }

```

POST http://(test_addr)/api/dbversion/v1/genTables

none form-data x-www-form-urlencoded raw binary JSON (application/json)

```

1 {
2   "dbname": "db_tool",
3   "sql_content": "CREATE TABLE IF NOT EXISTS student (id INT PRIMARY KEY AUTO_INCREMENT, first_name VARCHAR(255) NOT NULL, last_name VARCHAR(255) NOT NULL, age INT, gender VARCHAR(10), address VARCHAR
4     (500), email VARCHAR(255), phone VARCHAR(20), created_at DATETIME DEFAULT CURRENT_TIMESTAMP, updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP);
5   ALTER TABLE student ADD COLUMN date_of_birth DATE;
6   ALTER TABLE student MODIFY COLUMN age TINYINT;
7   ALTER TABLE student DROP COLUMN address;"
8 }

```

Body Cookies Headers (3) Test Results Status: 200 OK Time: 15 ms Size: 614 B Save Download

Pretty Raw Preview JSON

```

1 {
2   "code": 0,
3   "data": {
4     "items": [
5       {
6         "ddl": "CREATE TABLE `student` (\n  `id` INT AUTO_INCREMENT,\n  `first_name` VARCHAR(255) NOT NULL,\n  `last_name` VARCHAR(255) NOT NULL,\n  `age` TINYINT,\n  `gender`
7           VARCHAR(10),\n  `email` VARCHAR(255),\n  `phone` VARCHAR(20),\n  `created_at` DATETIME DEFAULT CURRENT_TIMESTAMP,\n  `updated_at` DATETIME DEFAULT CURRENT_TIMESTAMP ON
8             UPDATE CURRENT_TIMESTAMP,\n  `date_of_birth` DATE,\n);",
9         "extra": "",
10        "name": "student"
11      }
12    ],
13    "total": 1
14  },
15  "msg": ""
16 }

```

对比表结构

Go

```
1 func NewCompareTableStructService(moduleCtx *ctx.ModuleContext) Compare
   TableStructFunc {
2     return func(ctx context.Context, request *CompareTableStructReques
      t) (*CompareTableStructResponse, error) {
3         err := validator.Validate(request)
4         if err != nil {
5             return nil, err
6         }
7         sqlsSrcCoreInfosSet []*dao.TableCoreInfoSet
8         sqlsSrcExtInfosSet []*dao.TableExtInfoSet
9         sqlsDstCoreInfosSet []*dao.TableCoreInfoSet
10        sqlsDstExtInfosSet []*dao.TableExtInfoSet
11        sqlsSrc := business.ExtractDDLStatements(request.SqlContentSrc)
12        sqlsDst := business.ExtractDDLStatements(request.SqlContentDst)
13        for _, sql := range sqlsSrc {
14            stmtItems, _, _ := business.ParseDDL(sql, request.Dbname)
15            for _, stmtItem := range stmtItems {
16                sqlsSrcCoreInfosSet = append(sqlsSrcCoreInfosSet, stmtI
              tem.CoreInfos...)
17                sqlsSrcExtInfosSet = append(sqlsSrcExtInfosSet, stmtIte
              m.ExtInfos...)
18            }
19        }
20        for _, sql := range sqlsDst {
21            stmtItems, _, _ := business.ParseDDL(sql, request.Dbname)
22            for _, stmtItem := range stmtItems {
23                sqlsDstCoreInfosSet = append(sqlsDstCoreInfosSet, stmtI
              tem.CoreInfos...)
24                sqlsDstExtInfosSet = append(sqlsDstExtInfosSet, stmtIte
              m.ExtInfos...)
25            }
26        }
27        tableInfos := make([]*business.TableInfo, 0)
28        colInfoMap := make(map[string][]*dao.TableCoreInfoSet)
29        for _, colInfo := range sqlsSrcCoreInfosSet {
30            colInfoMap[*colInfo.TableName] = append(colInfoMap[*colInf
              o.TableName], colInfo)
```



```

31     }
32     optInfoMap := make(map[string][]*dao.TableExtInfoSet)
33     for _, optInfo := range sqlsSrcExtInfosSet {
34         optInfoMap[*optInfo.TableName] = append(optInfoMap[*optInfo
35             o.TableName], optInfo)
36     }
37     for tableName, coreInfos := range colInfoMap {
38         tableInfos = append(tableInfos, business.TableInfo{
39             TableName: tableName,
40             CoreInfo:   business.TableCoreInfoSetToTableCoreInfo(core
41                 Infos),
42             ExtInfo:   business.TableExtInfoSetToTableExtInfo(optIn
43                 foMap[tableName]),
44         })
45     }
46     sqlsDstCoreInfos := business.TableCoreInfoSetToTableCoreInfo(sq
47         lsDstCoreInfosSet)
48     sqlsDstExtInfos := business.TableExtInfoSetToTableExtInfo(sqlsD
49         stExtInfosSet)
50     anss, _ := business.GetUpgradeSqls(tableInfos, sqlsDstCoreInfo
51         s, sqlsDstExtInfos)
52     out := NewCompareTableStructResponse()
53     out.Data.Sql = anss
54     return &out, nil
55 }
56 }
57 }

```

compare_table_struct

POST http://{{test_addr}}/api/dbversion/v1/compareTables

Params Authorization Headers (9) **Body** Pre-request Script Tests

none form-data x-www-form-urlencoded **raw** binary JSON (application/json)

```
1 {
2   "dbname": "db_tool",
3   "sql_content_dst": "CREATE TABLE products (
4     product_id INT PRIMARY KEY AUTO_INCREMENT,
5     product_name VARCHAR(99) NOT NULL,
6     price DECIMAL(10, 2) NOT NULL,
7     description TEXT,
8     stock_quantity INT DEFAULT 3
9   );
10  ",
11   "sql_content_src": "CREATE TABLE products (
12     product_id INT PRIMARY KEY AUTO_INCREMENT,
13     product_name VARCHAR(100) NOT NULL,
14     price DECIMAL(10, 2) NOT NULL,
15     category VARCHAR(50),
16     stock_quantity INT DEFAULT 0
17   );"
18 }
```

Body Cookies Headers (3) Test Results

Pretty Raw Preview JSON

```
1 {
2   "code": 0,
3   "data": {
4     "sql": [
5       "ALTER TABLE products ADD COLUMN `description` TEXT;",
6       "ALTER TABLE products DROP COLUMN `category`;",
7       "ALTER TABLE products MODIFY COLUMN `product_name` VARCHAR(99) NOT NULL;",
8       "ALTER TABLE products MODIFY COLUMN `stock_quantity` INT DEFAULT '3';"
9     ],
10   },
11   "msg": ""
12 }
```

执行ddl语句

Bash

```
1 func NewExecDdlsService(moduleCtx *ctx.ModuleContext) ExecDdlsFunc {
2     return func(ctx context.Context, request *ExecDdlsRequest) (*ExecDd
3         lsResponse, error) {
4         err := validator.Validate(request)
5         if err != nil {
6             return nil, err
7         }
8         out := NewExecDdlsResponse()
9         str := "use "
10        database := fmt.Sprintf("ecis_%s", request.Dbname)
11        createSQL := fmt.Sprintf("CREATE DATABASE IF NOT EXISTS %s;", d
12            atabase)
13        _, err = moduleCtx.Db.Exec(ctx, createSQL)
14        if err != nil {
15            return nil, err
16        }
17        str = str + database + " ;"
18        _, err = moduleCtx.Db.Exec(ctx, str)
19        if err != nil {
20            return nil, err
21        }
22        total := 0
23        for _, sql := range request.Ddl {
24            fmt.Println(sql)
25            _, err := moduleCtx.Db.Exec(ctx, sql)
26            if err == nil {
27                total++
28            } else {
29                out.Msg = out.Msg + sql
30                fmt.Println()
31                fmt.Println(err)
32                fmt.Println()
33            }
34        }
35        out.Msg = strconv.Itoa(total)
36        return &out, nil
37    }
```

```
35     }  
36 }
```

POST dele POST g GET sh POST c [CONFL POST dele [CONFL POST crea [CONFL GET se POST crea POST dele POST gran POST c POST X +

execDdls

POST http://{{test_addr}}/api/dbversion/v1/execDdls

Params Authorization Headers (9) Body Pre-request Script Test Results

none form-data x-www-form-urlencoded raw binary JSON

```
1 {  
2   "dbname": "student",  
3   "ddl": [  
4     "CREATE TABLE IF NOT EXISTS student (id INT PRIMARY KEY AUTO_INCREMENT,  
5       VARCHAR(255), phone VARCHAR(20), created_at DATETIME DEFAULT CURRENT_TIMESTAMP);",  
6     "ALTER TABLE student ADD COLUMN date_of_birth DATE;",  
7     "ALTER TABLE student MODIFY COLUMN age TINYINT;",  
8     "ALTER TABLE student DROP COLUMN address;"  
9   ]  
10 }
```

Body Cookies Headers (3) Test Results

Pretty Raw Preview JSON

```
1 {  
2   "code": 0,  
3   "msg": "4"  
4 }
```

命令提示符 - mysql -uroot -p

```
mysql> show tables;  
Empty set (0.00 sec)  
  
mysql> show tables;  
+-----+  
Tables_in_ecis_student  
+-----+  
student  
+-----+  
1 row in set (0.00 sec)  
  
mysql> desc student;  
+-----+-----+-----+-----+-----+-----+  
Field      Type      Null    Key    Default      Extra  
+-----+-----+-----+-----+-----+-----+  
id          int       NO      PRI    NULL         aut  
first_name  varchar(255) NO      NO     NULL         aut  
last_name   varchar(255) NO      NO     NULL         aut  
age         tinyint   YES     NO     NULL         aut  
gender      varchar(10) YES     NO     NULL         aut  
email       varchar(255) YES     NO     NULL         aut  
phone       varchar(20) YES     NO     NULL         aut  
created_at  datetime  YES     NO     CURRENT_TIMESTAMP DEF  
updated_at  datetime  YES     NO     CURRENT_TIMESTAMP DEF  
date_of_birth date       YES     NO     NULL         aut  
+-----+-----+-----+-----+-----+-----+  
10 rows in set (0.01 sec)  
  
mysql>
```

升级sql（包含表cp信息、对比、执行ddl）

Bash

```
1 func NewUpdateSqlService(moduleCtx *ctx.ModuleContext) UpdateSqlFunc {
2     return func(ctx context.Context, request *UpdateSqlRequest) (*UpdateSqlResponse, error) {
3         err := validator.Validate(request)
4         if err != nil {
5             return nil, err
6         }
7         sqlsSrcCoreInfosSet []*dao.TableCoreInfoSet
8         sqlsSrcExtInfosSet []*dao.TableExtInfoSet
9         sqlsDstCoreInfosSet []*dao.TableCoreInfoSet
10        sqlsDstExtInfosSet []*dao.TableExtInfoSet
11        sqlsSrc := business.ExtractDDLStatements(request.SqlContentSrc)
12        sqlsDst := business.ExtractDDLStatements(request.SqlContentDst)
13        for _, sql := range sqlsSrc {
14            stmtItems, _, _ := business.ParseDDL(sql, request.Dbname)
15            for _, stmtItem := range stmtItems {
16                sqlsSrcCoreInfosSet = append(sqlsSrcCoreInfosSet, stmtItem.CoreInfos...)
17                sqlsSrcExtInfosSet = append(sqlsSrcExtInfosSet, stmtItem.ExtInfos...)
18            }
19        }
20        for _, sql := range sqlsDst {
21            stmtItems, _, _ := business.ParseDDL(sql, request.Dbname)
22            for _, stmtItem := range stmtItems {
23                sqlsDstCoreInfosSet = append(sqlsDstCoreInfosSet, stmtItem.CoreInfos...)
24                sqlsDstExtInfosSet = append(sqlsDstExtInfosSet, stmtItem.ExtInfos...)
25            }
26        }
27        tableInfos := make([]business.TableInfo, 0)
28        colInfoMap := make(map[string][]*dao.TableCoreInfoSet)
29        for _, colInfo := range sqlsSrcCoreInfosSet {
30            colInfoMap[*colInfo.TableName] = append(colInfoMap[*colInfo.TableName], colInfo)
31        }
```

```

32     optInfoMap := make(map[string][]*dao.TableExtInfoSet)
33     for _, optInfo := range sqlsSrcExtInfosSet {
34         optInfoMap[*optInfo.TableName] = append(optInfoMap[*optInfo
35             o.TableName], optInfo)
36     }
37     for tableName, coreInfos := range colInfoMap {
38         tableInfos = append(tableInfos, business.TableInfo{
39             TableName: tableName,
40             CoreInfo:   business.TableCoreInfoSetToTableCoreInfo(core
41                 eInfos),
42             ExtInfo:   business.TableExtInfoSetToTableExtInfo(optIn
43                 foMap[tableName]),
44         })
45     }
46     sqlsDstCoreInfos := business.TableCoreInfoSetToTableCoreInfo(sq
47         lsDstCoreInfosSet)
48     sqlsDstExtInfos := business.TableExtInfoSetToTableExtInfo(sqlsD
49         stExtInfosSet)
50     anss, _ := business.GetUpgradeSqls(tableInfos, sqlsDstCoreInfo
51         s, sqlsDstExtInfos)
52
53     str := "use "
54     database := request.Dbname
55     createSQL := fmt.Sprintf("CREATE DATABASE IF NOT EXISTS %s;", d
56         atabase)
57
58     str = str + database + " ;"
59
60     if request.Dns != "" {
61         // database := fmt.Sprintf("ecis_%s", request.Dbname)
62         db, err := sql.Open("mysql", request.Dns)
63         _, err = db.Exec(createSQL)
64         if err != nil {
65             return nil, err
66         }
67         _, err = db.Exec(str)
68         if err != nil {
69             return nil, err
70         }
71         total := 0
72         for _, sql := range anss {

```

```

65         fmt.Println(sql)
66         _, err := db.Exec(sql)
67         if err == nil {
68             total++
69         } else {
70             fmt.Println()
71             fmt.Println(err)
72             fmt.Println()
73         }
74     }
75 } else {
76     _, err = moduleCtx.Db.Exec(ctx, createSQL)
77     if err != nil {
78         return nil, err
79     }
80     _, err = moduleCtx.Db.Exec(ctx, str)
81     if err != nil {
82         return nil, err
83     }
84     total := 0
85     for _, sql := range anss {
86         fmt.Println(sql)
87         _, err := moduleCtx.Db.Exec(ctx, sql)
88         if err == nil {
89             total++
90         } else {
91             fmt.Println()
92             fmt.Println(err)
93             fmt.Println()
94         }
95     }
96 }
97
98 out := NewUpdateSqlResponse()
99 out.Data.Sql = anss
100
101 return &out, nil
102 }
103 }
104

```

这里测试其他账户登录进行操作的结果

update

POST http://{{test_addr}}/api/dbversion/v1/updateSql

Params

Authorization

Headers (9)

Body

Pre-request

none

form-data

x-www-form-urlencoded

raw

body

1 {

2 "dbname": "update_test",

3 "sql_content_src": "CREATE TABLE users (

4 id INT PRIMARY KEY AUTO_INCREMENT,

5 username VARCHAR(100) NOT NULL,

6 email VARCHAR(255) NOT NULL,

7 age INT,

8 created_at DATETIME DEFAULT CURRENT_TIMESTAMP

9);

10 ",

11 "sql_content_dst": "CREATE TABLE users (

12 id INT PRIMARY KEY AUTO_INCREMENT,

13 username VARCHAR(100) NOT NULL,

14 email VARCHAR(255) NOT NULL,

15 age TINYINT,

16 gender VARCHAR(10),

17 created_at DATETIME DEFAULT CURRENT_TIMESTAMP

18);

19 ",

20 "dns": "update:dbversion@tcp(127.0.0.1:3306)/update_test"

21 }

Body

Cookies

Headers (3)

Test Results

Pretty

Raw

Preview

JSON

body

1 {

2 "code": 0,

3 "data": {

4 "sql": [

5 "ALTER TABLE users ADD COLUMN `gender` VARCHAR(10);",

6 "ALTER TABLE users MODIFY COLUMN `age` TINYINT;"

7]

8 },

9 "msg": ""

10 }

命令提示符 - mysql -uupdate -p

created_at datetime YES CURRENT_TIMESTAMP

5 rows in set (0.00 sec)

mysql> desc users;

Field	Type	Null	Key	Default
id	int	NO	PRI	NULL
username	varchar(100)	NO		NULL
email	varchar(255)	NO		NULL
age	int	YES		NULL
created_at	datetime	YES		CURRENT_TIMESTAMP

5 rows in set (0.00 sec)

mysql> desc users;

Field	Type	Null	Key	Default
id	int	NO	PRI	NULL
username	varchar(100)	NO		NULL
email	varchar(255)	NO		NULL
age	tinyint	YES		NULL
created_at	datetime	YES		CURRENT_TIMESTAMP
gender	varchar(10)	YES		NULL

6 rows in set (0.00 sec)

mysql>