**迁移工具接口说明书**

**V1.0**

天津神州通用数据技术有限公司

目录

[一、 概述 4](#_Toc16934)

[1. 接口规约 4](#_Toc27724)

[2. 返回说明(统一的接口返回方式) 4](#_Toc32094)

[二、 迁移工具提供的接口 5](#_Toc22538)

[1． 登录模块 5](#_Toc10510)

[1. 获取公钥 5](#_Toc13020)

[2. 获取验证码 5](#_Toc32530)

[3. 登录验证 5](#_Toc31997)

[4. 登出验证 5](#_Toc15687)

[2． 用户管理 5](#_Toc26939)

[1. 新增用户 5](#_Toc3609)

[2. 删除用户 5](#_Toc6080)

[3. 修改用户 5](#_Toc31412)

[4. 修改用户密码 5](#_Toc22543)

[5. 分页查询 5](#_Toc23436)

[6. 查询用户权限 5](#_Toc13272)

[3． 角色管理 5](#_Toc8794)

[1. 新增角色 5](#_Toc20889)

[2. 删除角色 5](#_Toc28858)

[3. 修改角色 5](#_Toc29994)

[4. 分页查询 5](#_Toc18708)

[5. 查询所有角色 5](#_Toc21912)

[4． 菜单管理 5](#_Toc21367)

[1. 查询菜单树 5](#_Toc31493)

[2. 查询菜单列表 5](#_Toc16007)

[5． 数据迁移 5](#_Toc11370)

[1. 测试连接 5](#_Toc17591)

[2. 迁移参数保存接口 7](#_Toc29795)

[3. 获取字段类型映射 10](#_Toc24665)

[4. 保存字段类型映射 12](#_Toc15733)

[5. 获取所有模式信息 14](#_Toc19803)

[6. 获取表空间信息 16](#_Toc18796)

[7. 获取模式下的所有信息 18](#_Toc11678)

[8. 获取表下的所有信息 20](#_Toc9413)

[9. 批量获取表下的所有信息 22](#_Toc3402)

[10. 保存迁移配置信息 23](#_Toc32316)

[11. 保存模式信息 24](#_Toc11923)

[12. 执行迁移 25](#_Toc19446)

[6． 迁移历史 26](#_Toc196)

# 概述

1. 接口规约
   1. http:[ip]:[port]：通过平台的域名+端口来请求,token统一放在header中。
   2. 编码方式：UTF-8
   3. 请求协议：HTTP
2. 返回说明(统一的接口返回方式)

|  |  |  |
| --- | --- | --- |
| 参数名称 | 类型 | 描述 |
| code | int | 状态码：  0： 成功  -7：参数错误  -9：登录超时  999：未知异常，服务端错误  ... |
| msg | String | 信息描述 |
| data | Object | 返回的数据 |

1. 迁移工具提供的接口

## 登录模块

1. 获取公钥
   1. 请求路径：getPublicKey
   2. 请求方式：get
   3. 请求类型：application/json;charset=UTF-8
   4. 请求参数

无

* 1. 响应信息

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 描述 |
| code | int | 接口状态 0：成功 |
| msg | String | 信息描述 |
| data | Object | 返回的数据 |

* 1. 请求示例  
     <http://localhost:8081/getPublicKey?t=1611742344084>  
     请求参数  
      无

返回值：  
 {  
 "code": 0,  
 "msg": null,  
 "data":“xxx”//加密后的公钥

}

1. 获取验证码
   1. 请求路径：captcha.jpg
   2. 请求方式：get
   3. 请求类型：image/jpeg
   4. 请求参数

无

* 1. 响应信息  
     图片流
  2. 请求示例  
     <http://localhost:8081/captcha.jpg?t=1611742343536>  
     请求参数  
      无

返回值：  
 无

1. 登录验证
2. 登出验证

## 用户管理

1. 新增用户
2. 删除用户
3. 修改用户
4. 修改用户密码
5. 分页查询
6. 查询用户权限

## 角色管理

1. 新增角色
2. 删除角色
3. 修改角色
4. 分页查询
5. 查询所有角色

## 菜单管理

1. 查询菜单树
2. 查询菜单列表

## 数据迁移

**注：此模块接口需按照文档编号顺序依次执行**

1. 测试连接
   1. 请求路径：dataMigration/testConnection
   2. 请求方式：post
   3. 请求类型：application/json;charset=UTF-8
   4. 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 数据类型 | 必填 | 描述 |
| connectionType | int | Y | 连接类型 0：源端 1：目标端 |
| dataSource | String | Y | 数据库类型 |
| driver | String | Y | 驱动类型 |
| dataBase | String | Y | 数据库 |
| server | String | Y | 数据库ip |
| port | String | Y | 数据库端口 |
| userName | String | Y | 用户名 |
| Password | String | Y | 密码 |

* 1. 响应信息

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 描述 |
| code | int | 接口状态 0：成功 |
| msg | String | 信息描述 |
| data | Object | 返回的数据 |

* 1. 请求示例  
     http://localhost:8081/dataMigration/testConnection  
     请求参数  
     {

"connectionType":0,

"dataBase":"OSRDB",

"dataSource":"ShenTong",

"driver":"ShenTong JDBC Driver",

"password":"xxx",

"port":"2003",

"server":"localhost",

"userName":"xxx"

}

返回值：  
 {  
 "code": 0,  
 "msg": null,  
 "data": null

}

1. 迁移参数保存接口
   1. 请求路径：dataMigration/saveConfig
   2. 请求方式：post
   3. 请求类型：application/json;charset=UTF-8
   4. 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 数据类型 | 必填 | 描述 |
| isPartition | boolean | Y | 是否按照水平分区迁移 |
| isCheck | boolean | Y | 是否检查check约束 |
| migComment | boolean | Y | 是否迁移注释 |
| isSequence | boolean | Y | 是否迁移序列 |
| isAutoAssignPartitionName | boolean | Y | 后台自动分配水平分区分区名 |
| fromDB | boolean | Y | 是否从数据库读取序列 |
| isSortView | boolean | Y | 是否对视图排序 |
| isMigStorage | boolean | Y | 是否迁移存储选项 |
| isReloadMetadata | boolean | Y | 强制重新获取元数据对象信息 |
| isUpper | boolean | Y | 迁移时强制使用大写标识 |
| savePWD | boolean | Y | 保存数据库登录密码 |
| analyze | boolean | Y | 迁移完成后执行Analyze |
| isBreakpoint | boolean | Y | 断点续传 |
| isMigrateOraclePl | boolean | Y | 是否迁移Oracle pl对象 |
| isTranscoding | boolean | Y | Mysql数据编码转换(可用Navicat查看数据显示是否正常) |
| migErrorHandleStrategy | String | Y | 迁移遇错策略：遇错停止：stop，遇错忽略：neglect |
| colMultFactor | double | Y | 列长度倍数 |
| fetchSize | int | Y | 源数据库一次迁移的fetchSize |
| charSet | String | Y | 字符转换编码 |
| logMode | int | Y | 创建日志 0：是 1：否 2：与源端一致 |
| transaction | int | Y | 事务设置 0：自动提交 1：批提交 |
| threadCount | int | Y | 迁移线程数 |
| useCommon | boolean | Y | 使用普通方式 |
| isSerializable | boolean | Y | 强制串行化迁移索引 |
| isQuickMig | boolean | Y | 快速迁移 |
| isSentByRow | boolean | Y | 是否设置的是发送行数  false:设置缓冲区大小;true:发送行数 |
| bufferSize | int | Y | 大批量数据导入缓冲区(单位M) |
| maxRowLimits | int | Y | 大批量数据导入发送行数 |
| monitorTaskProgress | boolean | Y | 开启迁移线程监控 |
| monitorStrategy | int | Y | 监控类型：0：按数据大小 1：按行数 |

* 1. 响应信息

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 描述 |
| code | int | 接口状态 0：成功 |
| msg | String | 信息描述 |
| data | Object | 返回的数据 |

* 1. 请求示例  
     http://localhost:8081/dataMigration/saveConfig  
     请求参数：  
     {

"analyze":false,

"autoAssignPartitionName":false,

"bufferSize":10,

"charSet":"UTF8",

"check":true,

"checkSourceObject":false,

"colMultFactor":1,

"decode":false,

"encode":false,

"fetchCount":"50",

"fetchSize":1000,

"fromDB":false,

"log":1,

"max":512,

"maxRowLimits":1000,

"migComment":false,

"migErrorTactics":"neglect",

"migStorage":true,

"migrateOraclePl":false,

"min":64,

"monitorStrategy":1,

"monitorTaskProgress":false,

"onlyDefinition":false,

"onlySt2St":false,

"partition":true,

"preferencesChanged":false,

"quickMig":false,

"savePWD":true,

"sentByRow":true,

"sequence":true,

"serializable":false,

"sortView":false,

"threadCount":5,

"transaction":0,

"transcoding":false,

"transferType":0,

"upper":true,

"useCommonProtocol":false,

"useRowCache":true

}  
返回值：  
 {  
 "code": 0,  
 "msg": null,  
 "data": null

}

1. 获取字段类型映射
   1. 请求路径：dataMigration/getTypeMapping
   2. 请求方式：post
   3. 请求类型：application/json;charset=UTF-8
   4. 请求参数  
      无
   5. 响应信息

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 描述 |
| code | int | 接口状态 0：成功 |
| msg | String | 信息描述 |
| data | Object | 返回的数据 |

* 1. 请求示例  
     <http://localhost:8081/dataMigration/getTypeMapping?t=1611731359285>返回值  
     [

{

"accTypes":[

{

"id":"TCHAR2CHARPS",

"mValue":"",

"pLmax":"",

"pLmin":"",

"pMax":"",

"pMin":"",

"pValue":"",

"sLmax":"",

"sLmin":"",

"sMax":"",

"sMin":"",

"sValue":"",

"selected":true,

"ttypeName":"CHAR"

},

...

],

"hasL":false,

"hasP":false,

"hasS":false,

"id":"SCHARPSL",

"lExpression":"",

"lLmax":"",

"lLmin":"",

"lMax":"",

"lMin":"",

"lValue":"",

"pExpression":"",

"pLmax":"",

"pLmin":"",

"pMax":"",

"pMin":"",

"pValue":"",

"sExpression":"",

"sLmax":"",

"sLmin":"",

"sMax":"",

"sMin":"",

"sValue":"",

"sname":"CHAR",

"targetFieldType":"CHAR",

"targetLExpression":"",

"targetPExpression":"",

"targetSExpression":""

},

...

]

1. 保存字段类型映射
   1. 请求路径：dataMigration/saveTypeMapping
   2. 请求方式：post
   3. 请求类型：application/json;charset=UTF-8
   4. 请求参数  
      详见示例
   5. 响应信息

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 描述 |
| code | int | 接口状态 0：成功 |
| msg | String | 信息描述 |
| data | Object | 返回的数据 |

* 1. 请求示例  
     <http://localhost:8081/dataMigration/saveTypeMapping>[

{

"accTypes":[

{

"id":"TCHAR2CHARPS",

"mValue":"",

"pLmax":"",

"pLmin":"",

"pMax":"",

"pMin":"",

"pValue":"",

"sLmax":"",

"sLmin":"",

"sMax":"",

"sMin":"",

"sValue":"",

"selected":true,

"ttypeName":"CHAR"

},

...

],

"hasL":false,

"hasP":false,

"hasS":false,

"id":"SCHARPSL",

"lExpression":"",

"lLmax":"",

"lLmin":"",

"lMax":"",

"lMin":"",

"lValue":"",

"pExpression":"",

"pLmax":"",

"pLmin":"",

"pMax":"",

"pMin":"",

"pValue":"",

"sExpression":"",

"sLmax":"",

"sLmin":"",

"sMax":"",

"sMin":"",

"sValue":"",

"sname":"CHAR",

"targetFieldType":"CHAR",

"targetLExpression":"",

"targetPExpression":"",

"targetSExpression":""

},

...

]

1. 获取所有模式信息
   1. 请求路径：dataMigration/getAllPattern
   2. 请求方式：post
   3. 请求类型：application/json;charset=UTF-8
   4. 请求参数  
      无
   5. 返回数据

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 描述 |
| code | int | 接口状态 0：成功 |
| msg | String | 信息描述 |
| data | Object | 返回的数据 |

* 1. 请求示例  
     <http://localhost:8081/dataMigration/getAllPattern>  
     返回值  
     {

"sourcePatternName":[

"PUBLIC",

"STAGENT",

"SYSAUDIT",

...

],

"targetPattern":[

{

"create":false,

"hasLoadData":false,

"name":"DIRECTORIES",

"name4Mig":"DIRECTORIES",

"oid":"8"

},

{

"create":false,

"hasLoadData":false,

"name":"HR",

"name4Mig":"HR",

"oid":"35329"

},

...

],

"sourceSchemaIdNameMap":{

"1":"SYSDBA",

"12":"WMSYS",

"2":"SYSSECURE",

...

},

"targetSchemaIdNameMap":{

"1":"SYSDBA",

"12":"WMSYS",

"2":"SYSSECURE",

...

},

"sourcePattern":[

{

"create":false,

"hasLoadData":false,

"name":"DIRECTORIES",

"name4Mig":"DIRECTORIES",

"oid":"8"

},

{

"create":false,

"hasLoadData":false,

"name":"HR",

"name4Mig":"HR",

"oid":"35329"

},

{

"create":false,

"hasLoadData":false,

"name":"PUBLIC",

"name4Mig":"PUBLIC",

"oid":"2200"

}

...

]

}

1. 获取表空间信息
   1. 请求路径：dataMigration/getTableSpace
   2. 请求方式：post
   3. 请求类型：application/json;charset=UTF-8
   4. 请求参数  
      无
   5. 响应信息

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 描述 |
| sourceTablespace | List | 源端表空间集合 |
| targetTablespace | List | 目标端表空间集合 |

* 1. 请求示例  
     <http://localhost:8081/dataMigration/getTableSpace>

{

"code":0,

"data":{

"sourceTablespace":[

{

"commentDefinition":null,

"definition":null,

"enabled":"READ WRITE",

"optionDefinition":null,

"sourceName":"SYSTEM",

"targetName":"SYSTEM",

"targetNameByCompare":"SYSTEM",

"tsDataFiles":Array[1],

"tsFill":"70",

"tsInitSize":"64",

"tsLogging":"true",

"tsMaxSize":"0",

"tsNextSize":"64",

"tsPctFree":"10",

"tsPctUsed":"40",

"tsSplit":"50"

},

Object{...},

Object{...},

Object{...}

],

"targetTablespace":[

{

"commentDefinition":null,

"definition":null,

"enabled":"READ WRITE",

"optionDefinition":null,

"sourceName":"SYSTEM",

"targetName":"SYSTEM",

"targetNameByCompare":"SYSTEM",

"tsDataFiles":[

{

"autoExtensible":true,

"dataFile":"OSRDB01.dbf",

"initSize":"204800",

"maxSize":"0",

"nextSize":"20480"

}

],

"tsFill":"70",

"tsInitSize":"64",

"tsLogging":"true",

"tsMaxSize":"0",

"tsNextSize":"64",

"tsPctFree":"10",

"tsPctUsed":"40",

"tsSplit":"50"

},

Object{...},

Object{...},

Object{...}

]

},

"msg":null

}

1. 获取模式下的所有信息
   1. 请求路径：dataMigration/getPatternDataById
   2. 请求方式：post
   3. 请求类型：application/x-www-form-urlencoded;charset=UTF-8
   4. 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 数据类型 | 必填 | 描述 |
| id | String | Y | 模式ID |

* 1. 响应信息

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 描述 |
| tableNameList | List | 此模式下表名集合 |
| schemaInfo | Object | 模式信息 |

* 1. 请求示例  
     <http://localhost:8081/dataMigration/getPatternDataById>

请求参数

{ id: “xxx”}

返回值  
 {

"code":0,

"data":{

"tableNameList":Array[8],

"schemaInfo":{

"checks":Array[0],

"create":false,

"fullIndexInfos":Array[0],

"functionInfos":Array[0],

"hasLoadData":false,

"indexInfos":Array[0],

"mViewLogs":Array[0],

"materViewsInfo":Array[0],

"name":"SYSDBA",

"name4Mig":"SYSDBA",

"oid":"1",

"packages":Array[0],

"procedures":Array[1],

"sequences":Array[1],

"sqlTableInfos":Array[0],

"synonyms":Array[0],

"tableInfos":Array[8],

"triggers":Array[0],

"typeInfos":Array[0],

"viewDepend":null,

"viewInfos":Array[1]

}

},

"msg":null

}

1. 获取表下的所有信息
   1. 请求路径：dataMigration/getAllTableData
   2. 请求方式：post
   3. 请求类型：application/x-www-form-urlencoded;charset=UTF-8
   4. 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 数据类型 | 必填 | 描述 |
| schemaName | String | Y | 模式名称 |
| tableName | String | Y | 表名 |

* 1. 返回数据

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 描述 |
| targetSchema | String | 目的模式名 |
| mode | Int | 表的迁移模式 |
| primarykey | Object | 主键 |
| indexInfos | List | 索引信息列表 |
| checkInfos | List | 约束信息列表 |
| foreignKeys | List | 外键信息列表 |
| uniqueInfos | List | 唯一约束信息列表 |
| triggers | List | 触发器信息列表 |
| fullIndexInfos | List | 全文索引信息列表 |
| columns | List | 表列信息 |
| tableSpace | String | 表空间 |
| tableSegmentInit | String | 段初始大小 |
| tableSegmentNext | String | 段自增大小 |
| tableSegmentMax | String | 段最大大小 |
| tablePctFree | String | 表空间空闲百分比 |
| tablePctUsed | String | 表空间最小使用百分比 |
| tableLoging | String | 是否记录日志 |
| partitionInfo | String | 表分区信息 |

* 1. 请求示例  
     <http://localhost:8081/dataMigration/getAllTableData>

请求参数  
 {“schemaName”:“xxx”, “tableName”: “xxx”}  
 返回值  
 {

"code":0,

"data":{

"checkInfos":Array[0],

"columns":Array[2],

"commentDefinition":null,

"common":true,

"definition":null,

"edit":false,

"foreignKeys":Array[0],

"fullIndexInfos":Array[0],

"hasLoadCheck":true,

"hasLoadData":false,

"hasLoadForeignKey":true,

"hasLoadFullIndex":true,

"hasLoadIndex":true,

"hasLoadPartitionIndexInfo":false,

"hasLoadPartitionInfo":false,

"hasLoadTrigger":true,

"hasLoadUnique":true,

"hasLob":false,

"indexInfos":Array[0],

"isCommon":true,

"isMappingError":false,

"mappingError":false,

"mode":1,

"noNullCol":false,

"normal":false,

"optionDefinition":null,

"overFlowNumber":null,

"partitionInfo":null,

"primarykey":Object{...},

"recordStates":Array[0],

"sourceName":"DEPT",

"tHasLoadData":false,

"tableLoging":"YESLOG",

"tablePctFree":"10",

"tablePctUsed":"40",

"tableSegmentInit":"64",

"tableSegmentMax":"0",

"tableSegmentNext":"64",

"tableSpace":"SYSTEM",

"targetName":"DEPT",

"targetNameByCompare":"DEPT",

"targetSchema":"SYSDBA",

"targetSchemaForOriginal":"SYSDBA",

"triggers":Array[0],

"type":"r",

"uniqueInfos":Array[0],

"whereStr":null

},

"msg":null

}

1. 批量获取表下的所有信息
   1. 请求路径：dataMigration/batchGetAllTableData
   2. 请求方式：post
   3. 请求类型：application/json;charset=UTF-8
   4. 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 数据类型 | 必填 | 描述 |
| tables | List | Y | 表名 模式名集合 |

* 1. 返回数据  
     同dataMigration/getAllTableData接口
  2. 请求示例  
     [http://localhost:8081/dataMigration/batchGetAllTableData](http://localhost:8081/dataMigration/getAllTableData)

请求参数  
 [

{

"schemaName":"xxx",

"tableName":"xxx"

},

{

"schemaName":"xxx",

"tableName":"xxx"

}

]

返回值  
 同dataMigration/getAllTableData接口返回值

1. 保存迁移配置信息
   1. 请求路径：dataMigration/saveMigrationInfo
   2. 请求方式：post
   3. 请求类型：application/json;charset=UTF-8
   4. 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 数据类型 | 必填 | 描述 |
| isOpenConstraintMigrate | Boolean | Y | 是否开启了强制迁移 |
| checkTargetSchemaName | String | Y | 选中的强制迁移模式名 |
| checkMigrationCircle | int | Y | 选择的迁移范围 |

* 1. 响应信息

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 描述 |
| code | int | 接口状态 0：成功 |
| msg | String | 信息描述 |
| data | Object | 返回的数据 |

* 1. 请求示例  
     <http://localhost:8081/dataMigration/saveMigrationInfo>

请求参数  
 {

"checkMigrationCircle":0,

"checkTargetSchemaName":"DIRECTORIES",

"isOpenConstraintMigrate":false

}  
 返回值  
 {

"code":0,

"data":null,

"msg":null

}

1. 保存模式信息
   1. 请求路径：dataMigration/saveSchemaInfo
   2. 请求方式：post
   3. 请求类型：application/json;charset=UTF-8
   4. 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 数据类型 | 必填 | 描述 |
| schemaInfo | List | Y | 模式对象集合 |

* 1. 响应信息

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 描述 |
| code | int | 接口状态 0：成功 |
| msg | String | 信息描述 |
| data | Object | 返回的数据 |

* 1. 请求示例  
     <http://localhost:8081/dataMigration/saveSchemaInfo>

请求参数  
 [

{

"checks":Array[0],

"create":false,

"fullIndexInfos":Array[0],

"functionInfos":Array[0],

"hasLoadData":false,

"indexInfos":Array[0],

"mViewLogs":Array[0],

"materViewsInfo":Array[0],

"name":"SYSDBA",

"name4Mig":"SYSDBA",

"oid":"1",

"packages":Array[0],

"procedures":Array[1],

"sequences":Array[1],

"sqlTableInfos":Array[0],

"synonyms":Array[0],

"tableInfos":Array[8],

"triggers":Array[0],

"typeInfos":Array[0],

"viewDepend":null,

"viewInfos":Array[1]

}

...

]  
 返回值  
 {

"code":0,

"data":null,

"msg":null

}

1. 执行迁移
   1. 请求路径：dataMigration/startMigration
   2. 请求方式：post
   3. 请求类型：application/json;charset=UTF-8
   4. 请求参数  
      无
   5. 响应信息

|  |  |  |
| --- | --- | --- |
| 参数名 | 数据类型 | 描述 |
| code | int | 接口状态 0：成功 |
| msg | String | 信息描述 |
| data | Object | 返回的数据 |

* 1. 请求示例  
     <http://localhost:8081/dataMigration/startMigration>

## 迁移历史