

## Notes :

- 1.(hostname) in the "Rest Path" column is the path of Restful, the default value is http://ip:8080/HareRestful/webapi.
- 2.When you try to call any api, you have to call the connection method.
- 3.The date format of input and output parameters is JSON , please follow use the format to call restful services.
- 4.Avoid using the same name between Meta tables and HBase tables.
- 5.Add the jar "HareDB\_HBase\_Coprocessor-1.98.06.01.jar" in HareDBClient-1.98.06.01.jar/WEB-INF/lib to hdfs://{host}:8020/haredb/jar/haredbCP.jar

Functions	Http Method	Rest Path	Input parameters	Descriptions of input parameters	Output parameters	The descriptions of output parameters
Create a connection	POST	(hostname)/connect	{           "zookeeperHost":"host1",           "zookeeperPort":"2181",           "nameNodeHostPort":"hdfs://host1:8020",           "rmAddressHostPort":"host1:8032",           "rmSchedulerAddressHostPort":"host1:8030",           "rmResourceTrackerAddressHostPort":"host1:8031",           "rmAdminAddressHostPort":"host1:8033",           "mrJobHistoryAddress":"host1:10020",           "hiveConnType":"LOCAL",           "dbName":"default",           "dbBrand":"MYSQL",           "hiveServer2Url":"jdbc:hive2://host1:10000",           "metaStoreConnectURL":"jdbc:mysql://10.15.1.32:3306/hare",           "metaStoreConnectDriver":"com.mysql.jdbc.Driver",           "metaStoreConnectUserName":"root",           "metaStoreConnectPassword":"123456",           "enableKerberos":"true",           "hbaseMasterPrincipal":"hbase/_HOST@ISLAND.COM",           "hbaseRegionServerPrincipal":"hbase/_HOST@ISLAND.COM",           "dfsNameNodePrincipal":"hdfs/_HOST@ISLAND.COM",           "dfsDataNodePrincipal":"hdfs/_HOST@ISLAND.COM",           "hiveMetaStorePrincipal":"hive/_HOST@ISLAND.COM",           "yarnResourceMgrPrincipal":"yarn/_HOST@ISLAND.COM",           "yarnNodeMgrPrincipal":"yarn/_HOST@ISLAND.COM",           "account":"#@#%\$%",           "pwd":"\$%&%("         }	host1 is the host name of the cluster's Master Please edit host1 to your host name of Master metaStoreConnectURL: the connection string of mysql. metaStoreConnectUserName: the account of mysql. metaStoreConnectPassword: the password of mysql. enableKerberos: If set the parameter to true, the principal of starting Kerberos has to be set. Note: If hiveConnType is SERVER2 , hiveServer2Url has to be set.	{           "auth": "true",           "responseTime": "0",           "status": "success",           "connectionKey": "24d56e"         }	responseTime: the response time(mini second). status: the result(success/error). exception: error messages(when error occurs). auth:verified status. connectionKey: If success, the system will return a Token.
Create a table	POST	(hostname)/baseadmin/create	{           "tableName":"table1",           "columnFamilies":["cf","cf1"],           "account":"#@#%\$%",           "pwd":"\$%&%("         }	tableName: the name of data tables, and it could not be set in Chinese. columnFamilies: make sure the count of column family has no less than one, and it could not be set in Chinese. connectionKey: the Token of connections.	{           "responseTime": "391",           "status": "success",           "auth": "true",           "exception": "exception"         }	responseTime: the response time(mini second). status: the result(success/error). exception: error messages(when error occurs). auth: verified status.
Upload data by bulkload	POST	(hostname)/bulkload/schema/upload	{           "schemaFilePath":"hdfs://host1:8020/schema/schema.txt",           "dataPath":"hdfs://host1:8020/tmp/data.txt",           "resultPath":"hdfs://host1:8020/tmp/exception.txt",           "account":"#@#%\$%",           "pwd":"\$%&%("         }	schemaFilePath: the file path of Schema dataPath: the file path of data resultPath: the directory path of result(the directory can't be created before.) All of the path are stored in HDFS connectionKey: the Token of connections.	{           "jobName":"applicatid000001",           "responseTime":"22334",           "status": "success",           "auth": "true"         }	jobName: the job name of upload. responseTime: the response time(mini second), not the approach time. status: the result(success/error), but the data is not loaded into HBase. auth: verified status.
Query the status of bulkload	POST	(hostname)/bulkload/status	{           "jobName":"applicatid000001",           "account":"#@#%\$%",           "pwd":"\$%&%("         }	jobName: the job name of upload. connectionKey: the Token of connections.	{           "auth": "true",           "responseTime": "28",           "status": "success",           "bulkloadFinishTime": "30-01-2015 16:13:00",           "bulkloadStartTime": "30-01-2015 16:12:29",           "jobId": "job_1422588512235_0010",           "jobName": "Hare_Application_1422605545249",           "jobStatus": "SUCCEEDED",           "mapProgress": "0.0%",           "reduceProgress": "0.0%",           "jobErrorMessage": "error message",           "auth": "true",           "exception": "exception"         }	bulkloadStartTime:the start time of bulkload. bulkloadFinishTime:the end time of bulkload. jobName: the job name of upload. jobId: the job ID of upload. jobStatus: the upload status(RUNNING,PREP,SUCCEEDED, FAILED,KILLED). jobErrorMessage: the error messages of upload(when jobStatus is FAILED). mapProgress: the percentage of map. reduceProgress: the percentage of reduce. responseTime: the response time(mini second). status: the result(success/error). auth: verified status. exception: error messages(when error occurs).
Search data by HareQL	POST	(hostname)/hareql/query	{           "tempFilePath":"/temp/ddd",           "sql":"select * from table81",           "page":1,           "limit":10,           "account":"#@#%\$%",           "pwd":"\$%&%("         }	tempFilePath: the path of result.txt the directory must start with /temp sql: HareQL page: the numbers of pages limit: the record number per page connectionKey: the Token of connections.	{           "fileSize": "136485",           "heads":["key","cf,column1"],           "responseTime":"123456",           "results":["rk1tvalue1",           "rk1tvalue2",           "rk3tvalue4",           "rkBtnull"],           "rowSize": "10000",           "status": "success",           "auth": "true",           "exception": "exception"         }	fileSize: the file size of the result(bytes). heads: the columns of the result. responseTime: the response time(mini second). results: results.it is the separator of column. rowSize: the numbers of the result. status: the result(success/error). auth: verified status. exception: error messages(when error occurs).
Submit HareQL	POST	(hostname)/hareql/submit	{           "tempFilePath":"/temp/ddd",           "sql":"select * from table81",           "page":1,           "limit":10,           "account":"#@#%\$%",           "pwd":"\$%&%("         }	tempFilePath: the path of result.txt the directory must start with /temp sql: HareQL page: the numbers of pages limit: the record numbers per page connectionKey: the Token of connections.	{           "responseTime":"123456",           "status": "success",           "auth": "true",           "exception": "exception"         }	responseTime: the response time(mini second). status: the result(success/error). auth: verified status. exception: error messages(when error occurs). the API is called after using the API "Search data by HareQL".
Query the status of searching HareQL	POST	(hostname)/hareql/status	{           "tempFilePath":"/temp/ddd",           "account":"#@#%\$%",           "pwd":"\$%&%("         }	tempFilePath: the path of result.txt. the directory must start with /temp. connectionKey: the Token of connections.	{           "auth": "true",           "responseTime": "0",           "status": "success",           "exception": "exception"         }	responseTime: the response time(mini second). status: the status of Query(success,running,error). auth: verified status. exception: error messages(when error occurs).

Scan a table	POST	{hostname}/htable/scan	{       "tableName": "table1",       "pageSize": "10",       "limit": "10",       "account": "@#\$%\$%",       "pwd": "%\$%&%(**",       "connectionKey": "24d56e"     }	tableName: the table name of HBase. pageSize: the number of pages limit: the record number per page connectionKey: the Token of connections.	{       "auth": "true",       "responseTime": "17",       "status": "success",       "heads": {         "columnFamily": "cf",         "qualifier": "column1"},         {         "columnFamily": "cf",         "qualifier": "column2"       }     },     "results": [       {         "item": ["column1 value", "column2 value"]       },       {         "item": ["value9", "value10"]       }     ],     "exception": "exception"   }	responseTime: the response time(mini second) status: the result(success/error) heads: the column of the result. columnFamily: column family. qualifier: column name. results: the query result. item: records. auth: verified status. exception: error messages(when error occurs).
Delete data	POST	{hostname}/hareql/query	{       "tempFilePath": "/temp/ddd",       "sql": "delete from table1 where `key`='rowkey1'",       "account": "@#\$%\$%",       "pwd": "%\$%&%(**",       "connectionKey": "24d56e"     }	tempFilePath: the path of result.txt. the directory must start with /temp. sql: HareQL connectionKey: the Token of connections.	{       "fileSize": "0",       "responseTime": "1234",       "rowSize": "0",       "status": "success",       "auth": "true",       "exception": "exception"     }	fileSize: the file size of the result(bytes) responseTime: the response time(mini second) rowSize: the record numbers. status: the result(success/error) auth: verified status. exception: error messages(when error occurs)
Drop a table	POST	{hostname}/hbaseadmin/drop	{       "tableName": "table1",       "account": "@#\$%\$%",       "pwd": "%\$%&%(**",       "connectionKey": "24d56e"     }	tableName: the table name of HBase. connectionKey: the Token of connections.	{       "responseTime": "391",       "status": "success",       "auth": "true",       "exception": "exception"     }	responseTime: the response time(mini second). status: the result(success/error). exception: error messages(when error occurs). auth: verified status.
Create a Meta table	POST	{hostname}/haremeta/create	{       "hbaseTableName": "stana_host_c1",       "metaTableName": "stana_host_c1_meta",       "hbaseColumnNames": [         "key",         "cf:column1",         "cf:column2",         "cf:column3",         "cf:column4"       ],       "metaColumnNames": ["rowkey", "col1", "col2", "col3", "col4"],       "dataTypes": ["STRING", "STRING", "STRING", "STRING", "STRING"],       "account": "@#\$%\$%",       "pwd": "%\$%&%(**",       "connectionKey": "24d56e"     }	hbaseTableName: the table name of HBase and it is must created before. metaTableName: the table name of Meta table hbaseColumnNames: the column name of HBase metaColumnNames: the column name of Meta table dataTypes: the data type of column. Supports: STRING, DECIMAL, DOUBLE, FLOAT, BIGINT, INT, BOOLEAN, BINARY. connectionKey: the Token of connections. make sure the numbers of hbaseColumnNames, metaColumnNames, dataTypes must be the same.	{       "auth": "true",       "responseTime": "540",       "status": "success",       "exception": "exception"     }	responseTime: the response time(mini second). status: the result(success/error). exception: error messages(when error occurs). auth: verified status.
Drop a Meta table	POST	{hostname}/haremeta/drop	{       "metaTableName": "stana_host_c1_meta",       "account": "@#\$%\$%",       "pwd": "%\$%&%(**",       "connectionKey": "24d56e"     }	metaTableName: the table name of HBase. connectionKey: the Token of connections.	{       "auth": "true",       "responseTime": "870",       "status": "success",       "exception": "exception"     }	responseTime: the response time(mini second). status: the result(success/error). exception: error messages(when error occurs). auth: verified status.
Alter a Meta table	POST	{hostname}/haremeta/alter	{       "hbaseTableName": "stana_host_c1",       "metaTableName": "stana_host_c1_meta",       "hbaseColumnNames": [         "key",         "cf:column1",         "cf:column2",         "cf:column3",         "cf:column4"       ],       "metaColumnNames": ["rowkey", "col1", "col2", "col3", "col4"],       "dataTypes": ["STRING", "STRING", "STRING", "STRING", "STRING"],       "account": "@#\$%\$%",       "pwd": "%\$%&%(**",       "connectionKey": "24d56e"     }	hbaseTableName: the table name of HBase. metaTableName: the table name of Meta table hbaseColumnNames: the column name of HBase metaColumnNames: the column name of Meta table dataTypes: the data type of column. Supports: STRING, DECIMAL, DOUBLE, FLOAT, BIGINT, INT, BOOLEAN, BINARY. connectionKey: the Token of connections. make sure the numbers of hbaseColumnNames, metaColumnNames, dataTypes must be the same.	{       "auth": "true",       "responseTime": "540",       "status": "success",       "exception": "exception"     }	responseTime: the response time(mini second). status: the result(success/error). exception: error messages(when error occurs). auth: verified status.
Get the informations of Meta table	POST	{hostname}/haremeta/describe	{       "metaTableName": "stana_host_c1_meta",       "account": "@#\$%\$%",       "pwd": "%\$%&%(**",       "connectionKey": "24d56e"     }	metaTableName: the table name of Meta table connectionKey: the Token of connections.	{       "auth": "true",       "responseTime": "82",       "status": "success",       "dataTypes": [         "string",         "string",         "string",         "string"       ],       "hbaseColumnNames": [         "cf:column1",         "cf:column2",         "cf:column3",         "key"       ],       "hbaseTableName": "stana_host_bk1"     }	responseTime: the response time(mini second). status: the result(success/error). exception: error messages(when error occurs). auth: verified status. dataTypes: the data type of Meta table column. hbaseColumnNames: the column name of HBase. hbaseTableName: the table name of HBase.

When we use the upload function, we have to put several files into HDFS.

bulkload files	Format	Description
schemaFile	bulkloadseparator= bulkloadfileformat=NORMAL bulkloadtablename=table100 bulkloadoutputfolder=hdfs://host1:9000/result bulkloadcolumn=key,cf:column1,cf:column2 bulkloadcolumnrowkey=true,false bulkloadcolumnfamily=cf bulkloadexistheader=false	bulkloadseparator: the separator of column. bulkloadfileformat: file format, only text file. NORMAL. bulkloadtablename: the table name of HBase. bulkloadoutputfolder: the directory of result. bulkloadcolumn: the data column, no any special character, and must included a "key" column. bulkloadcolumnrowkey: is the rowkey, separate by ','. bulkloadcolumnfamily: column family. bulkloadexistheader: if the document includes the headers or not.
data	.key_row1_value column1_row1_value column2_row1_value .key_row2_value column1_row2_value column2_row2_value	

result	badlineinfo: 12 :key_row12_value column1_row12_value column2_row12_val		