

JSServer

 Apr 20, 2024 09:12

Supported Clients

SmartClient WebClient NGClient

Methods Summary

JSTableObject	createNewTable(tableName)	Creates in this server a new table with the specified name.
Boolean	dropTable(tableName)	Drops the table with the specified name from this server.
JSTableObject	getTable(tableName)	Returns a JSTable instance corresponding to the table with the specified name from this server.
Array	getTableNames()	Returns an array with the names of all tables in this server.
Boolean	isValid()	Get valid state for the server.
void	reloadDataModel()	Reloads the datamodel from the database, if changed externally or via rawSQL plugin.
Boolean	synchronizeWithDB(table)	Synchronizes a JSTable instance with the database.

Methods Details

createNewTable(tableName)
Creates in this server a new table with the specified name.

Parameters
[String](#) `tableName` The name of the table to create.

Returns
[JSTableObject](#) JSTableObject created table.

Supported Clients
SmartClient,WebClient,NGClient

Sample

```
var server = plugins.maintenance.getServer("example_data");
if (server)
{
    var table = server.createNewTable("new_table");
    if (table) {
        var pk = table.createNewColumn("new_table_id", JSColumn.INTEGER, 0);
        pk.rowIdentifierType = JSColumn.PK_COLUMN;
        if (server.synchronizeWithDB(table))
            application.output("New table created in the database.");
        else
            application.output("New table not created in database.");
    }
    else application.output("New table not created at all.");
}
```

dropTable(tableName)
Drops the table with the specified name from this server.

Parameters
[String](#) `tableName` The name of the table to drop.

Returns
[Boolean](#) boolean success.

Supported Clients
SmartClient,WebClient,NGClient

Sample

```
var server = plugins.maintenance.getServer("example_data");
if (server) {
    var result = server.dropTable("new_table");
    if (result)
        application.output("Table dropped.");
    else
        application.output("Table not dropped.");
}
```

getTable(tableName)

Returns a `JSTable` instance corresponding to the table with the specified name from this server.

Parameters

[String](#) `tableName` The name of the table to retrieve.

Returns

[JSTableObject](#) `JSTableObject` table.

Supported Clients

SmartClient, WebClient, NGClient

Sample

```
var server = plugins.maintenance.getServer("example_data");
if (server) {
    var table = server.getTable("employees");
    if (table) {
        var colNames = table.getColumnNames();
        application.output("Table has " + colNames.length + " columns.");
        for (var i=0; i<colNames.length; i++)
            application.output("Column " + i + ": " + colNames[i]);
    }
}
```

getTableNames()

Returns an array with the names of all tables in this server.

Returns

[Array](#) Array of String table names.

Supported Clients

SmartClient, WebClient, NGClient

Sample

```
var server = plugins.maintenance.getServer("example_data");
if (server) {
    var tableNames = server.getTableNames();
    application.output("There are " + tableNames.length + " tables.");
    for (var i=0; i<tableNames.length; i++)
        application.output("Table " + i + ": " + tableNames[i]);
}
else {
    plugins.dialogs.showInfoDialog("Attention", "Server 'example_data' cannot be found.", "OK");
}
```

isValid()

Get valid state for the server.

Returns

[Boolean](#) boolean valid state.

Supported Clients

SmartClient, WebClient, NGClient

Sample

```
var server = plugins.maintenance.getServer("example_data");
if (!server.isValid()) {
    application.output("Server not valid!");
}
```

reloadDataModel()

Reloads the datamodel from the database, if changed externally or via rawSQL plugin.

This call is not needed after a call to `synchronizeWithDB()`.

Supported Clients

SmartClient, WebClient, NGClient

Sample

```
var server = plugins.maintenance.getServer("example_data");
var result = plugins.rawSQL.executeSQL("example_data", null, 'CREATE TABLE raw_table (raw_table_id INTEGER)');
if (result) {
    application.output("Table created through rawSQL plugin.");
    if (server) {
        server.reloadDataModel();
        // All existing JSTableObject/JSColumn object references are invalid now! Use getTable to get
new ones.
        var table = server.getTable("raw_table");
        if (table) {
            var colNames = table.getColumnNames()
            application.output("Table has " + colNames.length + " columns.");
            for (var i=0; i<colNames.length; i++)
                application.output("Column " + i + ": " + colNames[i]);
        }
    }
}
else {
    application.output("Raw table creation failed: " + plugins.rawSQL.getException());
}
```

synchronizeWithDB(table)

Synchronizes a JSTable instance with the database. If columns were added to or removed from the JSTable instance, all these changes will now be persisted to the database.

Parameters

[JSTableObject](#) table A JSTableObject instance that should be synchronized.

Returns

[Boolean](#) boolean success.

Supported Clients

SmartClient, WebClient, NGClient

Sample

```
var server = plugins.maintenance.getServer("example_data");
if (server)
{
    var table = server.createNewTable("new_table");
    if (table) {
        var pk = table.createNewColumn("new_table_id", JSColumn.INTEGER, 0);
        pk.rowIdentifierType = JSColumn.PK_COLUMN;
        if (server.synchronizeWithDB(table))
            application.output("New table created in the database.");
        else
            application.output("New table not created in database.");
    }
    else application.output("New table not created at all.");
}
```