

- Optionally, clone SocketPro SDK at <https://github.com/udaparts/socketpro> for your evaluation.
- For coming demonstration with easy configuration, clone all files at <https://github.com/udaparts/nodejsdemos> into your computer.
- Distribute SocketPro libraries into system directory
 1. **Windows:** Copy all dlls at ../bin/win/x64 and x86 into c:\windows\system32 and syswow64, respectively.
 2. **Ubuntu (Other variants are similar):** Copy all libraries at ../bin/linux/ubuntu/ into /usr/lib.
- Set SocketPro MySQL/MariaDB DB server plugin
 1. Find MySQL/Mariadb database plugin directory by executing statement ***show variables where variable_name='plugin_dir'***.
 2. Copy MySQL/Mariadb plugin into database plugin directory
 - a. **Windows:** Copy smysql.dll at ../bin/(mysql8_0_11|mysql5_7_22|mariadb)/win64 or win86 into the above found database plugin directory. Execute the statement ***INSTALL PLUGIN UDAParts_SQL_Streaming SONAME 'smysql.dll'***.
 - b. **Linux:** Copy libsmysql.so at ../bin/(mysql8_0_11|mysql5_7_22|mariadb)/ into the above found database plugin directory. Execute the statement ***INSTALL PLUGIN UDAParts_SQL_Streaming SONAME 'libsmysql.so'***.
 3. Go to MySQL/MariaDB database data directory, which can be found by executing the query ***show variables where variable_name='datadir'***. Afterwards, you can find the two generated files, *sp_streaming_db_config.json* and *streaming_db.log*, for other advanced settings and error outputs, respectively. In case there is an error, the file *streaming_db.log* will very likely give you a hint to help. By this time, you can successfully run the test sample script now.
- Install sample database sakila
If your MySQL/MariaDB doesn't have sakila database installed, you can get it from the site https://github.com/datacharmer/test_db/tree/master/sakila.
- Configure SocketPro MySQL/MariaDB DB plugin for advanced features by modifying the file *sp_streaming_db_config.json*
The coming configurations are also optional. They are presented here for advanced features and other services.
 1. First of all, find entry ***services***, and change its string value to *sqlite;uasynqueue*. The MySQL/MariaDB database server plugin is going to load the two services (sqlite and server persistent queue). You can do so for other services. Each of services should be separated by the character semi-colon.
 2. Next, find entry ***monitored_tables***, and change its string value to *sakila.actor;sakila.country;sakila.category;sakila.language*. Doing so will force the MySQL/MariaDB database server plugin to monitor insert, update and delete trigger events for the four tables, actor, country, category and language. SocketPro use these trigger events for real-time cache at client or middle tier side.
 3. Stop MySQL/MariaDB database server, and restart it. By this time, the configuration file will be updated. In case there is an error, the log file *streaming_db.log* will help you out.

4. The following two steps are NOT necessary for MySQL 8 or later at all. However, if you use MySQL **5.7 or earlier**, or MariaDB, follow the below two steps to complete the previous setting 2:

Register a user defined function ***SetSQLStreamingPlugin*** by executing the statement ***CREATE FUNCTION SetSQLStreamingPlugin RETURNS INTEGER SONAME 'libmysql.so'*** and ***CREATE FUNCTION SetSQLStreamingPlugin RETURNS INTEGER SONAME 'mysql.dll'***, respectively on Linux and Windows platforms.

At last, call the user defined function ***SetSQLStreamingPlugin*** by executing a statement like ***select SetSQLStreamingPlugin('uid=root;pwd=Smash123')***. Here, the parameters *uid* and *pwd* represent user id and password, respectively.

5. In case you know C#, you can compile the real-time cache feature by compiling and running the test project at the directory `../socketpro/stream_sql/mysql/test_cache`.
- Copy SocketPro node.js adapter into node.js lib directory
SocketPro Node.js adapter, `nja.js` and `njadapter.node`, can be found at the directory `../bin/js` and `../bin/js/node-vx.y.z/(linux|win64|win86)`, respectively. Copy the two file into `../nodejs/lib/node`.
 - Copy all demo js files into the directory where node executable file is located.