KVDB-Manager

KVDB-Manager (key value database manager) – proxy manager which receives http query, creates connection to the key-value database and receives data from database.

What kind of database used – resolves at compile time by macro operations.

All settings for manager you can see in kvdbmanager.ini file.

Note: functions citrusleaf_init() and citrusleaf_shutdown() looks not so good with macro definitions. But it was the simple way for solving problem when this functions has multiple call. This function must be called only one time, when application starting and shutdown.

How it works

KVDBManager it's derived class from ASM::ServerApplication. This class must override virtual functions initialize, main, reinitialize and uninitialize. Other virtual functions may be determined only if you want set special options (like options for application or something else).

The main function in ASM::KVDBManager class contains main logic of application.

Main logic of application

Application listens port (port setted in the configuration file) and for each query application creates new thread (so works Poco::HTTPServer, but you can set max threads and other parameters for him in configuration file). Each query creates object “ASM::RequestHandler” (and in constructor of this object creates ORMFunctionalObject) and each object runs the function “ASM::RequestHandler::handleRequest”.

In depending on the value of the variable “request.type” different functions has been invoked (this functions you may see in the “Key-value DB data provider functional object messages protocol” file). When request type is detected (assume that it's a query for getting data from database) then we invoke polymorphic function 'get'.

If you want run application with default configuration file, which located in the same directory that binary file , you may not set additional parameter for configuration file (--config-file).

Default configuration file has the same name as the binary file, locates in the same directory and has .ini extension.

Otherwise , you must set path to the configuration file like this **./application --config-file=application_log.ini.**

Additional command line features you can see in help information : **./application -h** or **./application --help**

Minimum properties that has application :

--daemon Run application as a daemon.

--pidfile=path Write the process ID of the application to given file.

-ffile, --config-file=file Load configuration data from a file.

-h, --help Show help information.

For correctly work one of the several No-SQL databases must be started.

For example, if we use KVDBManager with Aerospike No-SQL database, we must start Aerospike server.

How to start Aerospike server :

/etc/init.d/citrusleaf start

How to stop Aerospike server :

/etc/init.d/citrusleaf stop

How to check the status :

sudo /etc/init.d/citrusleaf status

Configuration file for Aerospike :

/etc/citrusleaf/citrusleaf.conf

After start you may see **/usr/bin/cld --config-file /etc/citrusleaf/citrusleaf.conf** in top - it means that you have successful Aerospike start
.