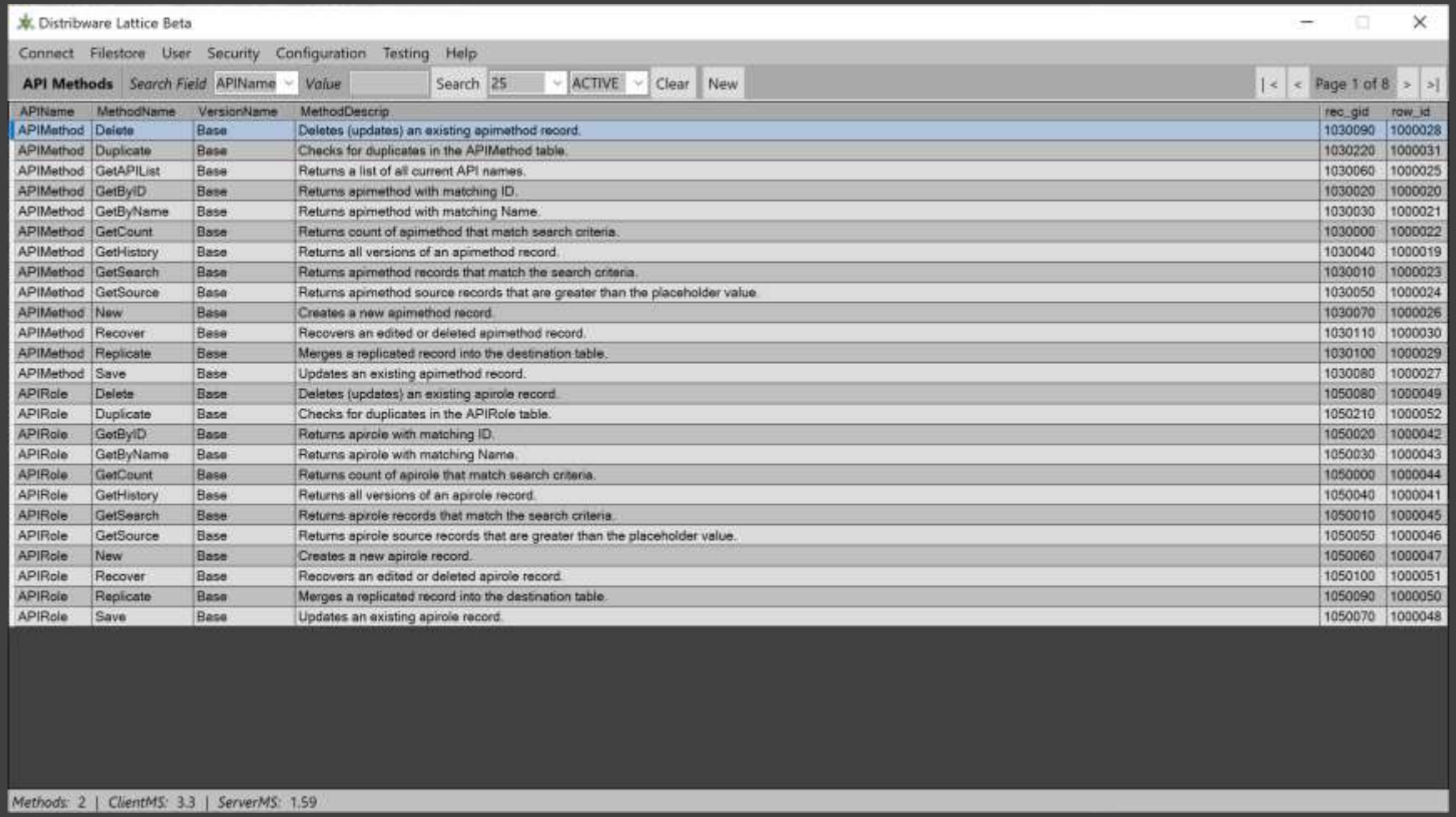
Monitoring

While logging stores data based on events that occur within a location (reactive), monitoring is based more on periodic proactive polling of the functionality in a location to insure that it is working correctly and also performing well. Much of the DGP monitoring functionality is actually a specialized form of testing.

Remote users of the Lattice application who are members of the RemoteMonitor role store end-to-end and server performance metrics in the LatticeMetrics table of the location to which they are connected. This is the same performance data shown in the status bar of the Lattice application during normal use, and is intended to collect performance data from production locations under realistic workloads. These metrics also prove that the specific functionality is working correctly along with the performance results.

Slow performance is often an indication of a "gray failure" in distributed systems. Reports, charts and graphs can be created using the data from the LatticeMetrics table to analyze trends in the data over time. The CountCheck and DupeCheck processes are another type of monitoring used to verify that the replicated data in each location is truly in synch with the other locations, and that no problems exist in the data. If problems are found, they can be repaired in place, usually as a background process.

Other API methods are used to monitor the count of the number of records stored in various database tables, to track the rate at which new records are being created. Simple reports, charts and graphs can then display these rates to admins in system dashboard UI's so they can observe problems and plan for the growth of a system.



APIName	MethodName	VersionName	MethodDescrip	rec_gid	row_id
APIMethod	Delete	Base	Deletes (updates) an existing apimethod record.	1030090	1000028
APIMethod	Duplicate	Base	Checks for duplicates in the APIMethod table.	1030220	1000031
APIMethod	GetAPIList	Base	Returns a list of all current API names.	1030060	1000025
APIMethod	GetByID	Base	Returns apimethod with matching ID.	1030020	1000020
APIMethod	GetByName	Base	Returns apimethod with matching Name.	1030030	1000021
APIMethod	GetCount	Base	Returns count of apimethod that match search criteria.	1030000	1000022
APIMethod	GetHistory	Base	Returns all versions of an apimethod record.	1030040	1000019
APIMethod	GetSearch	Base	Returns apimethod records that match the search criteria.	1030010	1000023
APIMethod	GetSource	Base	Returns apimethod source records that are greater than the placeholder value.	1030050	1000024
APIMethod	New	Base	Creates a new apimethod record.	1030070	1000026
APIMethod	Recover	Base	Recovers an edited or deleted apimethod record.	1030110	1000030
APIMethod	Replicate	Base	Merges a replicated record into the destination table.	1030100	1000029
APIMethod	Save	Base	Updates an existing apimethod record.	1030080	1000027
APIRole	Delete	Base	Deletes (updates) an existing apirole record.	1050080	1000049
APIRole	Duplicate	Base	Checks for duplicates in the APIRole table.	1050210	1000052
APIRole	GetByID	Base	Returns aprole with matching ID.	1050020	1000042
APIRole	GetByName	Base	Returns apirole with matching Name.	1050030	1000043
APIRole	GetCount	Base	Returns count of apirole that match search criteria.	1050000	1000044
APIRole	GetHistory	Base	Returns all versions of an apirole record.	1050040	1000041
APIRole	GetSearch	Base	Returns aprole records that match the search criteria.	1050010	1000045
APIRole	GetSource	Base	Returns apirole source records that are greater than the placeholder value.	1050050	1000046
APIRole	New	Base	Creates a new apirole record.	1050060	1000047
APIRole	Recover	Base	Recovers an edited or deleted apirole record.	1050100	1000051
APIRole	Replicate	Base	Merges a replicated record into the destination table.	1050090	1000050
APIRole	Save	Base	Updates an existing apirole record.	1050070	1000048

One example of DGP monitoring is the remote monitor functionality built into the Lattice application. Users that belong to the RemoteMonitor role save the performance metrics displayed in the status bar (along with other info) to the SysMetrics database of the location they are connected to. This monitors the functionality and performance of production system while in actual use.

Logging and Metrics Verification

1. *Errors and exceptions for each location of an environment can be viewed in the DGPErrors form of the Lattice application.*
2. *A tool like Splunk should be used to monitor the DGPErrors table and server Event Viewers to notify admins whenever problems occur. Dashboards can also be built using the error data from the DGPErrors table as well.*
3. *End-to-end and server performance data is stored in the LatticeMetrics table of the location and environment that the user is connected to. This is especially important for collecting performance metrics from production systems while they are in actual use, without adversely affecting those systems. Reports, charts, graphs can then be created to analyze the performance metrics over time. Dashboards can also be created to monitor the performance of the system using the LatticeMetrics data in real time.*
4. *When logging is turned on, data about the state and performance of automated processes are stored in the AutoWorkLog table. To manage the amount of log data generated and stored, this logging should only be turned on for a small number of processes at a time. The data in the AutoWorkLog table can be analyzed just like the LatticeMetrics data. Also, logging would be turned on for a period of time for a specific automated process to help "debug" problems or performance issues that have occurred.*