















-  Home (/)
-  Project (/alibaba/druid)
-  Activity (/alibaba/druid/activity)
-  Files (/alibaba/druid/tree/master)
-  Commits (/alibaba/druid/commits/master)
-  Graphs (/alibaba/druid/graphs/master)
-  MergeRequests (/alibaba/druid/merge_requests)
-  Milestones (/alibaba/druid/milestones)
-  Issues (/alibaba/druid/issues)
-  Wikis (/alibaba/druid/wikis/home)
-  Labels (/alibaba/druid/labels)

(<http://www.aliyun.com>) [管理控制台 \(http://home.console.aliyun.com\)](http://home.console.aliyun.com)
[登录 \(https://account.aliyun.com/login/login.htm\)](https://account.aliyun.com/login/login.htm)

Commit 84ed1699bf702dcae07ec57d2d69eb545ea89a8a
(/alibaba/druid/commit/84ed1699bf702dcae07ec57d2d69eb545ea89a8a)

Authored by  **wenshao (/u/wenshao)** 4 years ago

2 parents bde289e7 (/alibaba/druid/commit/bde289e71664e8f60d926bd884ce35e201543e50)
dcae4e22 (/alibaba/druid/commit/dcae4e2247ff254f7e7ef7ff2776b9e296c3dfcc)

 master (/alibaba/druid/tree/master)

Merge branch 'master' of https://github.com/wenshao/druid (https://github.com/wenshao/druid)

Showing **2 changed files** with **388 additions** and **385 deletions**

 **src/main/resources/support/http/resources/js/lang.js**

1	-\$.namespace("druid.lang");
2	-druid.lang = function() {
3	- var LANG_EN = 0;
4	- var LANG_CN = 1;
5	- var lang = {
6	- 'Index' : ['Index','首页'],

```
7      -      'DataSource' : ['DataSource', '数据源'],
8      -      'SQL' : ['SQL', 'SQL监控'],
9      -      'Wall' : ['Wall', 'SQL防火墙'],
10     -      'WebApp' : ['WebApp', 'Web应用'],
11     -      'WebURI' : ['WebURI', 'URI监控'],
12     -      'Web Session' : ['Web Session', 'Session监控'],
13     -      'Spring' : ['Spring', 'spring监控'],
14     -      'JSON API' : ['JSON API', 'JSON API'],
15     -      'ResetAll' : ['Reset All', '重置'],
16     -
17     -      'StatIndex' : ['Stat Index', '统计索引'],
18     -      'ViewJSONAPI' : ['View JSON API', '查看JSON API'],
19     -      'Version' : ['Version', '版本'],
20     -      'Drivers' : ['Drivers', '驱动'],
21     -      'ResetEnable' : ['ResetEnable', '是否允许重置'],
22     -      'ResetCount' : ['ResetCount', '重置次数'],
23     -      'JavaVersion' : ['JavaVersion', 'java版本'],
24     -      'JavaVMName' : ['JavaVMName', 'jvm名称'],
25     -      'JavaClassPath' : ['JavaClassPath', 'classpath路径'],
26     -      'StartTime' : ['StartTime', '启动时间'],
27     -
28     -      'DataSourceStatList' : ['DataSourceStat List', '数据源列表'],
29     -      'UserName' : ['UserName', '用户名'],
30     -      'URL' : ['URL', '连接地址'],
31     -      'DbType' : ['DbType', '数据库类型'],
32     -      'DriverClassName' : ['DriverClassName', '驱动类名'],
33     -      'FilterClassNames' : ['FilterClassNames', 'filter类名'],
34     -      'TestOnBorrow' : ['TestOnBorrow', '获取连接时检测'],
35     -      'TestWhileIdle' : ['TestWhileIdle', '空闲时检测'],
36     -      'TestOnReturn' : ['TestOnReturn', '连接放回连接池时检测'],
37     -      'InitialSize' : ['InitialSize', '初始化连接大小'],
38     -      'MinIdle' : ['MinIdle', '最小空闲连接数'],
39     -      'MaxActive' : ['MaxActive', '最大连接数'],
40     -      'QueryTimeout' : ['QueryTimeout', '查询超时时间'],
41     -      'TransactionQueryTimeout' : ['TransactionQueryTimeout', '事务查
42     询超时时间'],//
43     -      'LoginTimeout' : ['LoginTimeout', '登录超时时间'],//
44     -      'ValidConnectionCheckerClassName' :
45     ['ValidConnectionCheckerClassName', '连接有效性检查类名'],
46     -      'ExceptionSorterClassName' : ['ExceptionSorterClassName',
47     'ExceptionSorter类名'],//
48     -      'DefaultAutoCommit' : ['DefaultAutoCommit', '默认autocommit设
49     置'],
50     -      'DefaultReadOnly' : ['DefaultReadOnly', '默认只读设置'],
51     -      'DefaultTransactionIsolation' : ['DefaultTransactionIsolation',
52     '默认事务隔离'],//
53     -      'NotEmptyWaitCount' : ['NotEmptyWaitCount', '等待次数'],//
54     -      'NotEmptyWaitMillis' : ['NotEmptyWaitMillis', '等待最大时长'],//
55     -      'WaitThreadCount' : ['WaitThreadCount', '等待线程数量'],
56     -      'StartTransactionCount' : ['StartTransactionCount', '事务启动
57     数'],//
58     -      'TransactionHistogram' : ['TransactionHistogram', '事务时间分
59     布'],//
60     -      'PoolingCount' : ['PoolingCount', '池中连接数'],//
61     -      'PoolingPeak' : ['PoolingPeak', '池中连接数峰值'],//
```

```

55     - 'PoolingPeakTime' : ['PoolingPeakTime', '池中连接数峰值时
    间'],////
56     - 'ActiveCount' : ['ActiveCount', '活跃连接数'],
57     - 'ActivePeak' : ['ActivePeak', '活跃连接数峰值'],//
58     - 'ActivePeakTime' : ['ActivePeakTime', '活跃连接数峰值时间'],
59     - 'LogicConnectCount' : ['LogicConnectCount', '逻辑连接打开次数'],
60     - 'LogicCloseCount' : ['LogicCloseCount', '逻辑连接关闭次数'],
61     - 'LogicConnectErrorCount' : ['LogicConnectErrorCount', '逻辑连接
    错误次数'],
62     - 'PhysicalConnectCount' : ['PhysicalConnectCount', '物理连接打开次
    数'],
63     - 'PhysicalCloseCount' : ['PhysicalCloseCount', '物理关闭数量'],
64     - 'PhysicalConnectErrorCount' : ['PhysicalConnectErrorCount', '物
    理连接错误次数'],
65     - 'ExecuteCount' : ['ExecuteCount', '执行数'],
66     - 'ErrorCount' : ['ErrorCount', '错误数'],
67     - 'CommitCount' : ['CommitCount', '提交数'],
68     - 'RollbackCount' : ['RollbackCount', '回滚数'],
69     - 'PSCacheAccessCount' : ['PSCacheAccessCount', 'PSCache访问次
    数'],
70     - 'PSCacheHitCount' : ['PSCacheHitCount', 'PSCache命中次数'],
71     - 'PSCacheMissCount' : ['PSCacheMissCount', 'PSCache丢失次数'],//
72     - 'ConnectionHoldTimeHistogram' : ['ConnectionHoldTimeHistogram',
    '连接持有时间分布'],
73     - 'ClobOpenCount' : ['ClobOpenCount', 'Clob打开次数'],
74     - 'BlobOpenCount' : ['BlobOpenCount', 'Blob打开次数'],
75     -
76     - 'UserNameDesc' : ['Specify the username used when creating a new
    connection.', '指定建立连接是使用的用户名'],
77     - 'URLDesc' : ['The JDBC driver connection URL', 'jdbc连接字符
    串'],
78     - 'DbTypeDesc' : ['database type', '数据库类型'],
79     - 'DriverClassNameDesc' : ['The fully qualifed name of the JDBC
    driver class', 'jdbc驱动的类型'],
80     - 'FilterClassNamesDesc' : ['All the fully qualifed name of the
    filter classes', 'filter的类型'],
81     - 'TestOnBorrowDesc' : ['Test or not when borrow a connection',
    '是否在获得连接后检测其可用性'],
82     - 'TestWhileIdleDesc' : ['Test or not when a connection is idle
    for a while', '是否在连接空闲一段时间后检测其可用性'],
83     - 'TestOnReturnDesc' : ['Test or not when return a connection',
    '是否在连接放回连接池后检测其可用性'],
84     - 'InitialSizeDesc' : ['The size of datasource connections to
    create when initial a datasource', '连接池建立时创建的初始化连接数'],
85     - 'MinIdleDesc' : ['The minimum number of connections a pool
    should hold. ', '连接池中最小的活跃连接数'],
86     - 'MaxActiveDesc' : ['The maximum number of connections for a
    pool', '连接池中最大的活跃连接数'],
87     - 'QueryTimeoutDesc' : ['', '查询超时时间'],
88     - 'TransactionQueryTimeoutDesc' : ['', '事务查询超时时间'],
89     - 'LoginTimeoutDesc' : ['', ''],///
90     - 'ValidConnectionCheckerClassNameDesc' : ['', ''],
91     - 'ExceptionSorterClassNameDesc' : ['', ''],
92     - 'DefaultAutoCommitDesc' : ['', ''],
93     - 'DefaultReadOnlyDesc' : ['', ''],

```

```

94         'DefaultTransactionIsolationDesc' : ['', ''],
95         'NotEmptyWaitCountDesc' : ['Total times for wait to get a
connection', '获取连接时最多等待多少次'],//
96         'NotEmptyWaitMillisDesc' : ['Total millins for wait to get a
connection', '获取连接时最多等待多长时间'],
97         'WaitThreadCountDesc' : ['The current waiting thread count', '当
前等待获取连接的线程数'],
98         'StartTransactionCountDesc' : ['The count of start transaction',
'事务开始的个数'],
99         'TransactionHistogramDesc' : ['The histogram values of
transaction time, [0-10 ms, 10-100 ms, 100-1 s, 1-10 s, 10-100 s, >100 s]', '事
务运行时间分布, 分布区间为[0-10 ms, 10-100 ms, 100-1 s, 1-10 s, 10-100 s, >100
s]'],
100        'PoolingCountDesc' : ['The current usefull connection count',
'当前连接池中的数目'],//
101        'PoolingPeakDesc' : ['The usefull connection peak count', '连接
池中数目的峰值'],
102        'PoolingPeakTimeDesc' : ['The usefull connection peak time', '连
接池数目峰值出现的时间'],
103        'ActiveCountDesc' : ['The current active connection count', '当
前连接池中活跃连接数'],
104        'ActivePeakDesc' : ['The current active connection peak count',
'连接池中活跃连接数峰值'],
105        'ActivePeakTimeDesc' : ['The active connection peak time', '活跃
连接池峰值出现的时间'],
106        'LogicConnectCountDesc' : ['Total connect times from
datasource', '产生的逻辑连接建立总数'],
107        'LogicCloseCountDesc' : ['Total close connect times from
datasource', '产生的逻辑连接关闭总数'],
108        'LogicConnectErrorCountDesc' : ['Total connect error times', '产
生的逻辑连接出错总数'],
109        'PhysicalConnectCountDesc' : ['Create physical connection
count', '产生的物理连接建立总数'],
110        'PhysicalCloseCountDesc' : ['Close physical connnection count',
'产生的物理关闭总数'],
111        'PhysicalConnectErrorCountDesc' : ['Total physical connect error
times', '产生的物理连接失败总数'],
112        'ExecuteCountDesc' : ['', ''],
113        'ErrorCountDesc' : ['', ''],
114        'CommitCountDesc' : ['', ''],
115        'RollbackCountDesc' : ['', ''],
116        'PSCacheAccessCountDesc' : ['PerpareStatement access count',
'PSCache访问总数'],
117        'PSCacheHitCountDesc' : ['PerpareStatement hit count', 'PSCache
命中次数'],
118        'PSCacheMissCountDesc' : ['PerpareStatement miss count',
'PSCache丢失次数'],//
119        'ConnectionHoldTimeHistogramDesc' : ['The histogram values of
connection hold time, [0-1 ms, 1-10 ms, 10-100 ms, 100ms-1s, 1-10 s, 10-100 s,
100-1000 s, >1000 s]', '连接持有时间分布, 分布区间为[0-1 ms, 1-10 ms, 10-100 ms,
100ms-1s, 1-10 s, 10-100 s, 100-1000 s, >1000 s]'],
120        'ClobOpenCountDesc' : ['', 'Clob打开数'],
121        'BlobOpenCountDesc' : ['', 'Blob打开数'],
122        /**spring-detail*/
123        'Class' : ['Class', 'Class'],

```

```

124 -         'Method' : ['Method', 'Method'],
125 -         'ExecuteErrorCount' : ['ExecuteErrorCount', '执行出错数'],
126 -         'ExecuteTimeMillis' : ['ExecuteTimeMillis', '执行时间'],
127 -         'RunningCount' : ['RunningCount', '执行中'],
128 -         'ConcurrentMax' : ['ConcurrentMax', '最大并发'],
129 -         'JdbcExecuteCount' : ['JdbcExecuteCount', 'Jdbc执行数'],
130 -         'JdbcExecuteErrorCount' : ['JdbcExecuteErrorCount', 'Jdbc出错
131 数'],
132 -         'JdbcExecuteTimeMillis' : ['JdbcExecuteTimeMillis', 'Jdbc时间'],
133 -         'JdbcCommitCount' : ['JdbcCommitCount', '事务提交数'],
134 -         'JdbcRollbackCount' : ['JdbcRollbackCount', '事务回滚数'],
135 -         'JdbcFetchRowCount' : ['JdbcFetchRowCount', '读取行数'],
136 -         'JdbcUpdateCount' : ['JdbcUpdateCount', '更新行数'],
137 -         'JdbcPoolConnectionOpenCount' : ['JdbcPoolConnectionOpenCount',
138 '连接池获取连接次数'],
139 -         'JdbcPoolConnectionCloseCount' :
140 ['JdbcPoolConnectionCloseCount', '连接池关闭连接次数'],
141 -         'JdbcResultSetOpenCount' : ['JdbcResultSetOpenCount', 'ResultSet
142 打开次数'],
143 -         'JdbcResultSetCloseCount' : ['JdbcResultSetCloseCount',
144 'ResultSet关闭次数'],
145 -         /**sql-detail*/
146 -         'ParseView' : ['ParseView', '解析信息'],//
147 -         'Tables' : ['Tables', '表'],
148 -         'Fields' : ['Fields', '字段'],
149 -         'Cotions' : ['Cotions', '条件'],
150 -         'Relationships' : ['Relationships', '关联'],
151 -         'OrderByColumns' : ['OrderByColumns', '排序字段'],
152 -         'LastSlowView' : ['LastSlowView', '最后慢查询'],
153 -         'MaxTimespan' : ['MaxTimespan', '最慢'],
154 -         'MaxTimespanOccurTime' : ['MaxTimespanOccurTime', '最慢发生
155 时间'],
156 -         'LastSlowParameters' : ['LastSlowParameters', '最后慢查询参数'],
157 -         'LastErrorView' : ['LastErrorView', '最后错误视图'],
158 -         'LastErrorMessage' : ['LastErrorMessage', '最后错误信息'],
159 -         'LastErrorClass' : ['LastErrorClass', '最后错误类'],
160 -         'LastErrorTime' : ['LastErrorTime', '最后错误时间'],
161 -         'LastErrorStackTrace' : ['LastErrorStackTrace', '最后错误堆栈'],
162 -         'OtherView' : ['OtherView', '其他信息'],
163 -         'BatchSizeMax' : ['BatchSizeMax', '批处理最大值'],////
164 -         'BatchSizeTotal' : ['BatchSizeTotal', '批处理总数'],//
165 -         'ReaderOpenCount' : ['ReaderOpenCount', 'reader打开次数'],//
166 -         'InputStreamOpenCount' : ['InputStreamOpenCount', 'inputstream打
167 开次数'],
168 -         'ReadStringLength' : ['ReadStringLength', '读取字符串长度'],//
169 -         'ReadBytesLength' : ['ReadBytesLength', '读取字节长度'],//
170 -         'ExecHisto': ['ExecHisto', '执行时间分布'],
171 -         'ExecRsHisto': ['ExecRsHisto', '执行+RS时分布'],
172 -         'FetchRowHisto': ['FetchRowHisto', '读取行数分布'],
173 -         'UpdateHisto': ['UpdateHisto', '更新行数分布'],
174 -         'count1ms': ['count of 0-1 ms', '0-1毫秒次数'],
175 -         'count10ms': ['count of 1-10 ms', '1-10毫秒次数'],
176 -         'count100ms': ['count of 10-100 ms', '10-100毫秒次数'],

```

```
172 - 'count1s': ['count of 100ms-1s', '100-1000毫秒次数'],
173 - 'count10s': ['count of 1-10 s', '1-10秒次数'],
174 - 'count100s': ['count of 10-100 s', '10-100秒次数'],
175 - 'count1000s': ['count of 100-1000 s', '100-1000秒次数'],
176 - 'countBg1000s': ['count of >1000 s', '大于1000秒次数'],
177 -
178 - 'fetch0': ['count of 0 FetchRow', '读取行数为0'],
179 - 'fetch9': ['count of 1-9 FetchRow', '读取行数1-9'],
180 - 'fetch99': ['count of 10-99 FetchRow', '读取行数10-99'],
181 - 'fetch999': ['count of 100-999 FetchRow', '读取行数100-999'],
182 - 'fetch9999': ['count of 1000-9999 FetchRow', '读取行数1000-9999'],
183 - 'fetch99999': ['count of >9999 FetchRow', '读取行数大于9999'],
184 -
185 - 'update0': ['count of 0 UpdateCount', '更新行数为0'],
186 - 'update9': ['count of 1-9 UpdateCount', '更新行数1-9'],
187 - 'update99': ['count of 10-99 UpdateCount', '更新行数10-99'],
188 - 'update999': ['count of 100-999 UpdateCount', '更新行数100-999'],
189 - 'update9999': ['count of 1000-9999 UpdateCount', '更新行数1000-
9999'],
190 - 'update99999': ['count of >9999 UpdateCount', '更新行数大于9999'],
191 - /**wall*/
192 - 'CheckCount': ['CheckCount', '检查次数'],
193 - 'HardCheckCount': ['HardCheckCount', '硬检查次数'],
194 - 'ViolationCount': ['ViolationCount', '非法次数'],
195 - 'BlackListHitCount': ['BlackListHitCount', '黑名单命中次数'],
196 - 'BlackListSize': ['BlackListSize', '黑名单长度'],
197 - 'WhiteListHitCount': ['WhiteListHitCount', '白名单命中次数'],
198 - 'WhiteListSize': ['WhiteListSize', '白名单长度'],
199 - 'SyntaxErrorCount': ['SyntaxErrorCount', '语法错误次数'],
200 - 'TableName': ['TableName', '表名'],
201 -
202 - 'SelectCount': ['SelectCount', 'Select数'],
203 - 'InsertCount': ['InsertCount', 'Insert数'],
204 - 'UpdateCount': ['UpdateCount', 'Update数'],
205 - 'DeleteCount': ['DeleteCount', 'Delete数'],
206 - 'TruncateCount': ['TruncateCount', 'Truncate数'],
207 - 'CreateCount': ['CreateCount', 'Create数'],
208 - 'AlterCount': ['AlterCount', 'Alter数'],
209 - 'DropCount': ['DropCount', 'Drop数'],
210 - 'ReplaceCount': ['ReplaceCount', 'Replace数'],
211 - 'DeleteDataCount': ['DeleteDataCount', '删除数据行数'],
212 - 'UpdateDataCount': ['UpdateDataCount', '更新数据行数'],
213 - 'FetchRowCount': ['FetchRowCount', '读取行数'],
214 - /**session-detail*/
215 - 'SESSIONID': ['SESSIONID', 'SESSIONID'],
216 - 'UserAgent': ['UserAgent', 'UserAgent'],
217 - 'Principal': ['Principal', 'Principal'],
218 - 'CreateTime': ['CreateTime', '创建时间'],
219 - 'LastAccessTime': ['LastAccessTime', '最后访问时间'],
220 - 'RemoteAddress': ['RemoteAddress', '访问ip地址'],
221 - 'RequestCount': ['RequestCount', '请求次数'],
222 - 'RequestTimeMillisTotal': ['RequestTimeMillisTotal', '总共请求时
间'],
223 - 'RequestInterval': ['RequestInterval', '请求间隔'],
224 - /**weburi-detail*/
```



```

225     -         'RequestTimeMillis': ['RequestTimeMillis', '请求时间'],
226     -         'JdbcExecutePeak': ['JdbcExecutePeak', 'jdbc执行峰值'],
227     -         'JdbcFetchRowPeak': ['JdbcFetchRowPeak', 'jdbc查询取回行数峰值'],
228     -         'JdbcUpdatePeak': ['JdbcUpdatePeak', 'jdbc更新峰值']
229     -
230     -     };
231     -
232     -     var COOKIE_LANG_NAME = "cookie_lang";
233     -
234     -     function log(str) {
235     -         if (typeof (console) != 'undefined' && typeof(console.log) !=
236     -         'undefined') {
237     -             console.log(str);
238     -         } else {
239     -             $('body').append('<input type="hidden" value="' + str +
240     -             " />");
241     -         }
242     -     }
243     -
244     -     function setCookie(name,value,expires,path,domain,secure)
245     -     {
246     -         var expDays = expires*24*60*60*1000;
247     -         var expDate = new Date();
248     -         expDate.setTime(expDate.getTime()+expDays);
249     -         var expString = ((expires==null) ? "":
250     -         (";expires="+expDate.toGMTString()));
251     -         var pathString = ((path==null) ? "": (";path="+path));
252     -         var domainString = ((domain==null) ? "": (";domain="+domain));
253     -         var secureString = ((secure==true) ? ";secure": "");
254     -         document.cookie = name + "=" + escape(value) + expString +
255     -         pathString + domainString + secureString;
256     -     }
257     -
258     -     function getCookie(name)
259     -     {
260     -         var result = null;
261     -         var myCookie = document.cookie + ";";
262     -         var searchName = name + "=";
263     -         var startOfCookie = myCookie.indexOf(searchName);
264     -         var endOfCookie;
265     -         if (startOfCookie != -1)
266     -         {
267     -             startOfCookie += searchName.length;
268     -             endOfCookie = myCookie.indexOf(";",startOfCookie);
269     -             result =
270     -             unescape(myCookie.substring(startOfCookie,endOfCookie));
271     -         }
272     -         return result;
273     -     }
274     -
275     -     function setText($obj) {
276     -         var key = $obj.attr('langKey');
277     -         if (typeof(lang[key]) != 'undefined') {
278     -             var text = lang[key][druid.lang.langNow];
279     -             $obj.text(lang[key][druid.lang.langNow]);

```

```

275         } else {
276             log('key [' + key + '] not found');
277         }
278     }
279     function setTitle($obj) {
280         var key = $obj.attr('langKey');
281         if (typeof(lang[key]) != 'undefined') {
282             var title = lang[key][druid.lang.langNow];
283             $obj.attr('title', title);
284         } else {
285             log('key [' + key + '] not found');
286         }
287     }
288
289     return {
290         langNow : LANG_CN,
291         EVENT_LOAD_FINISHED : 'loadFinished',
292         init : function(langNow) {
293             if (typeof(langNow) != 'undefined') {
294                 this.setLangType(langNow);
295             } else {
296                 var langInCookie = getCookie(COOKIE_LANG_NAME);
297                 if (langInCookie == LANG_CN || langInCookie ==
LANG_EN) {
298                     this.setLangType(langInCookie);
299                 }
300             }
301             $(document).on(this.EVENT_LOAD_FINISHED, '.lang',
function() {
302                 log('load lang');
303                 setText($(this));
304             });
305             $(document).on(this.EVENT_LOAD_FINISHED, '.langTitle',
function() {
306                 log('load title');
307                 setTitle($(this));
308             });
309             this.trigger();
310
311             $(document).on('click', '.langSelector', function() {
312                 var langSelected = $(this).attr('langNow');
313                 druid.lang.setLangType(langSelected);
314                 druid.lang.trigger();
315                 return false;
316             });
317         },
318         setLangType : function (langNow) {
319             this.langNow = langNow;
320             setCookie(COOKIE_LANG_NAME, langNow, 30, '/');
321         },
322         getLangType : function () {
323             return this.langNow;
324         },
325         show : function($parent) {
326             var $obj;

```



```

327         -             var $objTitle;
328         -             if ($parent) {
329         -                 $obj = $parent.find('.lang');
330         -                 $objTitle = $parent.find('.langTitle');
331         -             } else {
332         -                 $obj = $('<div>.lang');
333         -                 $objTitle = $('<div>.langTitle');
334         -             }
335         -             $obj.each(function() {
336         -                 setText($(this));
337         -             });
338         -             $objTitle.each(function() {
339         -                 setTitle($(this));
340         -             });
341         -         },
342         -         trigger : function() {
343         -             log('to load lang now');
344         -             $('<div>.lang').trigger(this.EVENT_LOAD_FINISHED);//触发语言
显示事件
345         -             $('<div>.langTitle').trigger(this.EVENT_LOAD_FINISHED);//触发
语言显示事件
346         -             }
347         -         }
348         -     }());
1 +$.namespace("druid.lang");
2 +druid.lang = function() {
3 +     var LANG_EN = 0;
4 +     var LANG_CN = 1;
5 +     var lang = {
6 +         'Index' : ['Index','首页'],
7 +         'DataSource' : ['DataSource', '数据源'],
8 +         'SQL' : ['SQL','SQL监控'],
9 +         'Wall' : ['Wall', 'SQL防火墙'],
10 +         'WebApp' : ['WebApp', 'Web应用'],
11 +         'WebURI' : ['WebURI', 'URI监控'],
12 +         'Web Session' : ['Web Session', 'Session监控'],
13 +         'Spring' : ['Spring', 'spring监控'],
14 +         'JSON API' : ['JSON API', 'JSON API'],
15 +         'ResetAll' : ['Reset All', '重置'],
16 +
17 +         'StatIndex' : ['Stat Index', '统计索引'],
18 +         'ViewJSONAPI' : ['View JSON API','查看JSON API'],
19 +         'Version' : ['Version', '版本'],
20 +         'Drivers' : ['Drivers', '驱动'],
21 +         'ResetEnable' : ['ResetEnable', '是否允许重置'],
22 +         'ResetCount' : ['ResetCount', '重置次数'],
23 +         'JavaVersion' : ['JavaVersion', 'java版本'],
24 +         'JavaVMName' : ['JavaVMName', 'jvm名称'],
25 +         'JavaClassPath' : ['JavaClassPath', 'classpath路径'],
26 +         'StartTime' : ['StartTime', '启动时间'],
27 +
28 +         'DataSourceStatList' : ['DataSourceStat List', '数据源列表'],
29 +         'UserName' : ['UserName', '用户名'],
30 +         'URL' : ['URL', '连接地址'],
31 +         'DbType' : ['DbType', '数据库类型'],

```

```

32 +         'DriverClassName' : ['DriverClassName', '驱动类名'],
33 +         'FilterClassNames' : ['FilterClassNames', 'filter类名'],
34 +         'TestOnBorrow' : ['TestOnBorrow', '获取连接时检测'],
35 +         'TestWhileIdle' : ['TestWhileIdle', '空闲时检测'],
36 +         'TestOnReturn' : ['TestOnReturn', '连接放回连接池时检测'],
37 +         'InitialSize' : ['InitialSize', '初始化连接大小'],
38 +         'MinIdle' : ['MinIdle', '最小空闲连接数'],
39 +         'MaxActive' : ['MaxActive', '最大连接数'],
40 +         'QueryTimeout' : ['QueryTimeout', '查询超时时间'],
41 +         'TransactionQueryTimeout' : ['TransactionQueryTimeout', '事务查
    询超时时间'],//
42 +         'LoginTimeout' : ['LoginTimeout', '登录超时时间'],//
43 +         'ValidConnectionCheckerClassName' :
    ['ValidConnectionCheckerClassName', '连接有效性检查类名'],
44 +         'ExceptionSorterClassName' : ['ExceptionSorterClassName',
    'ExceptionSorter类名'],//
45 +         'DefaultAutoCommit' : ['DefaultAutoCommit', '默认autocommit设
    置'],
46 +         'DefaultReadOnly' : ['DefaultReadOnly', '默认只读设置'],
47 +         'DefaultTransactionIsolation' : ['DefaultTransactionIsolation',
    '默认事务隔离'],//
48 +         'NotEmptyWaitCount' : ['NotEmptyWaitCount', '等待次数'],//
49 +         'NotEmptyWaitMillis' : ['NotEmptyWaitMillis', '等待最大时长'],//
50 +         'WaitThreadCount' : ['WaitThreadCount', '等待线程数量'],
51 +         'StartTransactionCount' : ['StartTransactionCount', '事务启动
    数'],//
52 +         'TransactionHistogram' : ['TransactionHistogram', '事务时间分
    布'],//
53 +         'PoolingCount' : ['PoolingCount', '池中连接数'],//
54 +         'PoolingPeak' : ['PoolingPeak', '池中连接数峰值'],//
55 +         'PoolingPeakTime' : ['PoolingPeakTime', '池中连接数峰值时
    间'],////
56 +         'ActiveCount' : ['ActiveCount', '活跃连接数'],
57 +         'ActivePeak' : ['ActivePeak', '活跃连接数峰值'],//
58 +         'ActivePeakTime' : ['ActivePeakTime', '活跃连接数峰值时间'],
59 +         'LogicConnectCount' : ['LogicConnectCount', '逻辑连接打开次数'],
60 +         'LogicCloseCount' : ['LogicCloseCount', '逻辑连接关闭次数'],
61 +         'LogicConnectErrorCount' : ['LogicConnectErrorCount', '逻辑连接
    错误次数'],
62 +         'PhysicalConnectCount' : ['PhysicalConnectCount', '物理连接打开次
    数'],
63 +         'PhysicalCloseCount' : ['PhysicalCloseCount', '物理关闭数量'],
64 +         'PhysicalConnectErrorCount' : ['PhysicalConnectErrorCount', '物
    理连接错误次数'],
65 +         'ExecuteCount' : ['ExecuteCount', '执行数'],
66 +         'ErrorCount' : ['ErrorCount', '错误数'],
67 +         'CommitCount' : ['CommitCount', '提交数'],
68 +         'RollbackCount' : ['RollbackCount', '回滚数'],
69 +         'PSCacheAccessCount' : ['PSCacheAccessCount', 'PSCache访问次
    数'],
70 +         'PSCacheHitCount' : ['PSCacheHitCount', 'PSCache命中次数'],
71 +         'PSCacheMissCount' : ['PSCacheMissCount', 'PSCache丢失次数'],//
72 +         'ConnectionHoldTimeHistogram' : ['ConnectionHoldTimeHistogram',
    '连接持有时间分布'],
73 +         'ClobOpenCount' : ['ClobOpenCount', 'Clob打开次数'],

```

```

74 +             'BlobOpenCount' : ['BlobOpenCount', 'Blob打开次数'],
75 +
76 +             'UserNameDesc' : ['Specify the username used when creating a new
connection.', '指定建立连接是使用的用户名'],
77 +             'URLDesc' : ['The JDBC driver connection URL', 'jdbc连接字
串'],
78 +             'DbTypeDesc' : ['database type', '数据库类型'],
79 +             'DriverClassNameDesc' : ['The fully qualifed name of the JDBC
driver class', 'jdbc驱动的类型'],
80 +             'FilterClassNamesDesc' : ['All the fully qualifed name of the
filter classes', 'filter的类型'],
81 +             'TestOnBorrowDesc' : ['Test or not when borrow a connection',
'是否在获得连接后检测其可用性'],
82 +             'TestWhileIdleDesc' : ['Test or not when a connection is idle
for a while', '是否在连接空闲一段时间后检测其可用性'],
83 +             'TestOnReturnDesc' : ['Test or not when return a connection',
'是否在连接放回连接池后检测其可用性'],
84 +             'InitialSizeDesc' : ['The size of datasource connections to
create when initial a datasource', '连接池建立时创建的初始化连接数'],
85 +             'MinIdleDesc' : ['The minimum number of connections a pool
should hold. ', '连接池中最小的活跃连接数'],
86 +             'MaxActiveDesc' : ['The maximum number of connections for a
pool', '连接池中最大的活跃连接数'],
87 +             'QueryTimeoutDesc' : ['', '查询超时时间'],
88 +             'TransactionQueryTimeoutDesc' : ['', '事务查询超时时间'],
89 +             'LoginTimeoutDesc' : ['', ''],///
90 +             'ValidConnectionCheckerClassNameDesc' : ['', ''],
91 +             'ExceptionSorterClassNameDesc' : ['', ''],
92 +             'DefaultAutoCommitDesc' : ['', ''],
93 +             'DefaultReadOnlyDesc' : ['', ''],
94 +             'DefaultTransactionIsolationDesc' : ['', ''],
95 +             'NotEmptyWaitCountDesc' : ['Total times for wait to get a
connection', '获取连接时最多等待多少次'],//
96 +             'NotEmptyWaitMillisDesc' : ['Total millins for wait to get a
connection', '获取连接时最多等待多长时间'],
97 +             'WaitThreadCountDesc' : ['The current waiting thread count', '当
前等待获取连接的线程数'],
98 +             'StartTransactionCountDesc' : ['The count of start transaction',
'事务开始的个数'],
99 +             'TransactionHistogramDesc' : ['The histogram values of
transaction time, [0-10 ms, 10-100 ms, 100-1 s, 1-10 s, 10-100 s, >100 s]', '事
务运行时间分布, 分布区间为[0-10 ms, 10-100 ms, 100-1 s, 1-10 s, 10-100 s, >100
s]'],
100 +             'PoolingCountDesc' : ['The current usefull connection count',
'当前连接池中的数目'],//
101 +             'PoolingPeakDesc' : ['The usefull connection peak count', '连接
池中数目的峰值'],
102 +             'PoolingPeakTimeDesc' : ['The usefull connection peak time', '连
接池数目峰值出现的时间'],
103 +             'ActiveCountDesc' : ['The current active connection count', '当
前连接池中活跃连接数'],
104 +             'ActivePeakDesc' : ['The current active connection peak count',
'连接池中活跃连接数峰值'],
105 +             'ActivePeakTimeDesc' : ['The active connection peak time', '活跃
连接池峰值出现的时间'],

```

```

106 +         'LogicConnectCountDesc' : ['Total connect times from
      datasource', '产生的逻辑连接建立总数'],
107 +         'LogicCloseCountDesc' : ['Total close connect times from
      datasource', '产生的逻辑连接关闭总数'],
108 +         'LogicConnectErrorCountDesc' : ['Total connect error times', '产
      生的逻辑连接出错总数'],
109 +         'PhysicalConnectCountDesc' : ['Create physical connnection
      count', '产生的物理连接建立总数'],
110 +         'PhysicalCloseCountDesc' : ['Close physical connnection count',
      '产生的物理关闭总数'],
111 +         'PhysicalConnectErrorCountDesc' : ['Total physical connect error
      times', '产生的物理连接失败总数'],
112 +         'ExecuteCountDesc' : ['', ''],
113 +         'ErrorCountDesc' : ['', ''],
114 +         'CommitCountDesc' : ['', ''],
115 +         'RollbackCountDesc' : ['', ''],
116 +         'PSCacheAccessCountDesc' : ['PerpareStatement access count',
      'PSCache访问总数'],
117 +         'PSCacheHitCountDesc' : ['PerpareStatement hit count', 'PSCache
      命中次数'],
118 +         'PSCacheMissCountDesc' : ['PerpareStatement miss count',
      'PSCache丢失次数'],//
      +         'ConnectionHoldTimeHistogramDesc' : ['The histogram values of
      connection hold time, [0-1 ms, 1-10 ms, 10-100 ms, 100ms-1s, 1-10 s, 10-100 s,
119 100-1000 s, >1000 s]', '连接持有时间分布, 分布区间为[0-1 ms, 1-10 ms, 10-100 ms,
      100ms-1s, 1-10 s, 10-100 s, 100-1000 s, >1000 s]'],
120 +         'ClobOpenCountDesc' : ['', 'Clob打开数'],
121 +         'BlobOpenCountDesc' : ['', 'Blob打开数'],
122 +         /**spring-detail*/
123 +         'Class' : ['Class', 'Class'],
124 +         'Method' : ['Method', 'Method'],
125 +         'ExecuteErrorCount' : ['ExecuteErrorCount', '执行出错数'],
126 +         'ExecuteTimeMillis' : ['ExecuteTimeMillis', '执行时间'],
127 +         'RunningCount' : ['RunningCount', '执行中'],
128 +         'ConcurrentMax' : ['ConcurrentMax', '最大并发'],
129 +         'JdbcExecuteCount' : ['JdbcExecuteCount', 'Jdbc执行数'],
130 +         'JdbcExecuteErrorCount' : ['JdbcExecuteErrorCount', 'Jdbc出错
      数'],
131 +         'JdbcExecuteTimeMillis' : ['JdbcExecuteTimeMillis', 'Jdbc时间'],
132 +         'JdbcCommitCount' : ['JdbcCommitCount', '事务提交数'],
133 +         'JdbcRollbackCount' : ['JdbcRollbackCount', '事务回滚数'],
134 +         'JdbcFetchRowCount' : ['JdbcFetchRowCount', '读取行数'],
135 +         'JdbcUpdateCount' : ['JdbcUpdateCount', '更新行数'],
136 +         'JdbcPoolConnectionOpenCount' : ['JdbcPoolConnectionOpenCount',
      '连接池获取连接次数'],
      +         'JdbcPoolConnectionCloseCount' :
137 ['JdbcPoolConnectionCloseCount', '连接池关闭连接次数'],
      +         'JdbcResultSetOpenCount' : ['JdbcResultSetOpenCount', 'ResultSet
138 打开次数'],
      +         'JdbcResultSetCloseCount' : ['JdbcResultSetCloseCount',
139 'ResultSet关闭次数'],
140 +         /**sql-detail*/
141 +         'ParseView' : ['ParseView', '解析信息'],//
142 +         'Tables' : ['Tables', '表'],
143 +         'Fields' : ['Fields', '字段'],

```

```

144 +         'Codings' : ['Codings', '条件'],
145 +         'Relationships' : ['Relationships', '关联'],
146 +         'OrderByColumns' : ['OrderByColumns', '排序字段'],
147 +         'LastSlowView' : ['LastSlowView', '最后慢查询'],
148 +         'MaxTimespan' : ['MaxTimespan', '最慢'],
149 +         'MaxTimespanOccurTime' : ['MaxTimespanOccurTime', '最慢发生时间'],
150 +         'LastSlowParameters' : ['LastSlowParameters', '最后慢查询参数'],
151 +         'LastErrorView' : ['LastErrorView', '最后错误视图'],
152 +         'LastErrorMessage' : ['LastErrorMessage', '最后错误信息'],
153 +         'LastErrorClass' : ['LastErrorClass', '最后错误类'],
154 +         'LastErrorTime' : ['LastErrorTime', '最后错误时间'],
155 +         'LastErrorStackTrace' : ['LastErrorStackTrace', '最后错误堆栈'],
156 +         'OtherView' : ['OtherView', '其他信息'],
157 +         'BatchSizeMax' : ['BatchSizeMax', '批处理最大值'],////
158 +         'BatchSizeTotal' : ['BatchSizeTotal', '批处理总数'],//
159 +         'ReaderOpenCount' : ['ReaderOpenCount', 'reader打开次数'],//
160 +         'InputStreamOpenCount' : ['InputStreamOpenCount', 'inputstream打开次数'],
161 +         'ReadStringLength' : ['ReadStringLength', '读取字符串长度'],//
162 +         'ReadBytesLength' : ['ReadBytesLength', '读取字节长度'],//
163 +
164 +         'ExecHisto': ['ExecHisto', '执行时间分布'],
165 +         'ExecRsHisto': ['ExecRsHisto', '执行+RS时分布'],
166 +         'FetchRowHisto': ['FetchRowHisto', '读取行数分布'],
167 +         'UpdateHisto': ['UpdateHisto', '更新行数分布'],
168 +         'InTransactionCount' : ['InTransactionCount', '事务中次数'],
169 +         'MaxTimespanDesc' : ['Execute Time Millis Max', '最长的执行时间'],
170 +         'InTransactionCountDesc' : ['Execute In Transaction Count', '在事务中运行的次数'],
171 +
172 +         'count1ms': ['count of 0-1 ms', '0-1毫秒次数'],
173 +         'count10ms': ['count of 1-10 ms', '1-10毫秒次数'],
174 +         'count100ms': ['count of 10-100 ms', '10-100毫秒次数'],
175 +         'count1s': ['count of 100ms-1s', '100-1000毫秒次数'],
176 +         'count10s': ['count of 1-10 s', '1-10秒次数'],
177 +         'count100s': ['count of 10-100 s', '10-100秒次数'],
178 +         'count1000s': ['count of 100-1000 s', '100-1000秒次数'],
179 +         'countBg1000s': ['count of >1000 s', '大于1000秒次数'],
180 +
181 +         'fetch0': ['count of 0 FetchRow', '读取行数为0'],
182 +         'fetch9': ['count of 1-9 FetchRow', '读取行数1-9'],
183 +         'fetch99': ['count of 10-99 FetchRow', '读取行数10-99'],
184 +         'fetch999': ['count of 100-999 FetchRow', '读取行数100-999'],
185 +         'fetch9999': ['count of 1000-9999 FetchRow', '读取行数1000-9999'],
186 +         'fetch99999': ['count of >9999 FetchRow', '读取行数大于9999'],
187 +
188 +         'update0': ['count of 0 UpdateCount', '更新行数为0'],
189 +         'update9': ['count of 1-9 UpdateCount', '更新行数1-9'],
190 +         'update99': ['count of 10-99 UpdateCount', '更新行数10-99'],
191 +         'update999': ['count of 100-999 UpdateCount', '更新行数100-999'],
192 +         'update9999': ['count of 1000-9999 UpdateCount', '更新行数1000-9999'],
193 +         'update99999': ['count of >9999 UpdateCount', '更新行数大于9999'],
194 +         /**wall*/

```

```

195 +         'CheckCount': ['CheckCount', '检查次数'],
196 +         'HardCheckCount': ['HardCheckCount', '硬检查次数'],
197 +         'ViolationCount': ['ViolationCount', '非法次数'],
198 +         'BlackListHitCount': ['BlackListHitCount', '黑名单命中次数'],
199 +         'BlackListSize': ['BlackListSize', '黑名单长度'],
200 +         'WhiteListHitCount': ['WhiteListHitCount', '白名单命中次数'],
201 +         'WhiteListSize': ['WhiteListSize', '白名单长度'],
202 +         'SyntaxErrorCount': ['SyntaxErrorCount', '语法错误次数'],
203 +         'TableName': ['TableName', '表名'],
204 +
205 +         'SelectCount': ['SelectCount', 'Select数'],
206 +         'InsertCount': ['InsertCount', 'Insert数'],
207 +         'UpdateCount': ['UpdateCount', 'Update数'],
208 +         'DeleteCount': ['DeleteCount', 'Delete数'],
209 +         'TruncateCount': ['TruncateCount', 'Truncate数'],
210 +         'CreateCount': ['CreateCount', 'Create数'],
211 +         'AlterCount': ['AlterCount', 'Alter数'],
212 +         'DropCount': ['DropCount', 'Drop数'],
213 +         'ReplaceCount': ['ReplaceCount', 'Replace数'],
214 +         'DeleteDataCount': ['DeleteDataCount', '删除数据行数'],
215 +         'UpdateDataCount': ['UpdateDataCount', '更新数据行数'],
216 +         'FetchRowCount': ['FetchRowCount', '读取行数'],
217 +         /**session-detail*/
218 +         'SESSIONID': ['SESSIONID', 'SESSIONID'],
219 +         'UserAgent': ['UserAgent', 'UserAgent'],
220 +         'Principal': ['Principal', 'Principal'],
221 +         'CreateTime': ['CreateTime', '创建时间'],
222 +         'LastAccessTime': ['LastAccessTime', '最后访问时间'],
223 +         'RemoteAddress': ['RemoteAddress', '访问ip地址'],
224 +         'RequestCount': ['RequestCount', '请求次数'],
225 +         'RequestTimeMillisTotal': ['RequestTimeMillisTotal', '总共请求时
    间'],
226 +         'RequestInterval': ['RequestInterval', '请求间隔'],
227 +         /**weburi-detail*/
228 +         'RequestTimeMillis': ['RequestTimeMillis', '请求时间'],
229 +         'JdbcExecutePeak': ['JdbcExecutePeak', 'jdbc执行峰值'],
230 +         'JdbcFetchRowPeak': ['JdbcFetchRowPeak', 'jdbc查询取回行数峰值'],
231 +         'JdbcUpdatePeak': ['JdbcUpdatePeak', 'jdbc更新峰值']
232 +
233 +     };
234 +
235 +     var COOKIE_LANG_NAME = "cookie_lang";
236 +
237 +     function log(str) {
238 +         if (typeof (console) != 'undefined' && typeof(console.log) !=
    'undefined') {
239 +             console.log(str);
240 +         } else {
241 +             $('body').append('<input type="hidden" value="' + str +
    " />");
242 +         }
243 +     }
244 +
245 +     function setCookie(name,value,expires,path,domain,secure)
246 +     {

```



```

247 +         var expDays = expires*24*60*60*1000;
248 +         var expDate = new Date();
249 +         expDate.setTime(expDate.getTime()+expDays);
250 +         var expString = ((expires==null) ? "":
(";expires="+expDate.toGMTString()));
251 +         var pathString = ((path==null) ? "": (";path="+path));
252 +         var domainString = ((domain==null) ? "": (";domain="+domain));
253 +         var secureString = ((secure==true) ? ";secure: "; "");
254 +         document.cookie = name + "=" + escape(value) + expString +
pathString + domainString + secureString;
255 +     }
256 +
257 +     function getCookie(name)
258 +     {
259 +         var result = null;
260 +         var myCookie = document.cookie + ";";
261 +         var searchName = name + "=";
262 +         var startOfCookie = myCookie.indexOf(searchName);
263 +         var endOfCookie;
264 +         if (startOfCookie != -1)
265 +         {
266 +             startOfCookie += searchName.length;
267 +             endOfCookie = myCookie.indexOf(";",startOfCookie);
268 +             result =
unescape(myCookie.substring(startOfCookie,endOfCookie));
269 +         }
270 +         return result;
271 +     }
272 +
273 +     function setText($obj) {
274 +         var key = $obj.attr('langKey');
275 +         if (typeof(lang[key]) != 'undefined') {
276 +             var text = lang[key][druid.lang.langNow];
277 +             $obj.text(lang[key][druid.lang.langNow]);
278 +         } else {
279 +             log('key [' + key + '] not found');
280 +         }
281 +     }
282 +     function setTitle($obj) {
283 +         var key = $obj.attr('langKey');
284 +         if (typeof(lang[key]) != 'undefined') {
285 +             var title = lang[key][druid.lang.langNow];
286 +             $obj.attr('title', title);
287 +         } else {
288 +             log('key [' + key + '] not found');
289 +         }
290 +     }
291 +
292 +     return {
293 +         langNow : LANG_CN,
294 +         EVENT_LOAD_FINISHED : 'loadFinished',
295 +         init : function(langNow) {
296 +             if (typeof(langNow) != 'undefined') {
297 +                 this.setLangType(langNow);
298 +             } else {

```



```

299 +                 var langInCookie = getCookie(COOKIE_LANG_NAME);
300 +                 if (langInCookie == LANG_CN || langInCookie ==
    LANG_EN) {
301 +                     this.setLangType(langInCookie);
302 +                 }
303 +             }
304 +             $(document).on(this.EVENT_LOAD_FINISHED, '.lang',
    function() {
305 +                 log('load lang');
306 +                 setText($(this));
307 +             });
308 +             $(document).on(this.EVENT_LOAD_FINISHED, '.langTitle',
    function() {
309 +                 log('load title');
310 +                 setTitle($(this));
311 +             });
312 +             this.trigger();
313 +
314 +             $(document).on('click', '.langSelector', function() {
315 +                 var langSelected = $(this).attr('langNow');
316 +                 druid.lang.setLangType(langSelected);
317 +                 druid.lang.trigger();
318 +                 return false;
319 +             });
320 +         },
321 +         setLangType : function (langNow) {
322 +             this.langNow = langNow;
323 +             setCookie(COOKIE_LANG_NAME, langNow, 30, '/');
324 +         },
325 +         getLangType : function () {
326 +             return this.langNow;
327 +         },
328 +         show : function($parent) {
329 +             var $obj;
330 +             var $objTitle;
331 +             if ($parent) {
332 +                 $obj = $parent.find('.lang');
333 +                 $objTitle = $parent.find('.langTitle');
334 +             } else {
335 +                 $obj = $('<div>.lang');
336 +                 $objTitle = $('<div>.langTitle');
337 +             }
338 +             $obj.each(function() {
339 +                 setText($(this));
340 +             });
341 +             $objTitle.each(function() {
342 +                 setTitle($(this));
343 +             });
344 +         },
345 +         trigger : function() {
346 +             log('to load lang now');
347 +             +             $('<div>.lang').trigger(this.EVENT_LOAD_FINISHED); // 触发语言
    显示事件
348 +             +             $('<div>.langTitle').trigger(this.EVENT_LOAD_FINISHED); // 触发
    语言显示事件

```

```
349 +      }
350 +    }
351 +})();
```

...

...

src/main/resources/support/http/resources/sql.html

```
...    ... @@ -23,50 +23,50 @@
23    23                                     <tr>
24    24    <th>N</th>
25    25                                     <th><a
id="th-SQL" >SQL</a></th>
-                                     <th
26    width="50" title="Execute Count"><a id="th-ExecuteCount" class="lang"
langKey="ExecuteCount">ExecCount</a></th>
-                                     <th
27    width="50" title="Execute Time Millis Total"><a id="th-TotalTime" class="lang"
langKey="ExecuteTimeMillis">ExecTime</a></th>
-                                     <th
28    width="50" title="Execute Time Millis Max"><a id="th-MaxTimespan" class="lang"
langKey="MaxTimespan">ExecMax</a></th>
-                                     <th
29    width="50" title="Execute In Transaction Count"><a id="th-InTransactionCount"
class="lang" langKey="InTransactionCount">Txn</a></th>
-                                     <th
30    width="50" title="Error Count"><a id="th-ErrorCount" class="lang"
langKey="ErrorCount">Error</a></th>
-                                     <th
31    width="50" title="Update Count"><a id="th-UpdateCount" class="lang"
langKey="JdbcUpdateCount">Update</a></th>
-                                     <th
32    width="50" title="Fetch Row Count"><a id="th-FetchRowCount" class="lang"
langKey="JdbcFetchRowCount">FetchRow</a></th>
-                                     <th
33    width="50" title="Running Count"><a id="th-RunningCount" class="lang"
langKey="RunningCount">Running</a></th>
-                                     <th
34    width="50" title="Concurrent Max"><a id="th-ConcurrentMax" class="lang"
langKey="ConcurrentMax">Concurrent</a></th>
+                                     <th
26    width="50"><a id="th-ExecuteCount" class="lang"
langKey="ExecuteCount">ExecCount</a></th>
+                                     <th
27    width="50"><a id="th-TotalTime" class="lang"
langKey="ExecuteTimeMillis">ExecTime</a></th>
+                                     <th
28    width="50" class="langTitle" langKey="MaxTimespanDesc" title="Execute Time
Millis Max"><a id="th-MaxTimespan" class="lang"
langKey="MaxTimespan">ExecMax</a></th>
+                                     <th
29    width="50" class="langTitle" langKey="InTransactionCountDesc" title="Execute In
Transaction Count"><a id="th-InTransactionCount" class="lang"
langKey="InTransactionCount">Txn</a></th>
```

```

+                                                                 <th
30 width="50" ><a id="th-ErrorCount" class="lang" langKey="ErrorCount">Error</a>
  </th>
+                                                                 <th
31 width="50"><a id="th-UpdateCount" class="lang"
  langKey="JdbcUpdateCount">Update</a></th>
+                                                                 <th
32 width="50"><a id="th-FetchRowCount" class="lang"
  langKey="JdbcFetchRowCount">FetchRow</a></th>
+                                                                 <th
33 width="50"><a id="th-RunningCount" class="lang"
  langKey="RunningCount">Running</a></th>
+                                                                 <th
34 width="50"><a id="th-ConcurrentMax" class="lang"
  langKey="ConcurrentMax">Concurrent</a></th>
+                                                                 <th
35 35 align="left" width="100"><span class="lang" langKey="ExecHisto">ExecHisto</span>
  <br />[
36 -
  <a id="th-Histogram[0]" langTitle="count1ms" title="count of '0-1 ms'" >-</a>
37 -
  <a id="th-Histogram[1]" langTitle="count10ms" title="count of '1-10 ms'" >-</a>
  -
38 <a id="th-Histogram[2]" langTitle="count100ms" title="count of '10-100 ms'" >-
  </a>
  -
39 <a id="th-Histogram[3]" langTitle="count1s" title="count of '100ms-1 s'" >-</a>
  -
40 <a id="th-Histogram[4]" langTitle="count10s" title="count of '1-10 s'" >-</a>
  -
41 <a id="th-Histogram[5]" langTitle="count100s" title="count of '10-100 s'" >-</a>
  -
42 <a id="th-Histogram[6]" langTitle="count1000s" title="count of '100-1000 s'" >-
  </a>
  -
43 <a id="th-Histogram[7]" langTitle="countBg1000s" title="count of '> 1000 s'" >-
  </a> ]
+
36 <a id="th-Histogram[0]" class="langTitle" langKey="count1ms" title="count of '0-
  1 ms'" >-</a>
+
37 <a id="th-Histogram[1]" class="langTitle" langKey="count10ms" title="count of
  '1-10 ms'" >-</a>
+
38 <a id="th-Histogram[2]" class="langTitle" langKey="count100ms" title="count of
  '10-100 ms'" >-</a>
+
39 <a id="th-Histogram[3]" class="langTitle" langKey="count1s" title="count of
  '100ms-1 s'" >-</a>
+
40 <a id="th-Histogram[4]" class="langTitle" langKey="count10s" title="count of '1-
  10 s'" >-</a>
+
41 <a id="th-Histogram[5]" class="langTitle" langKey="count100s" title="count of
  '10-100 s'" >-</a>

```

```
+
42 <a id="th-Histogram[6]" class="langTitle" langKey="count1000s" title="count of
    '100-1000 s'" >-</a>
+
43 <a id="th-Histogram[7]" class="langTitle" langKey="countBg1000s" title="count of
    '> 1000 s'" >-</a> ]
44 44
    </th>
    <th
45 45 align="left" width="100"><span class="lang"
    langKey="ExecRsHisto">ExecRsHisto</span> <br />[
-
46 <a id="th-ExecuteAndResultHoldTimeHistogram[0]" langTitle="count1ms"
    title="count of '0-1 ms'" >-</a>
-
47 <a id="th-ExecuteAndResultHoldTimeHistogram[1]" langTitle="count10ms"
    title="count of '1-10 ms'" >-</a>
-
48 <a id="th-ExecuteAndResultHoldTimeHistogram[2]" langTitle="count100ms"
    title="count of '10-100 ms'" >-</a>
-
49 <a id="th-ExecuteAndResultHoldTimeHistogram[3]" langTitle="count1s" title="count
    of '100ms-1s'" >-</a>
-
50 <a id="th-ExecuteAndResultHoldTimeHistogram[4]" langTitle="count10s"
    title="count of '1-10 s'" >-</a>
-
51 <a id="th-ExecuteAndResultHoldTimeHistogram[5]" langTitle="count100s"
    title="count of '10-100 s'" >-</a>
-
52 <a id="th-ExecuteAndResultHoldTimeHistogram[6]" langTitle="count1000s"
    title="count of '100-1000 s'" >-</a>
-
53 <a id="th-ExecuteAndResultHoldTimeHistogram[7]" langTitle="countBg1000s"
    title="count of '> 1000 s'" >-</a> ]
+
46 <a id="th-ExecuteAndResultHoldTimeHistogram[0]" class="langTitle"
    langKey="count1ms" title="count of '0-1 ms'" >-</a>
+
47 <a id="th-ExecuteAndResultHoldTimeHistogram[1]" class="langTitle"
    langKey="count10ms" title="count of '1-10 ms'" >-</a>
+
48 <a id="th-ExecuteAndResultHoldTimeHistogram[2]" class="langTitle"
    langKey="count100ms" title="count of '10-100 ms'" >-</a>
+
49 <a id="th-ExecuteAndResultHoldTimeHistogram[3]" class="langTitle"
    langKey="count1s" title="count of '100ms-1s'" >-</a>
+
50 <a id="th-ExecuteAndResultHoldTimeHistogram[4]" class="langTitle"
    langKey="count10s" title="count of '1-10 s'" >-</a>
+
51 <a id="th-ExecuteAndResultHoldTimeHistogram[5]" class="langTitle"
    langKey="count100s" title="count of '10-100 s'" >-</a>
+
52 <a id="th-ExecuteAndResultHoldTimeHistogram[6]" class="langTitle"
    langKey="count1000s" title="count of '100-1000 s'" >-</a>
```

```

+
53 <a id="th-ExecuteAndResultHoldTimeHistogram[7]" class="langTitle"
    langKey="countBg1000s" title="count of '> 1000 s'" >-</a> ]
54 54
                                     </th>
                                     <th
55 55 align="left" width="100"><span class="lang"
    langKey="FetchRowHisto">FetchRowHisto</span> <br />[
-
56 <a id="th-FetchRowCountHistogram[0]" langTitle="fetch0" title="count of '0
    FetchRow'" >-</a>
-
57 <a id="th-FetchRowCountHistogram[1]" langTitle="fetch9" title="count of '1-9
    FetchRow'" >-</a>
-
58 <a id="th-FetchRowCountHistogram[2]" langTitle="fetch99" title="count of '10-99
    FetchRow'" >-</a>
-
59 <a id="th-FetchRowCountHistogram[3]" langTitle="fetch999" title="count of '100-
    999 FetchRow'" >-</a>
-
60 <a id="th-FetchRowCountHistogram[4]" langTitle="fetch9999" title="count of
    '1000-9999 FetchRow'" >-</a>
-
61 <a id="th-FetchRowCountHistogram[5]" langTitle="fetch99999" title="count of '>
    9999 FetchRow'" >-</a> ]
+
56 <a id="th-FetchRowCountHistogram[0]" class="langTitle" langKey="fetch0"
    title="count of '0 FetchRow'" >-</a>
+
57 <a id="th-FetchRowCountHistogram[1]" class="langTitle" langKey="fetch9"
    title="count of '1-9 FetchRow'" >-</a>
+
58 <a id="th-FetchRowCountHistogram[2]" class="langTitle" langKey="fetch99"
    title="count of '10-99 FetchRow'" >-</a>
+
59 <a id="th-FetchRowCountHistogram[3]" class="langTitle" langKey="fetch999"
    title="count of '100-999 FetchRow'" >-</a>
+
60 <a id="th-FetchRowCountHistogram[4]" class="langTitle" langKey="fetch9999"
    title="count of '1000-9999 FetchRow'" >-</a>
+
61 <a id="th-FetchRowCountHistogram[5]" class="langTitle" langKey="fetch99999"
    title="count of '> 9999 FetchRow'" >-</a> ]
62 62
                                     </th>
                                     <th
63 63 align="left" width="100"><span class="lang"
    langKey="UpdateHisto">UpdateHisto</span> <br />[
-
64 <a id="th-EffecteRowCountHistogram[0]" langTitle="update0" title="count of '0
    UpdateCount'" >-</a>
-
65 <a id="th-EffecteRowCountHistogram[1]" langTitle="update9" title="count of '1-9
    UpdateCount'" >-</a>

```

```
66      <a id="th-EffecteRowCountHistogram[2]" langTitle="update99" title="count of
      '10-99 UpdateCount'" >-</a>
67      <a id="th-EffecteRowCountHistogram[3]" langTitle="update999" title="count of
      '100-999 UpdateCount'" >-</a>
68      <a id="th-EffecteRowCountHistogram[4]" langTitle="update9999" title="count of
      '1000-9999 UpdateCount'" >-</a>
69      <a id="th-EffecteRowCountHistogram[5]" langTitle="update99999" title="count of
      '> 9999 UpdateCount'" >-</a> ]
      +
64      <a id="th-EffecteRowCountHistogram[0]" class="langTitle" langKey="update0"
      title="count of '0 UpdateCount'" >-</a>
      +
65      <a id="th-EffecteRowCountHistogram[1]" class="langTitle" langKey="update9"
      title="count of '1-9 UpdateCount'" >-</a>
      +
66      <a id="th-EffecteRowCountHistogram[2]" class="langTitle" langKey="update99"
      title="count of '10-99 UpdateCount'" >-</a>
      +
67      <a id="th-EffecteRowCountHistogram[3]" class="langTitle" langKey="update999"
      title="count of '100-999 UpdateCount'" >-</a>
      +
68      <a id="th-EffecteRowCountHistogram[4]" class="langTitle" langKey="update9999"
      title="count of '1000-9999 UpdateCount'" >-</a>
      +
69      <a id="th-EffecteRowCountHistogram[5]" class="langTitle" langKey="update99999"
      title="count of '> 9999 UpdateCount'" >-</a> ]
70      70                                     </th>
71      71                                     </tr>
72      72                                     </thead>
...    ...
```