



This repository Search

Pull requests Issues Gist



basho / **taste-of-riak**

Watch 121 Star 13 Fork 10

Code Issues 0 Pull requests 0 Projects 0 Wiki Pulse Graphs

Branch: master **taste-of-riak / nodejs / Ch02-Schemas-and-Indexes / app.js**

Find file Copy path

lukebakken Ch01 for NodeJS

94826cc on Apr 3, 2015

1 contributor

275 lines (236 sloc) 7.65 KB

Raw Blame History

```
1  'use strict';
2
3  var async = require('async');
4  var logger = require('winston');
5  var Riak = require('basho-riak-client');
6
7  // Un-comment to enable debug logging
8  // logger.remove(logger.transports.Console);
9  // logger.add(logger.transports.Console, {
10 //   level : 'debug',
11 //   colorize: true,
12 //   timestamp: true
13 // });
14
15 function checkErr(err) {
16   if (err) {
17     logger.error(err);
18     process.exit(1);
19   }
20 }
21
22 logger.info("Creating Data");
23
24 var customer = {
25   id : '1',
26   name : "John Smith",
27   address : "123 Main Street",
28   city : "Columbus",
29   state : "Ohio",
30   zip : "43210",
31   phone : "+1-614-555-5555",
32   createdAt : "2013-10-01 14:30:26",
33 };
34
35 var orders = createOrders();
36 var orderSummary = createOrderSummary(orders);
37
38 logger.info("Starting Client");
39 var node = new Riak.Node({remoteAddress: 'riak-test', remotePort: 10017});
40 var cluster = new Riak.Cluster({nodes: [node]});
41 var client = new Riak.Client(cluster);
42
43 var customersBucket = "Customers";
44 var ordersBucket = "Orders";
45 var orderSummariesBucket = "OrderSummaries";
46
47 store_objects();
48
49 function store_objects() {
50   logger.info("Storing Data");
```

```

51
52     var storeFuncs = [
53         function (async_cb) {
54             client.storeValue({ bucket: customersBucket, key: customer.id, value: customer }, async_cb);
55         },
56         function (async_cb) {
57             client.storeValue({
58                 bucket: orderSummariesBucket,
59                 key: orderSummary.customerId,
60                 value: orderSummary
61             }, async_cb);
62         }
63     ];
64
65     orders.forEach(function (order) {
66         storeFuncs.push(
67             function (async_cb) {
68                 client.storeValue({ bucket: ordersBucket, key: order.id, value: order }, async_cb);
69             });
70     });
71
72     async.parallel(storeFuncs, function (err, rslts) {
73         checkErr(err);
74         fetch_objects();
75     });
76 }
77
78 function fetch_objects() {
79     logger.info("Fetching related data by shared key");
80
81     var key = '1';
82     client.fetchValue({ bucket: customersBucket, key: key }, function (err, rslt) {
83         checkErr(err);
84         var fetchedCustomerObj = rslt.values.shift();
85         var fetchedCustomer = fetchedCustomerObj.value.toString("utf8");
86         client.fetchValue({ bucket: orderSummariesBucket, key: key}, function (err, rslt) {
87             checkErr(err);
88             var fetchedOrderSummaryObj = rslt.values.shift();
89             var fetchedOrderSummary = fetchedOrderSummaryObj.value.toString("utf8");
90             logger.info("Customer 1: %s", fetchedCustomer);
91             logger.info("OrderSummary 1: %s", fetchedOrderSummary);
92
93             add_index_data();
94         });
95     });
96 }
97
98 function add_index_data() {
99     logger.info("Adding Index Data");
100
101     var funcs = [
102         function (async_cb) {
103             client.fetchValue({ bucket: ordersBucket, key : '1' }, function (err, rslt) {
104                 checkErr(err);
105                 var fetchedOrderObj = rslt.values.shift();
106                 fetchedOrderObj.addToIndex('SalespersonId_int', 9000);
107                 fetchedOrderObj.addToIndex('OrderDate_bin', '2013-10-01');
108                 client.storeValue({ value: fetchedOrderObj }, function (err, rslt) {
109                     async_cb(err, rslt);
110                 });
111             });
112         },
113         function (async_cb) {
114             client.fetchValue({ bucket: ordersBucket, key : '2' }, function (err, rslt) {
115                 checkErr(err);
116                 var fetchedOrderObj = rslt.values.shift();
117                 fetchedOrderObj.addToIndex('SalespersonId_int', 9001);

```

```

118         fetchedOrderObj.addToIndex('OrderDate_bin', '2013-10-15');
119         client.storeValue({ value: fetchedOrderObj }, function (err, rslt) {
120             async_cb(err, rslt);
121         });
122     });
123 },
124     function (async_cb) {
125         client.fetchValue({ bucket: ordersBucket, key : '3' }, function (err, rslt) {
126             checkErr(err);
127             var fetchedOrderObj = rslt.values.shift();
128             fetchedOrderObj.addToIndex('SalespersonId_int', 9000);
129             fetchedOrderObj.addToIndex('OrderDate_bin', '2013-11-03');
130             client.storeValue({ value: fetchedOrderObj }, function (err, rslt) {
131                 async_cb(err, rslt);
132             });
133         });
134     }
135 ];
136
137     async.parallel(funcs, function (err, rsalts) {
138         checkErr(err);
139         index_queries();
140     });
141 }
142
143     function index_queries() {
144         logger.info("Index Queries");
145
146         var janesOrders = [];
147         client.secondaryIndexQuery({ bucket: ordersBucket, indexName: 'SalespersonId_int', indexKey: 9000 }, function (err, rslt) {
148             checkErr(err);
149
150             if (rslt.values.length > 0) {
151                 Array.prototype.push.apply(janesOrders,
152                     rslt.values.map(function (value) {
153                         return value.objectKey;
154                     })
155                 );
156             }
157
158             if (rslt.done) {
159                 logger.info("Jane's Orders: %s", janesOrders.join(', '));
160
161                 var octoberOrders = [];
162                 client.secondaryIndexQuery({ bucket: ordersBucket, indexName: 'OrderDate_bin',
163                     rangeStart: '2013-10-01', rangeEnd: '2013-10-31' }, function (err, rslt) {
164                     checkErr(err);
165
166                     if (rslt.values.length > 0) {
167                         Array.prototype.push.apply(octoberOrders,
168                             rslt.values.map(function (value) {
169                                 return value.objectKey;
170                             })
171                         );
172                     }
173
174                     if (rslt.done) {
175                         logger.info("October's Orders: %s", octoberOrders.join(', '));
176
177                         client_shutdown();
178                     }
179                 });
180             }
181         });
182     }
183
184     function client_shutdown() {

```

```

185     client.shutdown(function (state) {
186         if (state === Riak.Cluster.State.SHUTDOWN) {
187             logger.info("cluster stopped");
188             process.exit();
189         }
190     });
191 }
192
193 function createOrders() {
194     var orders = [];
195
196     var order1 = {
197         id : '1',
198         customerId : '1',
199         salespersonId : 9000,
200         items : [
201             {
202                 id: "TCV37GIT4NJ",
203                 title: "USB 3.0 Coffee Warmer",
204                 price: 15.99
205             },
206             {
207                 id: "PEG10BBF2PP",
208                 title: "eTablet Pro, 24GB; Grey",
209                 price: 399.99
210             },
211         ],
212         total : 415.98,
213         orderDate : "2013-10-01 14:42:26"
214     };
215     orders.push(order1);
216
217     var order2 = {
218         id : '2',
219         customerId : '1',
220         salespersonId : 9001,
221         items : [
222             {
223                 id : "OAX19XWN0QP",
224                 title : "GoSlo Digital Camera",
225                 price : 359.99
226             },
227         ],
228         total : 359.99,
229         orderDate : "2013-10-15 16:43:16"
230     };
231     orders.push(order2);
232
233     var order3 = {
234         id : '3',
235         customerId : '1',
236         salespersonId : 9000,
237         items : [
238             {
239                 id : "WYK12EPU5EZ",
240                 title : "Call of Battle : Goats - Gamesphere 4",
241                 price : 69.99
242             },
243             {
244                 id : "TJB84HAA80A",
245                 title : "Bricko Building Blocks",
246                 price : 4.99
247             },
248         ],
249         total : 74.98,
250         orderDate : "2013-11-03 17:45:28"
251     };
252     orders.push(order3);

```

```
253
254     return orders;
255 }
256
257 function createOrderSummary(orders) {
258
259     var orderSummary = {
260         customerId : '1',
261         summaries : []
262     };
263
264     orders.forEach(function (order) {
265         orderSummary.summaries.push({
266             orderId : order.id,
267             total : order.total,
268             orderDate : order.orderDate
269         });
270     });
271
272     return orderSummary;
273 }
274
```

