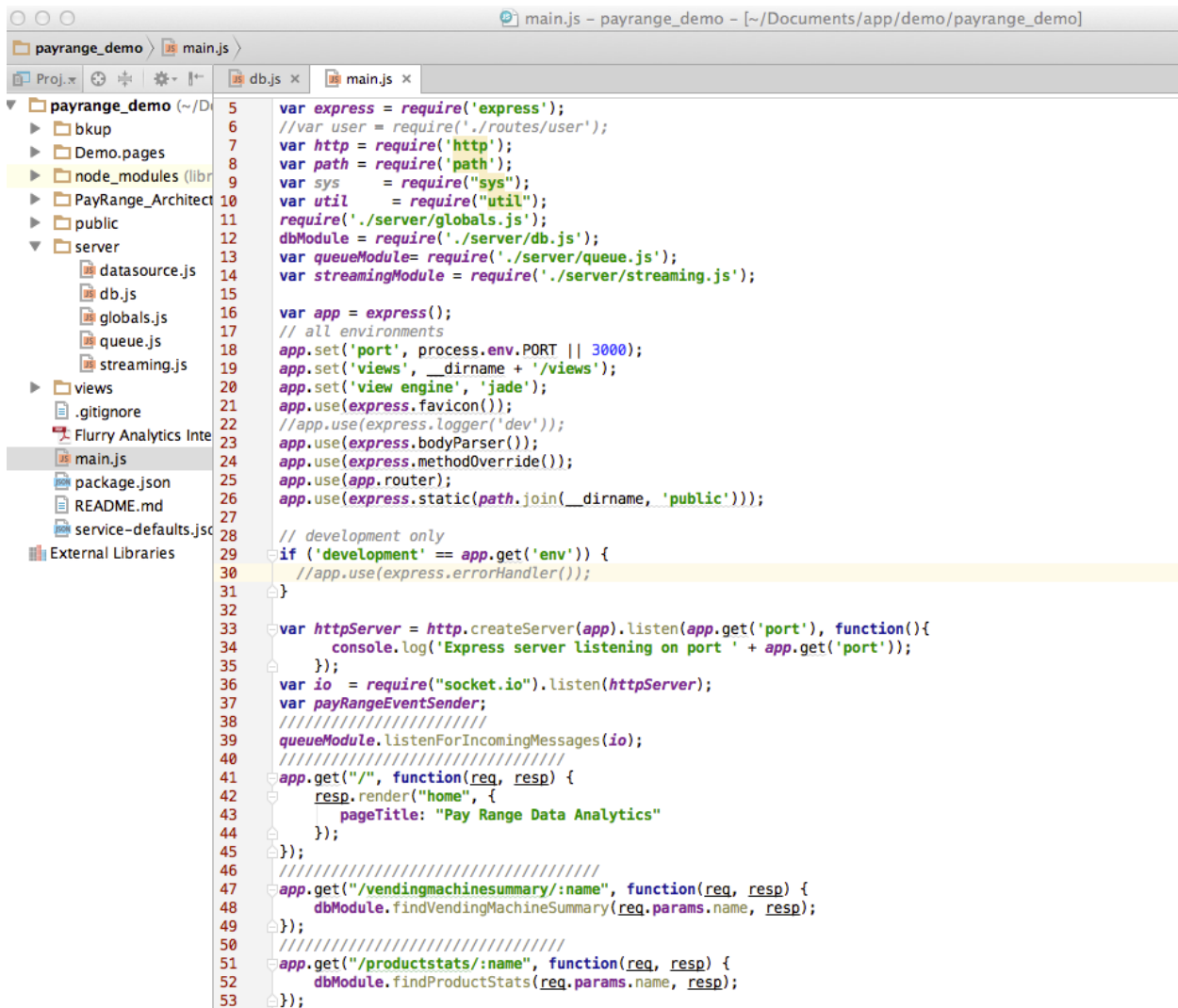


## Analytics App Demo

### Development Environment



```
main.js - payrange_demo - [~/Documents/app/demo/payrange_demo]

payrange_demo > main.js >
Proj. x db.js x main.js x

payrange_demo (~/.D
  bkup
  Demo.pages
  node_modules (lib
  PayRange_Architect
  public
  server
    datasource.js
    db.js
    globals.js
    queue.js
    streaming.js
  views
  .gitignore
  Flurry Analytics Inte
  main.js
  package.json
  README.md
  service-defaults.js
  External Libraries

5  var express = require('express');
6  //var user = require('./routes/user');
7  var http = require('http');
8  var path = require('path');
9  var sys = require('sys');
10 var util = require('util');
11 require('./server/globals.js');
12 dbModule = require('./server/db.js');
13 var queueModule = require('./server/queue.js');
14 var streamingModule = require('./server/streaming.js');
15
16 var app = express();
17 // all environments
18 app.set('port', process.env.PORT || 3000);
19 app.set('views', __dirname + '/views');
20 app.set('view engine', 'jade');
21 app.use(express.favicon());
22 //app.use(express.logger('dev'));
23 app.use(express.bodyParser());
24 app.use(express.methodOverride());
25 app.use(app.router);
26 app.use(express.static(path.join(__dirname, 'public')));
27
28 // development only
29 if ('development' == app.get('env')) {
30   //app.use(express.errorHandler());
31 }
32
33 var httpServer = http.createServer(app).listen(app.get('port'), function(){
34   console.log('Express server listening on port ' + app.get('port'));
35 });
36 var io = require("socket.io").listen(httpServer);
37 var payRangeEventSender;
38 ///////////////////////////////////////////////////
39 queueModule.listenForIncomingMessages(io);
40 ///////////////////////////////////////////////////
41 app.get("/", function(req, resp) {
42   resp.render("home", {
43     pageTitle: "Pay Range Data Analytics"
44   });
45 });
46 ///////////////////////////////////////////////////
47 app.get("/vendingmachinesummary/:name", function(req, resp) {
48   dbModule.findVendingMachineSummary(req.params.name, resp);
49 });
50 ///////////////////////////////////////////////////
51 app.get("/productstats/:name", function(req, resp) {
52   dbModule.findProductStats(req.params.name, resp);
53 });
```

**Runtime :**

**Start services :**

**>> mongod --dbpath=/data/mongo**

**>> redis-server**

**>> node main.js**

**Next we see in the console log :**

**=> data streams are ingested**

**=> features are extracted**

**=> messages are sent to queue**

**=> data sent to UI**

**=> raw data persisted in primary store (payrangedetails)**

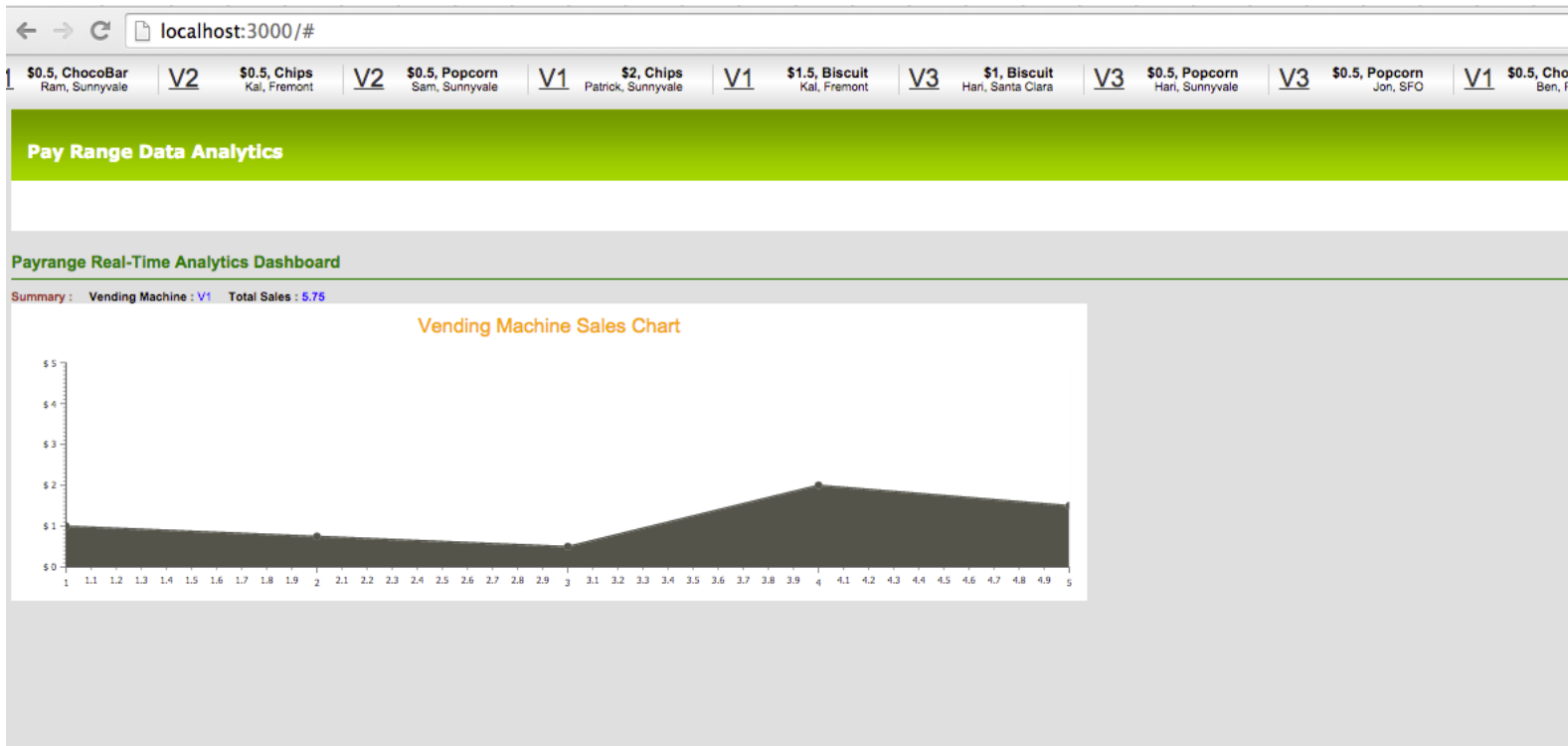
**=> all aggregate collections are updated with latest stats in parallel**

```
payrange_demo (~/.Docum 1)
Run main.js (1)
/usr/local/bin/node main.js
info: socket.io started
Express server listening on port 3000
Warning: missing space before text for line 10 of jade file "/Users/kaniska_mac/Documents/app/demo/payrange_demo/views/home.jade"
Warning: missing space before text for line 11 of jade file "/Users/kaniska_mac/Documents/app/demo/payrange_demo/views/home.jade"
debug: served static content /socket.io.js
debug: client authorized
info: handshake authorized fqxDaiVq7wzzh9IPCMpg
debug: setting request GET /socket.io/1/websocket/fqxDaiVq7wzzh9IPCMpg
debug: set heartbeat interval for client fqxDaiVq7wzzh9IPCMpg
debug: client authorized for
debug: websocket writing 1::
DEBUG: sending payrange event: {"timestamp":"2016-01-01T08:00:00.000Z","vendingmachine":"V1","user":"Hari","city":"Sunnyvale","product":"Popcorn","expense":1,"totalTimeSpent":1410,"totalPayments":14.5,"user":"Hari","__v":0}
DEBUG: Got an event
DEBUG: connection made...[object Object]
DEBUG: <===== START =====>
DEBUG: Incoming data ... {"timestamp":"2016-01-01T08:00:00.000Z","vendingmachine":"V1","user":"Hari","city":"Sunnyvale","product":"Popcorn","expense":1,"totalTimeSpent":1410,"totalPayments":14.5,"user":"Hari","__v":0}
DEBUG: Parsed data ... [object Object]
DEBUG: About to send data to Browser..
DEBUG: Sent data to Browser:
DEBUG: Records Count:
debug: websocket writing 3::{"timestamp":"2016-01-01T08:00:00.000Z","vendingmachine":"V1","user":"Hari","city":"Sunnyvale","product":"Popcorn","expense":1,"totalTimeSpent":1410,"totalPayments":14.5,"user":"Hari","__v":0}
DEBUG: Step 1 : Got a Stock Symbol..{"timestamp":"2016-01-01T08:00:00.000Z","vendingmachine":"V1","user":"Hari","city":"Sunnyvale","product":"Popcorn","expense":1,"totalTimeSpent":1410,"totalPayments":14.5,"user":"Hari","__v":0}
DEBUG: undefined
DEBUG: Found Summary { _id: 56c01dd2913b89d639078141,
  totalTimeSpent: 1410,
  totalPayments: 14.5,
  user: 'Hari',
  __v: 0 }
DEBUG: Step 4 : Saved the summary..{"_id":"56c01dd2913b89d639078141","totalTimeSpent":1510,"totalPayments":15.5,"user":"Hari","__v":0}
DEBUG: <===== END =====>
DEBUG: Found Summary undefined
DEBUG: Event : Create new Summary document
DEBUG: <===== END =====>
DEBUG: Found Summary { _id: 56c01dd2913b89d639078140,
  totalSales: 12.75,
  product: 'Popcorn',
  __v: 0 }
DEBUG: Step 4 : Saved the summary..{"_id":"56c01dd2913b89d639078140","totalSales":13.75,"product":"Popcorn","__v":0}
DEBUG: <===== END =====>
DEBUG: sending payrange event: {"timestamp":"2016-01-02T08:00:00.000Z","vendingmachine":"V2","user":"Bob","city":"Sunnyvale","product":"ChocoBar","expense":1,"totalTimeSpent":1410,"totalPayments":14.5,"user":"Bob","__v":0}
DEBUG: Got an event
DEBUG: <===== START =====>
```

Now click a Message

=> Total Sales of Vending Machine - displayed

=> also shows how it performs over a time period



We can find 'highest paying users', 'best selling vending machines', 'best selling products'

```
> db.userssummary.find({}, {user:1, totalTimeSpent:2, _id:0}).sort({"totalTimeSpent":-1})
{ "totalTimeSpent" : 1410, "user" : "Hari" }
{ "totalTimeSpent" : 390, "user" : "Rabi" }
{ "totalTimeSpent" : 300, "user" : "Ben" }
> db.productssummary.find({}, {product:1, totalSales:2, _id:0}).sort({"totalSales":-1})
{ "totalSales" : 12.75, "product" : "Popcorn" }
{ "totalSales" : 7.75, "product" : "ChocoBar" }
{ "totalSales" : 6.25, "product" : "Chips" }
{ "totalSales" : 6, "product" : "Biscuit" }
{ "totalSales" : 4, "product" : "Dry Fruits" }
{ "totalSales" : 1.5, "product" : "Kitkat" }
>
> db.userssummary.find({}, {user:1, totalPayments:2, _id:0}).sort({"totalPayments":-1})
{ "totalPayments" : 14.5, "user" : "Hari" }
{ "totalPayments" : 3.5, "user" : "Sam" }
{ "totalPayments" : 3, "user" : "Patrick" }
{ "totalPayments" : 2.75, "user" : "Rabi" }
{ "totalPayments" : 2.5, "user" : "Kal" }
{ "totalPayments" : 2, "user" : "Teri" }
{ "totalPayments" : 1.5, "user" : "Ram" }
{ "totalPayments" : 1.5, "user" : "Justin" }
{ "totalPayments" : 1.25, "user" : "Ben" }
{ "totalPayments" : 1, "user" : "Bob" }
{ "totalPayments" : 1, "user" : "Tom" }
>
> db.vendingsummary.find({}, {vendingmachine:1, totalSales:2, _id:0}).sort({"totalSales":-1})
{ "totalSales" : 18.25, "vendingmachine" : "V1" }
{ "totalSales" : 4.5, "vendingmachine" : "V7" }
{ "totalSales" : 4.5, "vendingmachine" : "V3" }
{ "totalSales" : 4, "vendingmachine" : "V2" }
{ "totalSales" : 2.5, "vendingmachine" : "V6" }
{ "totalSales" : 2.5, "vendingmachine" : "V5" }
{ "totalSales" : 1.5, "vendingmachine" : "V9" }
{ "totalSales" : 1.5, "vendingmachine" : "V10" }
>
```