**Exercise03\_01\_01 – Step 1**

terminal

npm install

------------------------------------

terminal

npm start

**Exercise03\_01\_01 – Step 2**

server.js

server.register([require('inert'), require('vision')], err => {

if (err) throw err;

------------------------------------

server.js

server.start(err => {

if (err) throw err;

------------------------------------

fluxGen.js

(seed, times, variability, positivity) => {

------------------------------------

fluxGen.js

module.exports = (seed, times, variability, positivity) => {

var output = [],

current = seed,

change;

for (var i = 0; i < times; i++) {

change = (Math.random() \* variability).toFixed(0);

if ((Math.random() \* positivity) <= (positivity / 2)) {

change = -change;

} else {

change = +change;

}

current += change;

output.push(current);

}

return output;

};

------------------------------------

pizza.js

var fluxGen = require(’../lib/fluxGen’);

**Exercise03\_01\_01 – Step 3**

pizza.js

~~var self = this;~~

------------------------------------

pizza.js

var addQuote = quote => this.quotes.push(quote);

------------------------------------

pizza.js

var getQuote = quoteIndex => this.quotes[quoteIndex];

------------------------------------

pizza.js

this.quotes = quotes || [this.startingQuote];

// private methods

var addQuote = quote => this.quotes.push(quote);

var getQuote = quoteIndex => this.quotes[quoteIndex];

**Exercise03\_01\_01 – Step 4**

pizza.js

this.variability = variability || getRand();

this.positivity = positivity || getRand();

------------------------------------

pizza.js

function Pizza (startingDate, quotes, ticker, name,   
 startingQuote, variability=getRand(), positivity=getRand()) {

------------------------------------

pizza.js

this.variability = variability;

this.positivity = positivity;

**Exercise03\_01\_01 – Step 5**

pizza.js

this.variability = variability || getRand();

this.positivity = positivity || getRand();

------------------------------------

pizza.js

function Pizza (startingDate, quotes, ...pizzaProps) {

------------------------------------

pizza.js

this.ticker = pizzaProps[0];

this.name = pizzaProps[1];

this.startingQuote = pizzaProps[2];

this.variability = pizzaProps[3] || getRand();

this.positivity = pizzaProps[4] || getRand();

**Exercise03\_01\_01 – Step 6**

dataStore.js

realPizzas[pizza[0]] = new Pizza(startingDate,   
 data.quotes[pizza[0]], pizza[0], pizza[1], pizza[2],   
 pizza[3], pizza[4]);

------------------------------------

dataStore.js

realPizzas[pizza[0]] = new Pizza(startingDate,   
 data.quotes[pizza[0]], ...pizza);

**Exercise03\_01\_01 – Step 7**

pizza.js

class Pizza {

constructor(startingDate, quotes, ...pizzaProps) {

this.startingDate = startingDate;

this.ticker = pizzaProps[0];

this.name = pizzaProps[1];

this.startingQuote = pizzaProps[2];

this.variability = pizzaProps[3] || getRand();

this.positivity = pizzaProps[4] || getRand();

this.quotes = quotes || [this.startingQuote];

}

------------------------------------

pizza.js

// public methods

getNext () {

var newQuote = fluxGen(this.getLast(), 1,   
 this.variability, this.positivity)[0];

addQuote(newQuote);

return newQuote;

}

------------------------------------

pizza.js

getLast () {

return getQuote(this.quotes.length - 1);

}

getDatedQuotes () {

var quotesMap = {},

curDate = startingDate;

this.quotes.forEach(function (quote) {

quotesMap[curDate] = quote;

curDate.setDate(curDate.getDate() + 1);

});

return quotesMap;

}

pizza.js

// private methods

\_addQuote (quote) {

return this.quotes.push(quote)

}

\_getQuote (quoteIndex) {

return this.quotes[quoteIndex];

}

------------------------------------

pizza.js

getDatedQuotes () {

var quotesMap = {},

curDate = this.startingDate;