**Exercise03\_04\_01 – Step 1**

dataStore.js

for(const pizza of pizzas) {

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

};

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

pizza.js

for(const quote of this.quotes) {

quotesMap[curDate] = quote;

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

}

**Exercise03\_04\_01 – Step 2**

mainContent.js

function\* mainContent() {

}

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

mainContent.js

function\* mainContent() {

const context = {

popSlices: yield,

mostPopular: yield,

newestSlice: yield,

mostImproved: yield,

pizzas: yield

};

return context;

}

module.exports = mainContent;

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

main.js

var popGen = require('../lib/popGen'),

dataStore = require('../lib/dataStore'),

mainContent = require('../generators/mainContent');

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

main.js

module.exports = function (request, reply) {

const gen = mainContent();

gen.next();

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

main.js

gen.next();

popGen.getPopularSlices()

.then((popSlices) => {

gen.next(popSlices);

})

main.js

.then((popSlices) => {

gen.next(popSlices);

return popGen.getMostPopular();

})

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

main.js

.then((popSlices) => {

gen.next(popSlices);

return popGen.getMostPopular();

})

.then((mostPopular) => {

gen.next(mostPopular);

return popGen.getNewestSlice();

})

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

main.js

.then((newestSlice) => {

gen.next(newestSlice);

return popGen.getMostImproved();

})

.then((mostImproved) => {

gen.next(mostImproved);

return dataStore.getPizzas();

})

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

main.js

.then((pizzas) => {

return reply.view('index', context);

})

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

main.js

.then((pizzas) => {

return reply.view('index', gen.next(pizzas).value);

})

**Exercise03\_04\_01 – Step 3**

itrObject.js

function\* itrObject(obj) {

}

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

itrObject.js

function\* itrObject(obj) {

for (const key in obj) {

if (obj.hasOwnProperty(key)) {

yield {

key,

val: obj[key]

};

}

}

}

module.exports = itrObject;

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

popGen.js

var api = require('./api'),

\_ = require('lodash'),

itrObject = require('../generators/itrObject');

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

popGen.js

for (const prop of itrObject(allQuotes)) {

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

popGen.js

for (const prop in itrObject(allQuotes)) {

diffQuotes.push({

ticker: prop.key,

diff: prop.val[prop.val.length - 1] - prop.val[0],

quote: prop.val[prop.val.length - 1]

});