{

"languageModel": {

"intents": [

{

"name": "AIntent",

"samples": [

"hi",

"hello",

"how",

"what"

],

"slots": []

},

{

"name": "AMAZON.CancelIntent",

"samples": []

},

{

"name": "AMAZON.HelpIntent",

"samples": []

},

{

"name": "AMAZON.StopIntent",

"samples": []

},

{

"name": "MyIntenta",

"samples": [

"Actual Energy Value"

],

"slots": [

{

"name": "ActualEnergyValue",

"type": "AMAZON.City"

}

]

},

{

"name": "MyIntentb",

"samples": [

"Actual Heating Time"

],

"slots": [

{

"name": "ActualHeatingTime",

"type": "AMAZON.City"

}

]

},

{

"name": "MyIntentc",

"samples": [

"Bad Job"

],

"slots": [

{

"name": "BadJob",

"type": "AMAZON.City"

}

]

},

{

"name": "MyIntentd",

"samples": [

"Cumulative Jobs"

],

"slots": [

{

"name": "CumulativeJobs",

"type": "AMAZON.City"

}

]

},

{

"name": "MyIntente",

"samples": [

"Active Axes"

],

"slots": [

{

"name": "ActiveAxes",

"type": "AMAZON.City"

}

]

},

{

"name": "MyIntentf",

"samples": [

"Bloch"

],

"slots": [

{

"name": "Bloch",

"type": "AMAZON.City"

}

]

},

{

"name": "MyIntentg",

"samples": [

"Cycle Time"

],

"slots": [

{

"name": "CycleTime",

"type": "AMAZON.City"

}

]

},

{

"name": "MyIntenti",

"samples": [

"Frequency"

],

"slots": [

{

"name": "Frequency",

"type": "AMAZON.City"

}

]

},

{

"name": "MyIntentj",

"samples": [

"Fovr"

],

"slots": [

{

"name": "Fovr",

"type": "AMAZON.City"

}

]

},

{

"name": "MyIntentk",

"samples": [

"GoodJob"

],

"slots": [

{

"name": "GoodJob",

"type": "AMAZON.City"

}

]

},

{

"name": "MyIntentl",

"samples": [

"Left Side Job Temperature"

],

"slots": [

{

"name": "LeftSideJobTemperature",

"type": "AMAZON.City"

}

]

},

{

"name": "MyIntentm",

"samples": [

"Line"

],

"slots": [

{

"name": "Line",

"type": "AMAZON.City"

}

]

},

{

"name": "MyIntentn",

"samples": [

"Mode"

],

"slots": [

{

"name": "Mode",

"type": "AMAZON.City"

}

]

},

{

"name": "MyIntento",

"samples": [

"Part Count"

],

"slots": [

{

"name": "PartCount",

"type": "AMAZON.City"

}

]

},

{

"name": "MyIntentp",

"samples": [

"Path Federate"

],

"slots": [

{

"name": "PathFederate",

"type": "AMAZON.City"

}

]

},

{

"name": "MyIntentq",

"samples": [

"Path Position"

],

"slots": [

{

"name": "PathPosition",

"type": "AMAZON.City"

}

]

},

{

"name": "MyIntentr",

"samples": [

"Program"

],

"slots": [

{

"name": "Program",

"type": "AMAZON.City"

}

]

},

{

"name": "MyIntents",

"samples": [

"Program Comment"

],

"slots": [

{

"name": "ProgramComment",

"type": "AMAZON.City"

}

]

},

{

"name": "MyIntentt",

"samples": [

"Right Side Job Temperature"

],

"slots": [

{

"name": "RightSideJobTemperature",

"type": "AMAZON.City"

}

]

},

{

"name": "MyIntentu",

"samples": [

"SspeedOvr"

],

"slots": [

{

"name": "SspeedOvr",

"type": "AMAZON.City"

}

]

},

{

"name": "MyIntentv",

"samples": [

"Tool Id"

],

"slots": [

{

"name": "ToolId",

"type": "AMAZON.City"

}

]

},

{

"name": "MyIntentw",

"samples": [

"Total Jobs"

],

"slots": [

{

"name": "TotalJob",

"type": "AMAZON.City"

}

]

},

{

"name": "MyIntentx",

"samples": [

"Voltage"

],

"slots": [

{

"name": "Voltage",

"type": "AMAZON.City"

}

]

}

],

"invocationName": "magic answers"

}

}

// alexa-cookbook sample code

// There are three sections, Text Strings, Skill Code, and Helper Function(s).

// You can copy and paste the entire file contents as the code for a new Lambda function,

// or copy & paste section #3, the helper function, to the bottom of your existing Lambda code.

// 1. Text strings =====================================================================================================

// Modify these strings and messages to change the behavior of your Lambda function

const AWSregion = 'us-east-1'; // us-east-1

const params = {

TableName: 'yesno',

Key:{ "id": "1" }

};

// 2. Skill Code =======================================================================================================

const Alexa = require('alexa-sdk');

const AWS = require('aws-sdk');

AWS.config.update({

region: AWSregion

});

exports.handler = function(event, context, callback) {

var alexa = Alexa.handler(event, context);

// alexa.appId = 'amzn1.echo-sdk-ams.app.1234';

// alexa.dynamoDBTableName = 'YourTableName'; // creates new table for session.attributes

alexa.registerHandlers(handlers);

alexa.execute();

};

const handlers = {

'LaunchRequest': function () {

this.response.speak('welcome to magic answers. ask me a yes or no question.').listen('try again');

this.emit(':responseReady');

},

'MyIntenta': function () {

var ActualEnergyValue = this.event.request.intent.slots.ActualEnergyValue.value;

console.log('ActualEnergyValue : ' + ActualEnergyValue);

readDynamoItem1(params, myResult1=>{

var say1 = '';

say1 = myResult1;

say1 = ' It is ' + myResult1;

this.response.speak(say1).listen('try again');

this.emit(':responseReady');

});

},

'MyIntentb': function () {

var ActualHeatingTime = this.event.request.intent.slots.ActualHeatingTime.value;

console.log('ActualHeatingTime : ' + ActualHeatingTime);

readDynamoItem2(params, myResult2=>{

var say2 = '';

say2 = myResult2;

say2 = ' It is ' + myResult2;

this.response.speak(say2).listen('try again');

this.emit(':responseReady');

});

},

'MyIntentc': function () {

var BadJob = this.event.request.intent.slots.BadJob.value;

console.log('BadJob : ' + BadJob);

readDynamoItem3(params, myResult3=>{

var say3 = '';

say3 = myResult3;

say3 = ' It is ' + myResult3;

this.response.speak(say3).listen('try again');

this.emit(':responseReady');

});

},

'MyIntentd': function () {

var CumulativeJobs = this.event.request.intent.slots.CumulativeJobs.value;

console.log('CumulativeJobs : ' + CumulativeJobs);

readDynamoItem4(params, myResult4=>{

var say4 = '';

say4 = myResult4;

say4 = ' It is ' + myResult4;

this.response.speak(say4).listen('try again');

this.emit(':responseReady');

});

},

'MyIntente': function () {

var ActiveAxes = this.event.request.intent.slots.ActiveAxes.value;

console.log('ActiveAxes : ' + ActiveAxes);

readDynamoItem5(params, myResult5=>{

var say5 = '';

say5 = myResult5;

say5 = ' It is ' + myResult5;

this.response.speak(say5).listen('try again');

this.emit(':responseReady');

});

},

'MyIntentf': function () {

var Bloch = this.event.request.intent.slots.Bloch.value;

console.log('Bloch : ' + Bloch);

readDynamoItem6(params, myResult6=>{

var say6 = '';

say6 = myResult6;

say6 = ' It is ' + myResult6;

this.response.speak(say6).listen('try again');

this.emit(':responseReady');

});

},

'MyIntentg': function () {

var CycleTime = this.event.request.intent.slots.CycleTime.value;

console.log('CycleTime : ' + CycleTime);

readDynamoItem7(params, myResult7=>{

var say7 = '';

say7 = myResult7;

say7 = ' It is ' + myResult7;

this.response.speak(say7).listen('try again');

this.emit(':responseReady');

});

},

/\*

'MyIntenth': function () {

var ActualHeatingTime = this.event.request.intent.slots.ActualHeatingTime.value;

console.log('ActualHeatingTim : ' + ActualHeatingTime);

readDynamoItem8(params, myResult8=>{

var say8 = '';

say8 = myResult8;

say8 = ' It is ' + myResult8;

this.response.speak(say8).listen('try again');

this.emit(':responseReady');

});

},\*/

'MyIntenti': function () {

var Frequency = this.event.request.intent.slots.Frequency.value;

console.log('Frequency : ' + Frequency);

readDynamoItem9(params, myResult9=>{

var say9 = '';

say9 = myResult9;

say9 = ' It is ' + myResult9;

this.response.speak(say9).listen('try again');

this.emit(':responseReady');

});

},

'MyIntentj': function () {

var Fovr = this.event.request.intent.slots.Fovr.value;

console.log('Fovr : ' + Fovr);

readDynamoItem10(params, myResult10=>{

var say10 = '';

say10 = myResult10;

say10 = ' It is ' + myResult10;

this.response.speak(say10).listen('try again');

this.emit(':responseReady');

});

},

'MyIntentk': function () {

var GoodJob = this.event.request.intent.slots.GoodJob.value;

console.log('GoodJob : ' + GoodJob);

readDynamoItem11(params, myResult11=>{

var say11= '';

say11= myResult11;

say11 = ' It is ' + myResult11;

this.response.speak(say11).listen('try again');

this.emit(':responseReady');

});

},

'MyIntentl': function () {

var LeftSideJobTemperature = this.event.request.intent.slots.LeftSideJobTemperature.value;

console.log('LeftSideJobTemperature : ' + LeftSideJobTemperature);

readDynamoItem12(params, myResult12=>{

var say12 = '';

say12 = myResult12;

say12 = ' It is ' + myResult12;

this.response.speak(say12).listen('try again');

this.emit(':responseReady');

});

},

'MyIntentm': function () {

var Line = this.event.request.intent.slots.Line.value;

console.log('Line : ' + Line);

readDynamoItem13(params, myResult13=>{

var say13 = '';

say13 = myResult13;

say13 = ' It is ' + myResult13;

this.response.speak(say13).listen('try again');

this.emit(':responseReady');

});

},

'MyIntentn': function () {

var Mode = this.event.request.intent.slots.Mode.value;

console.log('Mode : ' + Mode);

readDynamoItem14(params, myResult14=>{

var say14 = '';

say14 = myResult14;

say14 = ' It is ' + myResult14;

this.response.speak(say14).listen('try again');

this.emit(':responseReady');

});

},

'MyIntento': function () {

var PartCount = this.event.request.intent.slots.PartCount.value;

console.log('PartCount : ' + PartCount);

readDynamoItem15(params, myResult15=>{

var say15 = '';

say15 = myResult15;

say15= ' It is ' + myResult15;

this.response.speak(say15).listen('try again');

this.emit(':responseReady');

});

},

'MyIntentp': function () {

var PathFederate = this.event.request.intent.slots.PathFederate.value;

console.log('PathFederate : ' + PathFederate);

readDynamoItem16(params, myResult16=>{

var say16 = '';

say16 = myResult16;

say16 = ' It is ' + myResult16;

this.response.speak(say16).listen('try again');

this.emit(':responseReady');

});

},

'MyIntentq': function () {

var PathPosition = this.event.request.intent.slots.PathPosition.value;

console.log('PathPosition : ' + PathPosition);

readDynamoItem17(params, myResult17=>{

var say17 = '';

say17 = myResult17;

say17 = ' It is ' + myResult17;

this.response.speak(say17).listen('try again');

this.emit(':responseReady');

});

},

'MyIntentr': function () {

var Program = this.event.request.intent.slots.Program.value;

console.log('Program : ' + Program);

readDynamoItem18(params, myResult18=>{

var say18 = '';

say18 = myResult18;

say18 = ' It is ' + myResult18;

this.response.speak(say18).listen('try again');

this.emit(':responseReady');

});

},

'MyIntents': function () {

var ProgramComment = this.event.request.intent.slots.ProgramComment.value;

console.log('ProgramComment : ' + ProgramComment);

readDynamoItem19(params, myResult19=>{

var say19 = '';

say19 = myResult19;

say19 = ' It is ' + myResult19;

this.response.speak(say19).listen('try again');

this.emit(':responseReady');

});

},

'MyIntentt': function () {

var RightSideJobTemperature = this.event.request.intent.slots.RightSideJobTemperature.value;

console.log('RightSideJobTemperature : ' + RightSideJobTemperature);

readDynamoItem20(params, myResult20=>{

var say20 = '';

say20 = myResult20;

say20 = ' It is ' + myResult20;

this.response.speak(say20).listen('try again');

this.emit(':responseReady');

});

},

'MyIntentu': function () {

var SspeedOvr = this.event.request.intent.slots.SspeedOvr.value;

console.log('SspeedOvr : ' + SspeedOvr);

readDynamoItem21(params, myResult21=>{

var say21 = '';

say21 = myResult21;

say21 = ' It is ' + myResult21;

this.response.speak(say21).listen('try again');

this.emit(':responseReady');

});

},

'MyIntentv': function () {

var ToolId = this.event.request.intent.slots.ToolId.value;

console.log('SspeedOvr : ' + ToolId);

readDynamoItem22(params, myResult22=>{

var say22 = '';

say22 = myResult22;

say22 = ' It is ' + myResult22;

this.response.speak(say22).listen('try again');

this.emit(':responseReady');

});

},

'MyIntentw': function () {

var TotalJob = this.event.request.intent.slots.TotalJob.value;

console.log('TotalJob : ' + TotalJob);

readDynamoItem23(params, myResult23=>{

var say23 = '';

say23 = myResult23;

say23 = ' It is ' + myResult23;

this.response.speak(say23).listen('try again');

this.emit(':responseReady');

});

},

'MyIntentx': function () {

var Voltage = this.event.request.intent.slots.Voltage.value;

console.log(' Voltage: ' + Voltage);

readDynamoItem24(params, myResult24=>{

var say24 = '';

say24 = myResult24;

say24 = ' It is ' + myResult24;

this.response.speak(say24).listen('try again');

this.emit(':responseReady');

});

},

'AMAZON.HelpIntent': function () {

this.response.speak('ask me a yes or no question.').listen('try again');

this.emit(':responseReady');

},

'AMAZON.CancelIntent': function () {

this.response.speak('Goodbye!');

this.emit(':responseReady');

},

'AMAZON.StopIntent': function () {

this.response.speak('Goodbye!');

this.emit(':responseReady');

}

};

// END of Intent Handlers {} ========================================================================================

// 3. Helper Function =================================================================================================

function readDynamoItem1(params, callback) {

var AWS = require('aws-sdk');

AWS.config.update({region: AWSregion});

var docClient = new AWS.DynamoDB.DocumentClient();

console.log('reading item from DynamoDB table');

docClient.get(params, (err, data) => {

if (err) {

console.error("Unable to read item. Error JSON:", JSON.stringify(err, null, 2));

} else {

console.log("GetItem succeeded:", JSON.stringify(data, null, 2));

callback(data.Item.ActualEnergyValue); // this particular row has an attribute called message

}

});

}

function readDynamoItem2(params, callback) {

var AWS = require('aws-sdk');

AWS.config.update({region: AWSregion});

var docClient = new AWS.DynamoDB.DocumentClient();

console.log('reading item from DynamoDB table');

docClient.get(params, (err, data) => {

if (err) {

console.error("Unable to read item. Error JSON:", JSON.stringify(err, null, 2));

} else {

console.log("GetItem succeeded:", JSON.stringify(data, null, 2));

callback(data.Item.ActualHeatingTime); // this particular row has an attribute called message

}

});

}

function readDynamoItem3(params, callback) {

var AWS = require('aws-sdk');

AWS.config.update({region: AWSregion});

var docClient = new AWS.DynamoDB.DocumentClient();

console.log('reading item from DynamoDB table');

docClient.get(params, (err, data) => {

if (err) {

console.error("Unable to read item. Error JSON:", JSON.stringify(err, null, 2));

} else {

console.log("GetItem succeeded:", JSON.stringify(data, null, 2));

callback(data.Item.BadJob); // this particular row has an attribute called message

}

});

}

function readDynamoItem4(params, callback) {

var AWS = require('aws-sdk');

AWS.config.update({region: AWSregion});

var docClient = new AWS.DynamoDB.DocumentClient();

console.log('reading item from DynamoDB table');

docClient.get(params, (err, data) => {

if (err) {

console.error("Unable to read item. Error JSON:", JSON.stringify(err, null, 2));

} else {

console.log("GetItem succeeded:", JSON.stringify(data, null, 2));

callback(data.Item.CumulativeJobs); // this particular row has an attribute called message

}

});

}

function readDynamoItem5(params, callback) {

var AWS = require('aws-sdk');

AWS.config.update({region: AWSregion});

var docClient = new AWS.DynamoDB.DocumentClient();

console.log('reading item from DynamoDB table');

docClient.get(params, (err, data) => {

if (err) {

console.error("Unable to read item. Error JSON:", JSON.stringify(err, null, 2));

} else {

console.log("GetItem succeeded:", JSON.stringify(data, null, 2));

callback(data.Item.ActiveAxes); // this particular row has an attribute called message

}

});

}

function readDynamoItem6(params, callback) {

var AWS = require('aws-sdk');

AWS.config.update({region: AWSregion});

var docClient = new AWS.DynamoDB.DocumentClient();

console.log('reading item from DynamoDB table');

docClient.get(params, (err, data) => {

if (err) {

console.error("Unable to read item. Error JSON:", JSON.stringify(err, null, 2));

} else {

console.log("GetItem succeeded:", JSON.stringify(data, null, 2));

callback(data.Item.Bloch); // this particular row has an attribute called message

}

});

}function readDynamoItem7(params, callback) {

var AWS = require('aws-sdk');

AWS.config.update({region: AWSregion});

var docClient = new AWS.DynamoDB.DocumentClient();

console.log('reading item from DynamoDB table');

docClient.get(params, (err, data) => {

if (err) {

console.error("Unable to read item. Error JSON:", JSON.stringify(err, null, 2));

} else {

console.log("GetItem succeeded:", JSON.stringify(data, null, 2));

callback(data.Item.CycleTime); // this particular row has an attribute called message

}

});

/\*

}function readDynamoItem8(params, callback) {

var AWS = require('aws-sdk');

AWS.config.update({region: AWSregion});

var docClient = new AWS.DynamoDB.DocumentClient();

console.log('reading item from DynamoDB table');

docClient.get(params, (err, data) => {

if (err) {

console.error("Unable to read item. Error JSON:", JSON.stringify(err, null, 2));

} else {

console.log("GetItem succeeded:", JSON.stringify(data, null, 2));

callback(data.Item.CumulativeJobs); // this particular row has an attribute called message

}

});

\*/

}function readDynamoItem9(params, callback) {

var AWS = require('aws-sdk');

AWS.config.update({region: AWSregion});

var docClient = new AWS.DynamoDB.DocumentClient();

console.log('reading item from DynamoDB table');

docClient.get(params, (err, data) => {

if (err) {

console.error("Unable to read item. Error JSON:", JSON.stringify(err, null, 2));

} else {

console.log("GetItem succeeded:", JSON.stringify(data, null, 2));

callback(data.Item.Frequency); // this particular row has an attribute called message

}

});

}function readDynamoItem10(params, callback) {

var AWS = require('aws-sdk');

AWS.config.update({region: AWSregion});

var docClient = new AWS.DynamoDB.DocumentClient();

console.log('reading item from DynamoDB table');

docClient.get(params, (err, data) => {

if (err) {

console.error("Unable to read item. Error JSON:", JSON.stringify(err, null, 2));

} else {

console.log("GetItem succeeded:", JSON.stringify(data, null, 2));

callback(data.Item.Fovr); // this particular row has an attribute called message

}

});

}function readDynamoItem11(params, callback) {

var AWS = require('aws-sdk');

AWS.config.update({region: AWSregion});

var docClient = new AWS.DynamoDB.DocumentClient();

console.log('reading item from DynamoDB table');

docClient.get(params, (err, data) => {

if (err) {

console.error("Unable to read item. Error JSON:", JSON.stringify(err, null, 2));

} else {

console.log("GetItem succeeded:", JSON.stringify(data, null, 2));

callback(data.Item.GoodJob); // this particular row has an attribute called message

}

});

}function readDynamoItem12(params, callback) {

var AWS = require('aws-sdk');

AWS.config.update({region: AWSregion});

var docClient = new AWS.DynamoDB.DocumentClient();

console.log('reading item from DynamoDB table');

docClient.get(params, (err, data) => {

if (err) {

console.error("Unable to read item. Error JSON:", JSON.stringify(err, null, 2));

} else {

console.log("GetItem succeeded:", JSON.stringify(data, null, 2));

callback(data.Item.LeftSideJobTemperature); // this particular row has an attribute called message

}

});

}function readDynamoItem13(params, callback) {

var AWS = require('aws-sdk');

AWS.config.update({region: AWSregion});

var docClient = new AWS.DynamoDB.DocumentClient();

console.log('reading item from DynamoDB table');

docClient.get(params, (err, data) => {

if (err) {

console.error("Unable to read item. Error JSON:", JSON.stringify(err, null, 2));

} else {

console.log("GetItem succeeded:", JSON.stringify(data, null, 2));

callback(data.Item.Line); // this particular row has an attribute called message

}

});

}function readDynamoItem14(params, callback) {

var AWS = require('aws-sdk');

AWS.config.update({region: AWSregion});

var docClient = new AWS.DynamoDB.DocumentClient();

console.log('reading item from DynamoDB table');

docClient.get(params, (err, data) => {

if (err) {

console.error("Unable to read item. Error JSON:", JSON.stringify(err, null, 2));

} else {

console.log("GetItem succeeded:", JSON.stringify(data, null, 2));

callback(data.Item.Mode); // this particular row has an attribute called message

}

});

}function readDynamoItem15(params, callback) {

var AWS = require('aws-sdk');

AWS.config.update({region: AWSregion});

var docClient = new AWS.DynamoDB.DocumentClient();

console.log('reading item from DynamoDB table');

docClient.get(params, (err, data) => {

if (err) {

console.error("Unable to read item. Error JSON:", JSON.stringify(err, null, 2));

} else {

console.log("GetItem succeeded:", JSON.stringify(data, null, 2));

callback(data.Item.PartCount); // this particular row has an attribute called message

}

});

}function readDynamoItem16(params, callback) {

var AWS = require('aws-sdk');

AWS.config.update({region: AWSregion});

var docClient = new AWS.DynamoDB.DocumentClient();

console.log('reading item from DynamoDB table');

docClient.get(params, (err, data) => {

if (err) {

console.error("Unable to read item. Error JSON:", JSON.stringify(err, null, 2));

} else {

console.log("GetItem succeeded:", JSON.stringify(data, null, 2));

callback(data.Item.PathFederate); // this particular row has an attribute called message

}

});

}function readDynamoItem17(params, callback) {

var AWS = require('aws-sdk');

AWS.config.update({region: AWSregion});

var docClient = new AWS.DynamoDB.DocumentClient();

console.log('reading item from DynamoDB table');

docClient.get(params, (err, data) => {

if (err) {

console.error("Unable to read item. Error JSON:", JSON.stringify(err, null, 2));

} else {

console.log("GetItem succeeded:", JSON.stringify(data, null, 2));

callback(data.Item.PathPosition); // this particular row has an attribute called message

}

});

}function readDynamoItem18(params, callback) {

var AWS = require('aws-sdk');

AWS.config.update({region: AWSregion});

var docClient = new AWS.DynamoDB.DocumentClient();

console.log('reading item from DynamoDB table');

docClient.get(params, (err, data) => {

if (err) {

console.error("Unable to read item. Error JSON:", JSON.stringify(err, null, 2));

} else {

console.log("GetItem succeeded:", JSON.stringify(data, null, 2));

callback(data.Item.Program); // this particular row has an attribute called message

}

});

}function readDynamoItem19(params, callback) {

var AWS = require('aws-sdk');

AWS.config.update({region: AWSregion});

var docClient = new AWS.DynamoDB.DocumentClient();

console.log('reading item from DynamoDB table');

docClient.get(params, (err, data) => {

if (err) {

console.error("Unable to read item. Error JSON:", JSON.stringify(err, null, 2));

} else {

console.log("GetItem succeeded:", JSON.stringify(data, null, 2));

callback(data.Item.ProgramComment); // this particular row has an attribute called message

}

});

}function readDynamoItem20(params, callback) {

var AWS = require('aws-sdk');

AWS.config.update({region: AWSregion});

var docClient = new AWS.DynamoDB.DocumentClient();

console.log('reading item from DynamoDB table');

docClient.get(params, (err, data) => {

if (err) {

console.error("Unable to read item. Error JSON:", JSON.stringify(err, null, 2));

} else {

console.log("GetItem succeeded:", JSON.stringify(data, null, 2));

callback(data.Item.RightSideJobTemperature); // this particular row has an attribute called message

}

});

}function readDynamoItem21(params, callback) {

var AWS = require('aws-sdk');

AWS.config.update({region: AWSregion});

var docClient = new AWS.DynamoDB.DocumentClient();

console.log('reading item from DynamoDB table');

docClient.get(params, (err, data) => {

if (err) {

console.error("Unable to read item. Error JSON:", JSON.stringify(err, null, 2));

} else {

console.log("GetItem succeeded:", JSON.stringify(data, null, 2));

callback(data.Item.SspeedOvr); // this particular row has an attribute called message

}

});

}

function readDynamoItem22(params, callback) {

var AWS = require('aws-sdk');

AWS.config.update({region: AWSregion});

var docClient = new AWS.DynamoDB.DocumentClient();

console.log('reading item from DynamoDB table');

docClient.get(params, (err, data) => {

if (err) {

console.error("Unable to read item. Error JSON:", JSON.stringify(err, null, 2));

} else {

console.log("GetItem succeeded:", JSON.stringify(data, null, 2));

callback(data.Item.ToolId); // this particular row has an attribute called message

}

});

}

function readDynamoItem23(params, callback) {

var AWS = require('aws-sdk');

AWS.config.update({region: AWSregion});

var docClient = new AWS.DynamoDB.DocumentClient();

console.log('reading item from DynamoDB table');

docClient.get(params, (err, data) => {

if (err) {

console.error("Unable to read item. Error JSON:", JSON.stringify(err, null, 2));

} else {

console.log("GetItem succeeded:", JSON.stringify(data, null, 2));

callback(data.Item.TotalJob); // this particular row has an attribute called message

}

});

}

function readDynamoItem24(params, callback) {

var AWS = require('aws-sdk');

AWS.config.update({region: AWSregion});

var docClient = new AWS.DynamoDB.DocumentClient();

console.log('reading item from DynamoDB table');

docClient.get(params, (err, data) => {

if (err) {

console.error("Unable to read item. Error JSON:", JSON.stringify(err, null, 2));

} else {

console.log("GetItem succeeded:", JSON.stringify(data, null, 2));

callback(data.Item.Voltage); // this particular row has an attribute called message

}

});

}