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

var request = require('request');

var customvars = (event['customVars']!==undefined) ? event['customVars'] : {};

try{

customvars = JSON.parse(customvars);

}catch(e){}

function isValidDate(date) {

var n="";

if(date.length>=6){

// var sp = n.split('/');

var n = date ;

var dateReg =/^(0?[1-9]|[12][0-9]|3[01])([- /.])(0?[1-9]|1[012])\2(19|20)\d\d$/;

if(date.match(dateReg)==null){return false;}

}else if(date.length>=3){

var obj = new Date();

var n = date+"/"+obj.getFullYear();

}

//console.log("DATE: "+ n);

var temp = n.split(/[\.\-\/]/);

var d = new Date(temp[2] + '/' + temp[1] + '/' + temp[0]);

// return (d && (d.getMonth() + 1) == temp[1] && d.getDate() == Number(temp[0]) && d.getFullYear() == (temp[2]));

if((d && (d.getMonth() + 1) == temp[1] && d.getDate() == Number(temp[0]) && d.getFullYear() == (temp[2]))){

return n;

}

}

var valid\_dt=isValidDate(event.reply);

var validating\_date="";var date\_checker="";

if(valid\_dt!==undefined && valid\_dt!==null && valid\_dt!==false){

validating\_date=valid\_dt;

console.log("valid date "+valid\_dt);

var split\_dt= valid\_dt.split(/[\.\-\/]/);

var selectedDate = new Date(split\_dt[1]+"/"+split\_dt[0]+"/"+split\_dt[2]); //MM-DD-YYYY

var currentDate = new Date(dateobj.getFullYear(),dateobj.getMonth(),dateobj.getDate());

if (selectedDate >= currentDate) {

date\_checker=true;

console.log("valid date+Future Date/current date");

}else{

date\_checker=false;

console.log("Provide future date:");

}

}else{

console.log("invalid date "+event.reply);

validating\_date=event.reply;

}

var responseJSON="";

if(date\_checker!=="" && date\_checker!==false){

responseJSON = {

"response": "", // what the bot will respond with

"continue": true, // "true" will result in Motion AI continuing the flow based on connections, whie "false" will make Motion AI hit this module again when the user replies

"customPayload": "", // OPTIONAL: working data to examine in future calls to this function to keep track of state

"quickReplies": null, // OPTIONAL: a JSON string array containing suggested/quick replies to display to the user .. i.e., ["Hello","World"]

"cards": null, // OPTIONAL: a cards JSON object to display a carousel to the user (see docs)

"customVars": {valid\_date:validating\_date}, // OPTIONAL: an object or stringified object with key-value pairs to set custom variables eg: {"key":"value"} or '{"key":"value"}'

"nextModule": 750543 // // OPTIONAL: the ID of a module to follow this Node JS module

}

}else{

responseJSON = {

"response": customvars.\_date\_error\_, // what the bot will respond with

"continue": true, // "true" will result in Motion AI continuing the flow based on connections, whie "false" will make Motion AI hit this module again when the user replies

"customPayload": "",

"quickReplies": null, // OPTIONAL: a JSON string array containing suggested/quick replies to display to the user .. i.e., ["Hello","World"]

"cards": null, // OPTIONAL: a cards JSON object to display a carousel to the user (see docs)

"customVars": {invalid\_date:event.reply}, // OPTIONAL: an object or stringified object with key-value pairs to set custom variables eg: {"key":"value"} or '{"key":"value"}'

"nextModule":"" // // OPTIONAL: the ID of a module to follow this Node JS module

}

}

console.log("Resposne "+JSON.stringify(responseJSON));

// callback to return data to Motion AI (must exist, or bot will not work)

callback(null, responseJSON);

};