Index.js

'use strict';

const functions = require('firebase-functions');

const admin=require('firebase-admin');

const {WebhookClient} = require('dialogflow-fulfillment');

const {Card, Suggestion} = require('dialogflow-fulfillment');

process.env.DEBUG = 'dialogflow:debug'; // enables lib debugging statements

admin.initializeApp(functions.config().firebase);

const db=admin.firestore();

exports.dialogflowFirebaseFulfillment = functions.https.onRequest((request, response) => {

  const agent = new WebhookClient({ request, response });

  console.log('Dialogflow Request headers: ' + JSON.stringify(request.headers));

  console.log('Dialogflow Request body: ' + JSON.stringify(request.body));

  function welcome(agent) {

    agent.add(`Welcome to my agent now!`);

  }

  function fallback(agent) {

    agent.add(`I didn't understand`);

    agent.add(`I'm sorry, can you try again?`);

  }

  function writeToDb(agent){

      const databaseEntry=agent.parameters.databaseEntry;

      const dialogflowAgentRef=db.collection('dialogflow').doc('agent');

      return db.runTransaction(t=>{

          t.set(dialogflowAgentRef, {entry: databaseEntry});

          return Promise.resolve('Write complete');

      })

      .then(doc=>{

          agent.add(`Wrote "${databaseEntry}" to the firestore database.`);

      }).catch(err=>{

          console.log(`Error writing to firestore: ${err}`);

          agent.add(`Failed to write "${databaseEntry}" to firestore database.`);

      });

  }

  function readFromDb(agent){

      const dialogflowAgentDoc=db.collection('dialogflow').doc('agent');

      return dialogflowAgentDoc.get()

      .then(doc=>{

          if(!doc.exists){

              agent.add('No data found in database.');

          } else{

              agent.add(doc.data().entry);

          }

          return Promise.resolve('Read complete.');

      }).catch(()=>{

          agent.add('Error reading entry from database.');

          agent.add('Please add an entry to database by saying something.');

      });

  }

  function account\_general\_savings\_client\_balance(agent){

    const dialogflowAgentDoc=db.collection('dialogflow').doc('agent');

    return dialogflowAgentDoc.get()

    .then(doc=>{

        if(!doc.exists){

            agent.add('No data found in database.');

        } else{

            agent.add('Here is account savings balance'+doc.data().account\_saving);

        }

        return Promise.resolve('Read complete.');

    }).catch(()=>{

        agent.add('Error reading entry from database.');

        agent.add('Please add an entry to database by saying something.');

    });

}

  // // Uncomment and edit to make your own intent handler

  // // uncomment `intentMap.set('your intent name here', yourFunctionHandler);`

  // // below to get this function to be run when a Dialogflow intent is matched

  // function yourFunctionHandler(agent) {

  //   agent.add(`This message is from Dialogflow's Cloud Functions for Firebase editor!`);

  //   agent.add(new Card({

  //       title: `Title: this is a card title`,

  //       imageUrl: 'https://developers.google.com/actions/images/badges/XPM\_BADGING\_GoogleAssistant\_VER.png',

  //       text: `This is the body text of a card.  You can even use line\n  breaks and emoji! 💁`,

  //       buttonText: 'This is a button',

  //       buttonUrl: 'https://assistant.google.com/'

  //     })

  //   );

  //   agent.add(new Suggestion(`Quick Reply`));

  //   agent.add(new Suggestion(`Suggestion`));

  //   agent.setContext({ name: 'weather', lifespan: 2, parameters: { city: 'Rome' }});

  // }

  // // Uncomment and edit to make your own Google Assistant intent handler

  // // uncomment `intentMap.set('your intent name here', googleAssistantHandler);`

  // // below to get this function to be run when a Dialogflow intent is matched

  // function googleAssistantHandler(agent) {

  //   let conv = agent.conv(); // Get Actions on Google library conv instance

  //   conv.ask('Hello from the Actions on Google client library!') // Use Actions on Google library

  //   agent.add(conv); // Add Actions on Google library responses to your agent's response

  // }

  // // See https://github.com/dialogflow/fulfillment-actions-library-nodejs

  // // for a complete Dialogflow fulfillment library Actions on Google client library v2 integration sample

  // Run the proper function handler based on the matched Dialogflow intent name

  let intentMap = new Map();

  intentMap.set('Default Welcome Intent', welcome);

  intentMap.set('write\_to\_db', writeToDb);

  intentMap.set('read\_from\_db', readFromDb);

  intentMap.set('account\_general\_savings\_client\_balance', account\_general\_savings\_client\_balance);

  intentMap.set('Default Fallback Intent', fallback);

  // intentMap.set('your intent name here', yourFunctionHandler);

  // intentMap.set('your intent name here', googleAssistantHandler);

  agent.handleRequest(intentMap);

});

Package.json

{

  "name": "dialogflowFirebaseFulfillment",

  "description": "This is the default fulfillment for a Dialogflow agents using Cloud Functions for Firebase",

  "version": "0.0.1",

  "private": true,

  "license": "Apache Version 2.0",

  "author": "Google Inc.",

  "engines": {

    "node": "8"

  },

  "scripts": {

    "start": "firebase serve --only functions:dialogflowFirebaseFulfillment",

    "deploy": "firebase deploy --only functions:dialogflowFirebaseFulfillment"

  },

  "dependencies": {

    "actions-on-google": "^2.2.0",

    "dialogflow": "^0.6.0",

    "dialogflow-fulfillment": "^0.5.0",

    "firebase-admin": "^8.0.0",

    "firebase-functions": "^3.3.0"

  }

}