```
// See https://github.com/dialogflow/dialogflow-fulfillment-nodejs
// for Dialogflow fulfillment library docs, samples, and to report issues
'use strict';

const functions = require('firebase-functions');
const {WebhookClient} = require('dialogflow-fulfillment');
const {Card, Suggestion} = require('dialogflow-fulfillment');
const admin = require("firebase-admin");

admin.initializeApp();
let db = admin.database();
let ref = db.ref("games");

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

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(new Card({
           title: `Title: this is a card title`,
           imageUrl: 'https://developers.google.com/actions/images/badges/XPM_BADGING_GoogleAssistant_VER.pn
           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(`Higher`));
       agent.add(new Suggestion(`Lower`));
       agent.add(new Suggestion(`Correct`));
  }

  function fallback(agent)
  {
    agent.add(`I didn't understand`);
    agent.add(`I'm sorry, can you try again?`);
  }

  function setName(agent)
  {
      let session = agent.session;
      let name = agent.parameters["given-name"];
      let key = session.substring(session.lastIndexOf('/') + 1).replace("-", "_");

      agent.add(`setting name to ${name} for session ${session} -> ${key}`);
      let obj = {};
      obj[key] = {name: name};

      return ref.set(obj);
  }

  function getName(agent)
  {
      let session = agent.session;
      let key = session.substring(session.lastIndexOf('/') + 1).replace("-", "_");

      return db.ref("games/" + key).once("value").then((snapshot) =>
      {
          let name = snapshot.child("name").val();

          agent.add(`${name} for session ${session}`)
      });
  }

  // // 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.pn
  //       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('Default Fallback Intent', fallback);
  intentMap.set('my-name-is', setName);
  intentMap.set('what-is-my-name', getName);
  agent.handleRequest(intentMap);
});
```