'use strict';

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

//=========================================================================================================================================

//TODO: The items below this comment need your attention

//=========================================================================================================================================

//Replace with your app ID (OPTIONAL). You can find this value at the top of your skill's page on http://developer.amazon.com.

//Make sure to enclose your value in quotes, like this: var APP\_ID = "amzn1.ask.skill.bb4045e6-b3e8-4133-b650-72923c5980f1";

var APP\_ID = "amzn1.ask.skill.35fb829a-5562-4eb5-8832-a22d8f469f21";

//This function returns a descriptive sentence about your data. Before a user starts a quiz, they can ask about a specific data element,

//like "Ohio." The skill will speak the sentence from this function, pulling the data values from the appropriate record in your data.

function getSpeechDescription(item)

{

var sentence = "<break time='1s'/>" + item.StateName + " in London has a postcode of " + " <break strength='strong'/><say-as interpret-as='digits'>" + item.Postcode + "</say-as>. Which other postcode would you like to know about, or you can say start a quiz?";

return sentence;

}

//We have provided two ways to create your quiz questions. The default way is to phrase all of your questions like: "What is X of Y?"

//If this approach doesn't work for your data, take a look at the commented code in this function. You can write a different question

//structure for each property of your data.

function getQuestion(counter, property, item)

{

return "Here is your " + counter + "th question. What is the " + formatCasing(property) + " of " + item.StateName + "?";

/\*

switch(property)

{

case "City":

return "Here is your " + counter + "th question. In what city do the " + item.League + "'s " + item.Mascot + " play?";

break;

case "Sport":

return "Here is your " + counter + "th question. What sport do the " + item.City + " " + item.Mascot + " play?";

break;

case "HeadCoach":

return "Here is your " + counter + "th question. Who is the head coach of the " + item.City + " " + item.Mascot + "?";

break;

default:

return "Here is your " + counter + "th question. What is the " + formatCasing(property) + " of the " + item.Mascot + "?";

break;

}

\*/

}

//This is the function that returns an answer to your user during the quiz. Much like the "getQuestion" function above, you can use a

//switch() statement to create different responses for each property in your data. For example, when this quiz has an answer that includes

//a state abbreviation, we add some SSML to make sure that Alexa spells that abbreviation out (instead of trying to pronounce it.)

function getAnswer(property, item)

{

switch(property)

{

case "Abbreviation":

return "The " + formatCasing(property) + " of " + item.StateName + " is <say-as interpret-as='spell-out'>" + item[property] + "</say-as>. "

break;

default:

return "The " + formatCasing(property) + " of " + item.StateName + " is " + item[property] + ". "

break;

}

}

//This is a list of positive speechcons that this skill will use when a user gets a correct answer. For a full list of supported

//speechcons, go here: https://developer.amazon.com/public/solutions/alexa/alexa-skills-kit/docs/speechcon-reference

var speechConsCorrect = ["Booya", "All righty", "Bam", "Bazinga", "Bingo", "Boom", "Bravo", "Cha Ching", "Cheers", "Dynomite",

"Hip hip hooray", "Hurrah", "Hurray", "Huzzah", "Oh dear. Just kidding. Hurray", "Kaboom", "Kaching", "Oh snap", "Phew",

"Righto", "Way to go", "Well done", "Whee", "Woo hoo", "Yay", "Wowza", "Yowsa"];

//This is a list of negative speechcons that this skill will use when a user gets an incorrect answer. For a full list of supported

//speechcons, go here: https://developer.amazon.com/public/solutions/alexa/alexa-skills-kit/docs/speechcon-reference

var speechConsWrong = ["Argh", "Aw man", "Blarg", "Blast", "Boo", "Bummer", "Darn", "D'oh", "Dun dun dun", "Eek", "Honk", "Le sigh",

"Mamma mia", "Oh boy", "Oh dear", "Oof", "Ouch", "Ruh roh", "Shucks", "Uh oh", "Wah wah", "Whoops a daisy", "Yikes"];

//This is the welcome message for when a user starts the skill without a specific intent.

var WELCOME\_MESSAGE = "Welcome to the London Postcodes Quiz Game! You can ask me about a postcode, or you can ask me to start a quiz. What would you like to do?";

//This is the message a user will hear when they start a quiz.

var START\_QUIZ\_MESSAGE = "OK. I will ask you 10 questions about the London Postcodes.";

//This is the message a user will hear when they try to cancel or stop the skill, or when they finish a quiz.

var EXIT\_SKILL\_MESSAGE = "Thank you for playing the London Postcodes Quiz Game! Let's play again soon!";

//This is the message a user will hear after they ask (and hear) about a specific data element.

var REPROMPT\_SPEECH = "Which other London area would you like to know about?";

//This is the message a user will hear when they ask Alexa for help in your skill.

var HELP\_MESSAGE = "I know lots of things about London Postcodes. You can ask me about a postcode, and I'll tell you what I know. You can also test your knowledge by asking me to start a quiz. What would you like to do?";

//This is the response a user will receive when they ask about something we weren't expecting. For example, say "pizza" to your

//skill when it starts. This is the response you will receive.

function getBadAnswer(item) { return "I'm sorry. " + item + " is not something I know very much about in this skill. " + HELP\_MESSAGE; }

//This is the message a user will receive after each question of a quiz. It reminds them of their current score.

function getCurrentScore(score, counter) { return "Your current score is " + score + " out of " + counter + ". "; }

//This is the message a user will receive after they complete a quiz. It tells them their final score.

function getFinalScore(score, counter) { return "Your final score is " + score + " out of " + counter + ". "; }

//These next four values are for the Alexa cards that are created when a user asks about one of the data elements.

//This only happens outside of a quiz.

//If you don't want to use cards in your skill, set the USE\_CARDS\_FLAG to false. If you set it to true, you will need an image for each

//item in your data.

var USE\_CARDS\_FLAG = false;

//This is what your card title will be. For our example, we use the name of the state the user requested.

function getCardTitle(item) { return item.StateName;}

//This is the small version of the card image. We use our data as the naming convention for our images so that we can dynamically

//generate the URL to the image. The small image should be 720x400 in dimension.

//function getSmallImage(item) { return "https://m.media-amazon.com/images/G/01/mobile-apps/dex/alexa/alexa-skills-kit/tutorials/quiz-game/state\_flag/720x400/" + item.Abbreviation + ".\_TTH\_.png"; }

//This is the large version of the card image. It should be 1200x800 pixels in dimension.

//function getLargeImage(item) { return "https://m.media-amazon.com/images/G/01/mobile-apps/dex/alexa/alexa-skills-kit/tutorials/quiz-game/state\_flag/1200x800/" + item.Abbreviation + ".\_TTH\_.png"; }

//=========================================================================================================================================

//TODO: Replace this data with your own.

//=========================================================================================================================================

var data = [

{StateName: "Mile End Stepney Whitechapel", Postcode: "EAST 1" },

{StateName: "Bethnal Green Shoreditch", Postcode: "EAST 2" },

{StateName: "Bow Bromley by Bow", Postcode: "EAST 3" },

{StateName: "Chingford Highams Park", Postcode: "EAST 4" },

{StateName: "Clapton", Postcode: "EAST 5" },

{StateName: "East Ham Beckton", Postcode: "EAST 6" },

{StateName: "Forest gate", Postcode: "EAST 7" },

{StateName: "Hackney Dalston", Postcode: "EAST 8" },

{StateName: "Hackney Homerton", Postcode: "EAST 9" },

{StateName: "Leyton", Postcode: "EAST 10" },

{StateName: "Leytonstone", Postcode: "EAST 11" },

{StateName: "Manor Park", Postcode: "EAST 12" },

{StateName: "Plaistow", Postcode: "EAST 13" },

{StateName: "Isle of Dogs Millwall Poplar ", Postcode: "EAST 14" },

{StateName: "Stratford West Ham", Postcode: "EAST 15" },

{StateName: "Canning Town North Woolwich", Postcode: "EAST 16" },

{StateName: "Walthamstow", Postcode: "EAST 17" },

{StateName: "South Woodford", Postcode: "EAST 18" },

{StateName: "Olympic Park Stratford", Postcode: "EAST 20" },

{StateName: "Barnsbury Canonbury Islington", Postcode: "NORTH 1" },

{StateName: "East Finchley", Postcode: "NORTH 2" },

{StateName: "Finchley Central", Postcode: "NORTH 3" },

{StateName: "Finsbury Park Manor House", Postcode: "NORTH 4" },

{StateName: "Highbury", Postcode: "NORTH 5" },

{StateName: "Highgate", Postcode: "NORTH 6" },

{StateName: "Holloway", Postcode: "NORTH 7" },

{StateName: "Crouch End Hornsey", Postcode: "NORTH 8" },

{StateName: "Lower Edmonton", Postcode: "NORTH 9" },

{StateName: "Muswell Hill", Postcode: "NORTH 10" },

{StateName: "Friern Barnet New Southgate", Postcode: "NORTH 11" },

{StateName: "North Finchley Woodside Park", Postcode: "NORTH 12" },

{StateName: "Palmers Green", Postcode: "NORTH 13" },

{StateName: "Southgate", Postcode: "NORTH 14" },

{StateName: "Seven Sisters", Postcode: "NORTH 15" },

{StateName: "Stamford Hill Stoke Newington", Postcode: "NORTH 16" },

{StateName: "Tottenham", Postcode: "NORTH 17" },

{StateName: "Upper Edmonton", Postcode: "NORTH 18" },

{StateName: "Archway Tufnell Park", Postcode: "NORTH 19" },

{StateName: "Totteridge Whetstone", Postcode: "NORTH 20" },

{StateName: "Winchmore Hill", Postcode: "NORTH 21" },

{StateName: "Alexandra Palace Wood Green", Postcode: "NORTH 22" },

{StateName: "Bermondsey Borough Southwark Waterloo", Postcode: "SOUTH EAST 1" },

{StateName: "Abbey Wood", Postcode: "SOUTH EAST 2" },

{StateName: "Blackheath Westcombe Park", Postcode: "SOUTH EAST 3" },

{StateName: "Brockley Crofton Park Honor Oak Park", Postcode: "SOUTH EAST 4" },

{StateName: "Camberwell", Postcode: "SOUTH EAST 5" },

{StateName: "Bellingham Catford Hither Green", Postcode: "SOUTH EAST 6" },

{StateName: "Charlton", Postcode: "SOUTH EAST 7" },

{StateName: "Deptford", Postcode: "SOUTH EAST 8" },

{StateName: "Eltham Mottingham", Postcode: "SOUTH EAST 9" },

{StateName: "Greenwich", Postcode: "SOUTH EAST 10" },

{StateName: "Lambeth", Postcode: "SOUTH EAST 11" },

{StateName: "Grove Park Lee", Postcode: "SOUTH EAST 12" },

{StateName: "Hither Green Lewisham", Postcode: "SOUTH EAST 13" },

{StateName: "New Cross New Cross Gate", Postcode: "SOUTH EAST 14" },

{StateName: "Nunhead Peckham", Postcode: "SOUTH EAST 15" },

{StateName: "Rotherhithe South Bermondsey Surrey Docks", Postcode: "SOUTH EAST 16" },

{StateName: "Elephant & Castle Walworth", Postcode: "SOUTH EAST 17" },

{StateName: "Plumstead Woolwich", Postcode: "SOUTH EAST 18" },

{StateName: "Crystal Palace Upper Norwood", Postcode: "SOUTH EAST 19" },

{StateName: "Anerley Penge", Postcode: "SOUTH EAST 20" },

{StateName: "Dulwich", Postcode: "SOUTH EAST 21" },

{StateName: "East Dulwich", Postcode: "SOUTH EAST 22" },

{StateName: "Forest Hill", Postcode: "SOUTH EAST 23" },

{StateName: "Herne Hill", Postcode: "SOUTH EAST 24" },

{StateName: "South Norwood", Postcode: "SOUTH EAST 25" },

{StateName: "Sydenham", Postcode: "SOUTH EAST 26" },

{StateName: "Tulse Hill West Norwood", Postcode: "SOUTH EAST 27" },

{StateName: "Thamesmead", Postcode: "SOUTH EAST 28" },

{StateName: "Belgravia Pimlico Westminster", Postcode: "SOUTH WEST 1" },

{StateName: "Brixton Streatham Hill", Postcode: "SOUTH WEST 2" },

{StateName: "Brompton Chelsea", Postcode: "SOUTH WEST 3" },

{StateName: "Clapham", Postcode: "SOUTH WEST 4" },

{StateName: "Earl's Court", Postcode: "SOUTH WEST 5" },

{StateName: "Fulham Parson's Green", Postcode: "SOUTH WEST 6" },

{StateName: "South Kensington", Postcode: "SOUTH WEST 7" },

{StateName: "Nine Elms South Lambeth", Postcode: "SOUTH WEST 8" },

{StateName: "Brixton Stockwell", Postcode: "SOUTH WEST 9" },

{StateName: "West Brompton World's End", Postcode: "SOUTH WEST 10" },

{StateName: "Battersea Clapham Junction", Postcode: "SOUTH WEST 11" },

{StateName: "Balham", Postcode: "SOUTH WEST 12" },

{StateName: "Barnes Castelnau", Postcode: "SOUTH WEST 13" },

{StateName: "East Sheen Mortlake", Postcode: "SOUTH WEST 14" },

{StateName: "Putney Roehampton", Postcode: "SOUTH WEST 15" },

{StateName: "Norbury Streatham", Postcode: "SOUTH WEST 16" },

{StateName: "Tooting", Postcode: "SOUTH WEST 17" },

{StateName: "Earlsfield Wandsworth", Postcode: "SOUTH WEST 18" },

{StateName: "Merton Wimbledon", Postcode: "SOUTH WEST 19" },

{StateName: "Raynes Park South Wimbledon", Postcode: "SOUTH WEST 20" },

{StateName: "Marylebone Mayfair Soho", Postcode: "WEST 1" },

{StateName: "Bayswater Paddington", Postcode: "WEST 2" },

{StateName: "Acton", Postcode: "WEST 3" },

{StateName: "Chiswick", Postcode: "WEST 4" },

{StateName: "Ealing", Postcode: "WEST 5" },

{StateName: "Hammersmith", Postcode: "WEST 6" },

{StateName: "Hanwell", Postcode: "WEST 7" },

{StateName: "Kensington", Postcode: "WEST 8" },

{StateName: "Maida Vale Warwick Avenue", Postcode: "WEST 9" },

{StateName: "Ladbroke Grove North Kensington", Postcode: "WEST 10" },

{StateName: "Holland Park Notting Hill", Postcode: "WEST 11" },

{StateName: "Shepherd's Bush", Postcode: "WEST 12" },

{StateName: "West Ealing", Postcode: "WEST 13" },

{StateName: "West Kensington", Postcode: "WEST 14" },

{StateName: "Camden Town Regent's Park", Postcode: "NORTH WEST 1" },

{StateName: "Cricklewood Neasden", Postcode: "NORTH WEST 2" },

{StateName: "Hampstead Swiss Cottage", Postcode: "NORTH WEST 3" },

{StateName: "Brent Cross Hendon", Postcode: "NORTH WEST 4" },

{StateName: "Kentish Town", Postcode: "NORTH WEST 5" },

{StateName: "Kilburn Queens Park West Hampstead", Postcode: "NORTH WEST 6" },

{StateName: "Mill Hill", Postcode: "NORTH WEST 7" },

{StateName: "Saint John's Wood", Postcode: "NORTH WEST 8" },

{StateName: "Colindale Kingsbury", Postcode: "NORTH WEST 9" },

{StateName: "Harlesden Kensal Green Willesden", Postcode: "NORTH WEST 10" },

{StateName: "Golders Green Hampstead Garden Suburb", Postcode: "NORTH WEST 11" }

];

//=========================================================================================================================================

//Editing anything below this line might break your skill.

//=========================================================================================================================================

var counter = 0;

var states = {

START: "\_START",

QUIZ: "\_QUIZ"

};

const handlers = {

"LaunchRequest": function() {

this.handler.state = states.START;

this.emitWithState("Start");

},

"QuizIntent": function() {

this.handler.state = states.QUIZ;

this.emitWithState("Quiz");

},

"AnswerIntent": function() {

this.handler.state = states.START;

this.emitWithState("AnswerIntent");

},

"AMAZON.HelpIntent": function() {

this.emit(":ask", HELP\_MESSAGE, HELP\_MESSAGE);

},

"Unhandled": function() {

this.handler.state = states.START;

this.emitWithState("Start");

}

};

var startHandlers = Alexa.CreateStateHandler(states.START,{

"Start": function() {

this.emit(":ask", WELCOME\_MESSAGE, HELP\_MESSAGE);

},

"AnswerIntent": function() {

var item = getItem(this.event.request.intent.slots);

if (item[Object.getOwnPropertyNames(data[0])[0]] != undefined)

{

if (USE\_CARDS\_FLAG)

{

var imageObj = {smallImageUrl: getSmallImage(item), largeImageUrl: getLargeImage(item)};

this.emit(":askWithCard", getSpeechDescription(item), REPROMPT\_SPEECH, getCardTitle(item), getTextDescription(item), imageObj);

}

else

{

this.emit(":ask", getSpeechDescription(item), REPROMPT\_SPEECH);

}

}

else

{

this.emit(":ask", getBadAnswer(item), getBadAnswer(item));

}

},

"QuizIntent": function() {

this.handler.state = states.QUIZ;

this.emitWithState("Quiz");

},

"AMAZON.StopIntent": function() {

this.emit(":tell", EXIT\_SKILL\_MESSAGE);

},

"AMAZON.CancelIntent": function() {

this.emit(":tell", EXIT\_SKILL\_MESSAGE);

},

"AMAZON.HelpIntent": function() {

this.emit(":ask", HELP\_MESSAGE, HELP\_MESSAGE);

},

"Unhandled": function() {

this.emitWithState("Start");

}

});

var quizHandlers = Alexa.CreateStateHandler(states.QUIZ,{

"Quiz": function() {

this.attributes["response"] = "";

this.attributes["counter"] = 0;

this.attributes["quizscore"] = 0;

this.emitWithState("AskQuestion");

},

"AskQuestion": function() {

if (this.attributes["counter"] == 0)

{

this.attributes["response"] = START\_QUIZ\_MESSAGE + " ";

}

var random = getRandom(0, data.length-1);

var item = data[random];

var propertyArray = Object.getOwnPropertyNames(item);

var property = propertyArray[getRandom(1, propertyArray.length-1)];

this.attributes["quizitem"] = item;

this.attributes["quizproperty"] = property;

this.attributes["counter"]++;

var question = getQuestion(this.attributes["counter"], property, item);

var speech = this.attributes["response"] + question;

this.emit(":ask", speech, question);

},

"AnswerIntent": function() {

var response = "";

var item = this.attributes["quizitem"];

var property = this.attributes["quizproperty"]

var correct = compareSlots(this.event.request.intent.slots, item[property]);

if (correct)

{

response = getSpeechCon(true);

this.attributes["quizscore"]++;

}

else

{

response = getSpeechCon(false);

}

response += getAnswer(property, item);

if (this.attributes["counter"] < 10)

{

response += getCurrentScore(this.attributes["quizscore"], this.attributes["counter"]);

this.attributes["response"] = response;

this.emitWithState("AskQuestion");

}

else

{

response += getFinalScore(this.attributes["quizscore"], this.attributes["counter"]);

this.emit(":tell", response + " " + EXIT\_SKILL\_MESSAGE);

}

},

"AMAZON.StartOverIntent": function() {

this.emitWithState("Quiz");

},

"AMAZON.StopIntent": function() {

this.emit(":tell", EXIT\_SKILL\_MESSAGE);

},

"AMAZON.CancelIntent": function() {

this.emit(":tell", EXIT\_SKILL\_MESSAGE);

},

"AMAZON.HelpIntent": function() {

this.emit(":ask", HELP\_MESSAGE, HELP\_MESSAGE);

},

"Unhandled": function() {

this.emitWithState("AnswerIntent");

}

});

function compareSlots(slots, value)

{

for (var slot in slots)

{

if (slots[slot].value != undefined)

{

if (slots[slot].value.toString().toLowerCase() == value.toString().toLowerCase())

{

return true;

}

}

}

return false;

}

function getRandom(min, max)

{

return Math.floor(Math.random() \* (max-min+1)+min);

}

function getRandomSymbolSpeech(symbol)

{

return "<say-as interpret-as='spell-out'>" + symbol + "</say-as>";

}

function getItem(slots)

{

var propertyArray = Object.getOwnPropertyNames(data[0]);

var value;

for (var slot in slots)

{

if (slots[slot].value !== undefined)

{

value = slots[slot].value;

for (var property in propertyArray)

{

var item = data.filter(x => x[propertyArray[property]].toString().toLowerCase() === slots[slot].value.toString().toLowerCase());

if (item.length > 0)

{

return item[0];

}

}

}

}

return value;

}

function getSpeechCon(type)

{

var speechCon = "";

if (type) return "<say-as interpret-as='interjection'>" + speechConsCorrect[getRandom(0, speechConsCorrect.length-1)] + "! </say-as><break strength='strong'/>";

else return "<say-as interpret-as='interjection'>" + speechConsWrong[getRandom(0, speechConsWrong.length-1)] + " </say-as><break strength='strong'/>";

}

function formatCasing(key)

{

key = key.split(/(?=[A-Z])/).join(" ");

return key;

}

function getTextDescription(item)

{

var text = "";

for (var key in item)

{

text += formatCasing(key) + ": " + item[key] + "\n";

}

return text;

}

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

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

alexa.appId = APP\_ID;

alexa.registerHandlers(handlers, startHandlers, quizHandlers);

alexa.execute();

};