```
'use strict';

exports.handler = function (event, context) {
    try {
        console.log("event.session.application.applicationId=" +
event.session.application.applicationId);


        if (event.session.new) {
            onSessionStarted({requestId: event.request.requestId}, event.session);
        }

        if (event.request.type === "LaunchRequest") {
            onLaunch(event.request,
                event.session,
                function callback(sessionAttributes, speechletResponse) {
                    context.succeed(buildResponse(sessionAttributes, speechletResponse));
                });
        } else if (event.request.type === "IntentRequest") {
            onIntent(event.request,
                event.session,
                function callback(sessionAttributes, speechletResponse) {
                    context.succeed(buildResponse(sessionAttributes, speechletResponse));
                });
        } else if (event.request.type === "SessionEndedRequest") {
            onSessionEnded(event.request, event.session);
            context.succeed();
        }
    } catch (e) {
        context.fail("Exception: " + e);
    }
};

/**
 * Called when the session starts.
 */
function onSessionStarted(sessionStartedRequest, session) {
    console.log("onSessionStarted requestId=" + sessionStartedRequest.requestId
        + ", sessionId=" + session.sessionId);
}

/**
 * Called when the user invokes the skill without specifying what they want.
 */
```

```
function onLaunch(launchRequest, session, callback) {
   console.log("onLaunch requestId=" + launchRequest.requestId
      + ", sessionId=" + session.sessionId);

   var cardTitle = "You have not specified any intent !"
   var speechOutput = "You have not specified any intent!"
   console.log("No intent specified at invokation");
   callback(session.attributes,
      buildSpeechletResponse(cardTitle, speechOutput, "", true));
}

/**
 * Called when the user specifies an intent for this skill.
 */
function onIntent(intentRequest, session, callback) {
   console.log("onIntent requestId=" + intentRequest.requestId
      + ", sessionId=" + session.sessionId);

   var intent = intentRequest.intent,
      intentName = intentRequest.intent.name;

   // dispatch custom intents to handlers here
   if (intentName == 'start') {
      console.log("intent = ", intentName);
      handleStartRequest(intent, session, callback);
   }
   else {
      console.log("invalid intent = ", intentName);
      throw "Invalid intent";
   }
}

/**
 * Called when the user ends the session.
 * Is not called when the skill returns shouldEndSession=true.
 */
function onSessionEnded(sessionEndedRequest, session) {
   console.log("onSessionEnded requestId=" + sessionEndedRequest.requestId
      + ", sessionId=" + session.sessionId);

}

function handleStartRequest(intent, session, callback) {
   callback(session.attributes,
```

```
    buildSpeechletResponseWithoutCard("Hello, you need to write the code using Zoom Room API for me to start a meeting", "", "true"));
}

// ------- Helper functions to build responses -------

function buildSpeechletResponse(title, output, repromptText, shouldEndSession) {
  return {
    outputSpeech: {
      type: "PlainText",
      text: output
    },
    card: {
      type: "Simple",
      title: title,
      content: output
    },
    reprompt: {
      outputSpeech: {
        type: "PlainText",
        text: repromptText
      }
    },
    shouldEndSession: shouldEndSession
  };
}

function buildSpeechletResponseWithoutCard(output, repromptText, shouldEndSession) {
  return {
    outputSpeech: {
      type: "PlainText",
      text: output
    },
    reprompt: {
      outputSpeech: {
        type: "PlainText",
        text: repromptText
      }
    },
    shouldEndSession: shouldEndSession
  };
}

function buildResponse(sessionAttributes, speechletResponse) {
```

```
    return {
        version: "1.0",
        sessionAttributes: sessionAttributes,
        response: speechletResponse
    };
}
```