

```

1  const Alexa = require('ask-sdk-core');
2  var https = require('https');
3
4  const PlayHandler = {
5      canHandle(handlerInput)
6      {
7          return (
8              handlerInput.requestEnvelope.request.type === 'LaunchRequest' ||
9              (
10                 handlerInput.requestEnvelope.request.type === 'IntentRequest' &&
11                 handlerInput.requestEnvelope.request.intent.name === 'Play'
12             ) ||
13             (
14                 handlerInput.requestEnvelope.request.type === 'IntentRequest' &&
15                 handlerInput.requestEnvelope.request.intent.name ===
16                 'AMAZON.ResumeIntent'
17             )
18         );
19     },
20     handle(handlerInput)
21     {
22         return handlerInput.responseBuilder
23             .addDirective({
24                 type: 'AudioPlayer.Play',
25                 playBehavior: 'REPLACE_ALL',
26                 audioItem:{
27                     stream:{
28                         token: '0',
29                         url: 'https://radiovegit.radioca.st/radiovegit_aac',
30                         offsetInMilliseconds: 0
31                     }
32                 }
33             })
34             .getResponse();
35     };
36
37     const PauseStopHandler = {
38         canHandle(handlerInput)
39         {
40             return (
41                 handlerInput.requestEnvelope.request.type === 'IntentRequest' &&
42                 (
43                     handlerInput.requestEnvelope.request.intent.name ===
44                     'AMAZON.CancelIntent' ||
45                     handlerInput.requestEnvelope.request.intent.name ===
46                     'AMAZON.StopIntent'
47                 ) ||
48                 (
49                     handlerInput.requestEnvelope.request.type === 'IntentRequest' &&
50                     handlerInput.requestEnvelope.request.intent.name ===
51                     'AMAZON.PauseIntent'
52                 )
53             );
54         },
55         handle(handlerInput)
56         {
57             return handlerInput.responseBuilder
58                 .addDirective({
59                     type: 'AudioPlayer.ClearQueue',
60                     clearBehavior: 'CLEAR_ALL'
61                 })
62                 .getResponse();
63         }
64     };
65
66     const HelpIntentHandler = {
67         canHandle(handlerInput)
68         {
69             return handlerInput.requestEnvelope.request.type === 'IntentRequest' &&
70                 handlerInput.requestEnvelope.request.intent.name === 'AMAZON.HelpIntent';
71         },
72         handle(handlerInput)

```

```

70     {
71         const speechText = 'Per maggiori informazioni visita il sito www punto
radioveg punto it';
72         return handlerInput.responseBuilder
73             .speak(speechText)
74             .getResponse();
75     }
76 };
77
78 const SessionEndedRequestHandler = {
79     canHandle(handlerInput)
80     {
81         return handlerInput.requestEnvelope.request.type === 'SessionEndedRequest';
82     },
83     handle(handlerInput)
84     {
85         return handlerInput.responseBuilder.getResponse();
86     }
87 };
88
89 const IntentReflectorHandler = {
90     canHandle(handlerInput)
91     {
92         return handlerInput.requestEnvelope.request.type === 'IntentRequest';
93     },
94     handle(handlerInput)
95     {
96         const intentName = handlerInput.requestEnvelope.request.intent.name;
97         const speechText = 'Per maggiori informazioni visita il sito www punto
radioveg punto it';
98         return handlerInput.responseBuilder
99             .speak(speechText)
100             .getResponse();
101     }
102 };
103
104 const ErrorHandler = {
105     canHandle()
106     {
107         return true;
108     },
109     handle(handlerInput, error)
110     {
111         const speechText = `Non ho capito quello che hai detto, puoi ripetere per
favore?`;
112         return handlerInput.responseBuilder
113             .speak(speechText)
114             .reprompt(speechText)
115             .getResponse();
116     }
117 };
118
119 exports.handler = Alexa.SkillBuilders.custom()
120     .addRequestHandlers(
121         PlayHandler,
122         PauseStopHandler,
123         HelpIntentHandler,
124         SessionEndedRequestHandler,
125         IntentReflectorHandler)
126     .addErrorHandlers(
127         ErrorHandler)
128     .lambda();

```