Getting Started with EchoToEG for Amazon Alexa

Overview

EchoToEG allows you to trigger events in EventGhost using voice commands through Amazon Alexa.

There are 3 ways to interact which are configured in Alexa skills.

Option 1 – Invocation, intent and action:-

- (Invocation, Intent, Action)
  - Example 1: Alexa, "Event ghost", "perform action", "lights on"
  - Example 2: Alexa, "ask Event ghost to perform action lights on"
  - Example 3: Alexa, "Event ghost", "Dim lights", "in dining room"
- Benefits:
  - The actions can be configured with all sorts of pre-configured structure using the Alexa toolkit. This is useful if you want content like numbers, dates, weekdays etc...
  - Can be configured with lots of commands without spending a lot of time configuring Alexa skills.
  - The example is configured with Action slot defined as a search query, this allows you to pass any command to EG to trigger events (i,e Alexa, "Event ghost", "perform action", whatever you say)
- Drawback:
  - Lots of words to initiate actions

Option 2 - Invocation and intent:-

- Shorter (Invocation, Intent)
  - Example 1: Alexa, "Event ghost", " play music"
  - Example 2: Alexa, "ask Event ghost to play music"
- Benefits:
  - Requires a little effort to configure
  - Provides restricted options to ensure Alexa understands the command.
  - Can be configured with multiple utterances (word variances)
- Drawbacks:
  - Not as short as it can be

Option 3 - Invocation only:-

- Shortest (Invocation)
  - Example 1: Alexa, "initiate music"
- Benefits:
  - Few words required
- Drawbacks:
  - Most effort to configure, requires an Alexa skill per command, linked by the skillID to the EG python code.
  - You have to stay clear of Alexa's trigger commands (and there are loads which you will stumble over lots! For example "Play" or "start" would have been the obvious choice rather than "initiate music", if they weren't restricted!)

Table of Contents

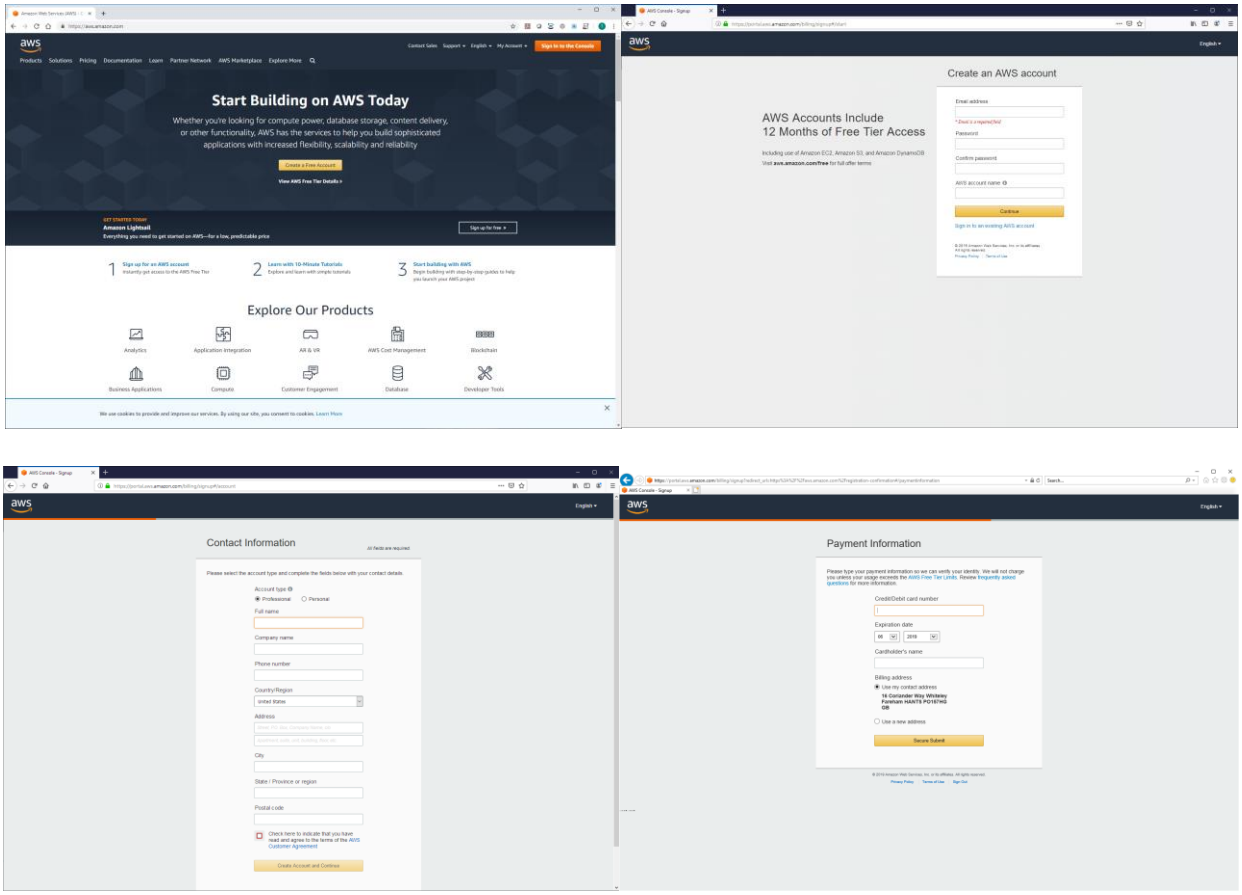
Setup Tutorial

Section 1 (Create your AWS Lambda function for your skill to use)

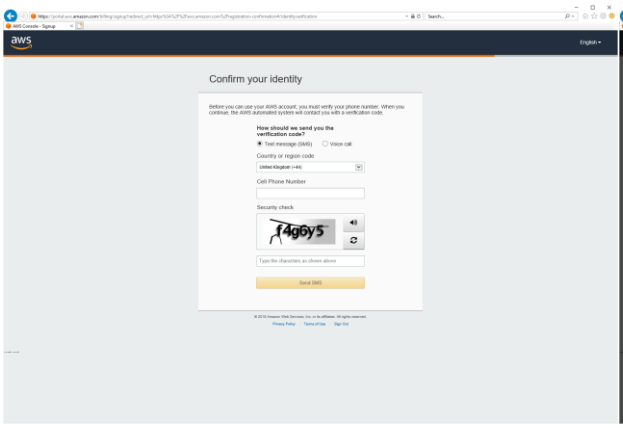
1. Download or clone the EchoToEventGhost github project https://github.com/nexgenclassic/EchoToEventGhost
2. Make sure you have your EventGhost Webserver is running and you can post to from outside your network.
    - Get Your External IP, https://whatismyipaddress.com/
    - Plus the post that your have configured for the EventGhost webserver to run on (80 is default). If you need help with configuring port forwarding look here http://www.wikihow.com/Set-Up-Port-Forwarding-on-a-Router
    - To test go to http://ip:port/page?event ie. http://192.168.1.1:80/index.html?EG_Event and an event should be logged in EventGhost

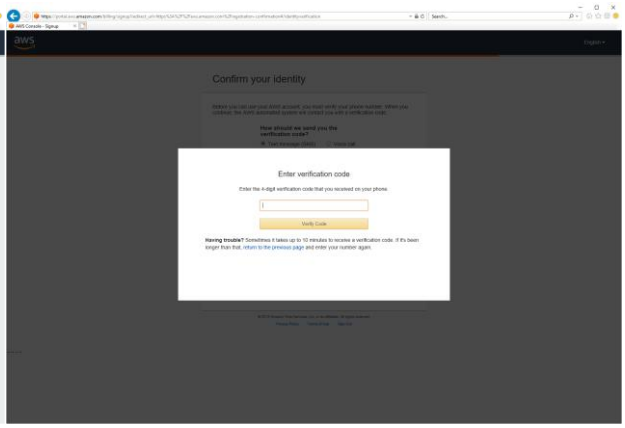
Note: If you don't have a static external IP address it is recommended you setup a domain name and dynamic DNS. https://www.duckdns.org/install.jsp is a good starting point.

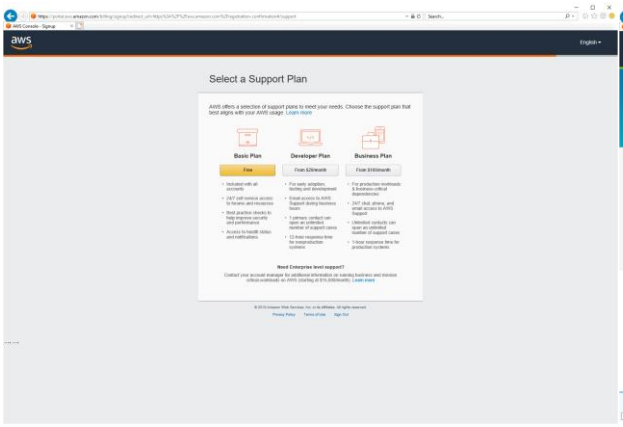
3. If you do not already have an account on AWS, go to Amazon Web Services (https://aws.amazon.com/) and create an account.

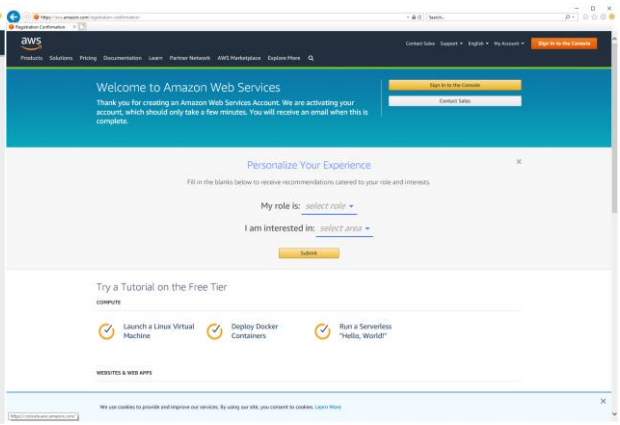


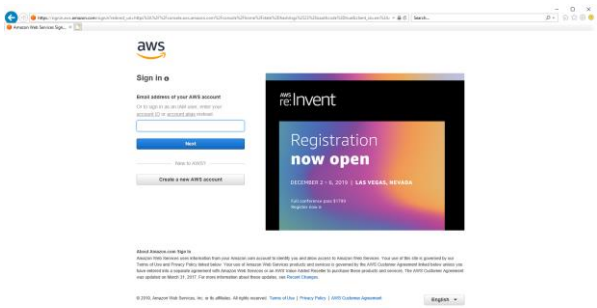
Steps may vary, however images show process in June '19









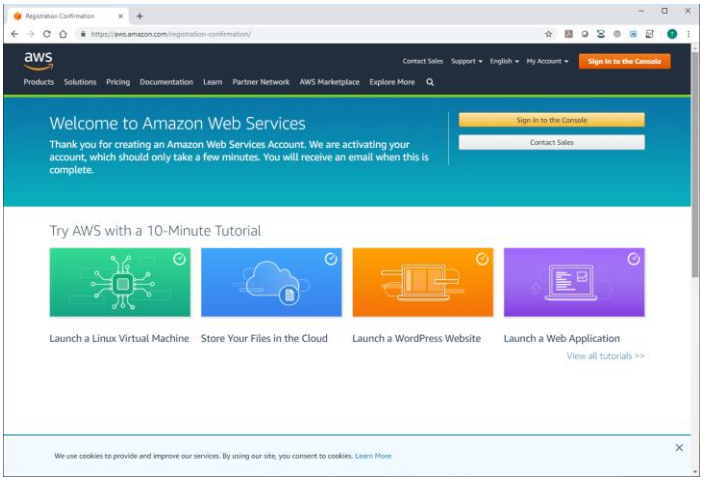


Steps may vary, however images show process in June ‘19

Note: You will need to enter your credit card details. You may see a <$1 transaction on your card to verify payment, however this will not be debited.

Note: You get a significant amount of use free every month, you will not exceed this with your personal Alexa/EG usage alone.

4. Log in to the AWS Management Console.



5. Click the region drop-down in the upper-right corner of the console and select either **US East (N. Virginia)** or **EU (Ireland)**. Lambda functions for Alexa skills must be hosted in either the **US East (N. Virginia)** or **EU (Ireland)** region.

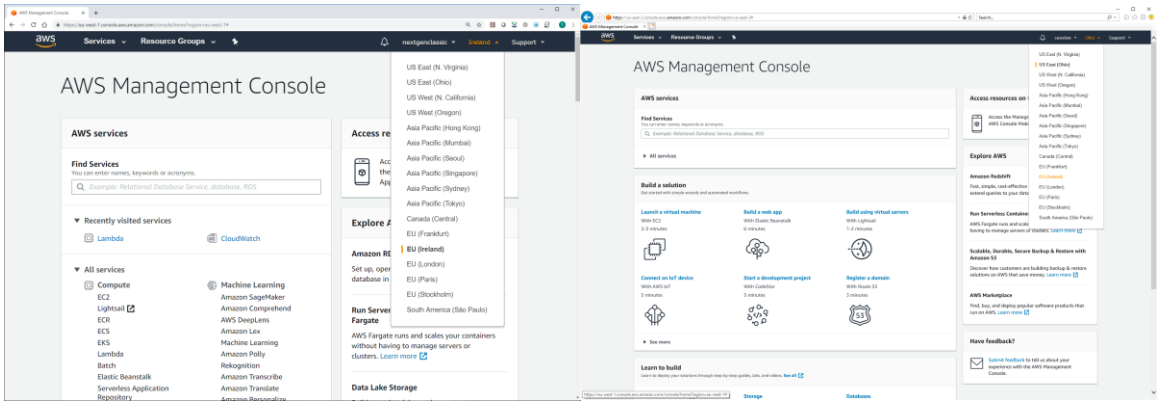
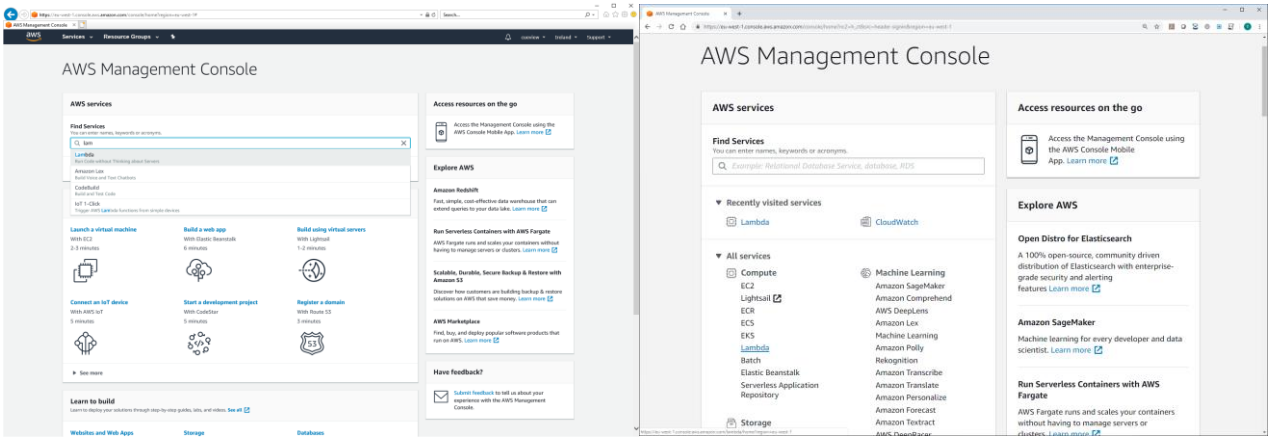


Image depends on whether you are an existing or new user

6. Navigate to **All services → Lambda**.

Note: The search facility is an easy way to navigate.



7. Click **Create a Function**.

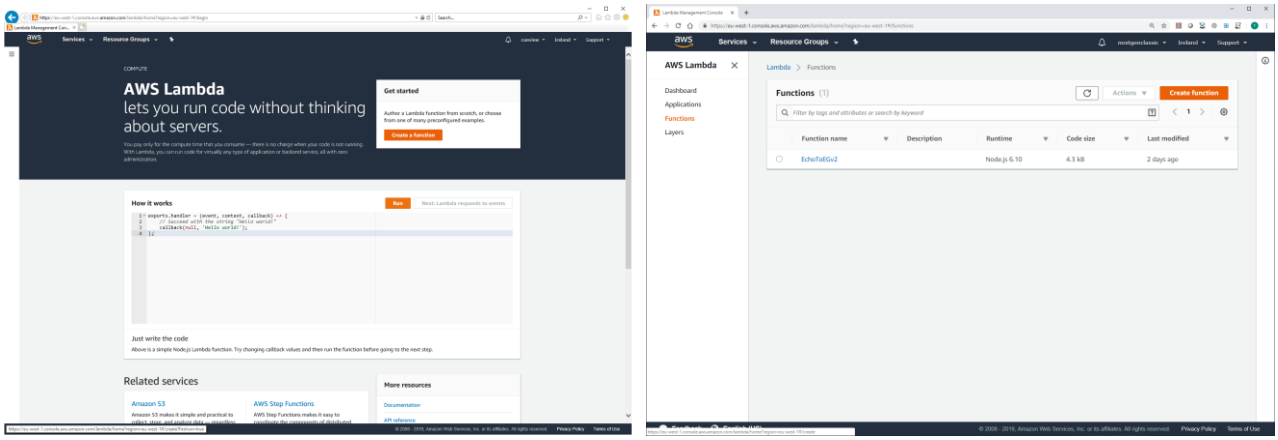
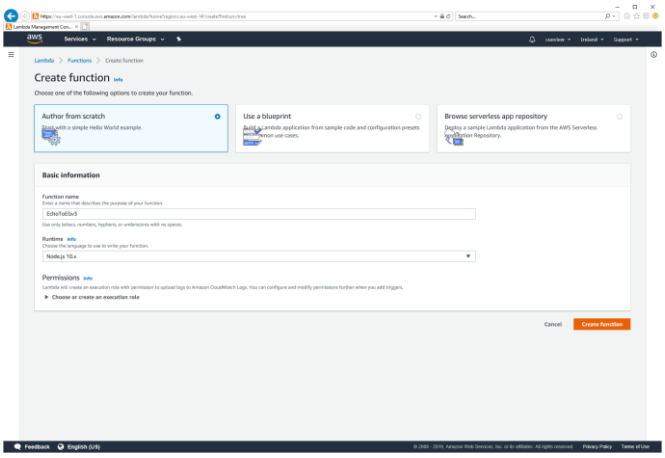


Image depends on whether you have existing functions

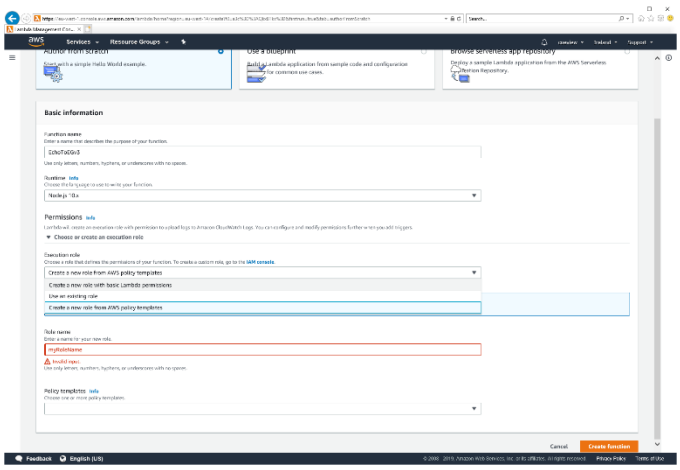
8. Select **Author from scratch**.



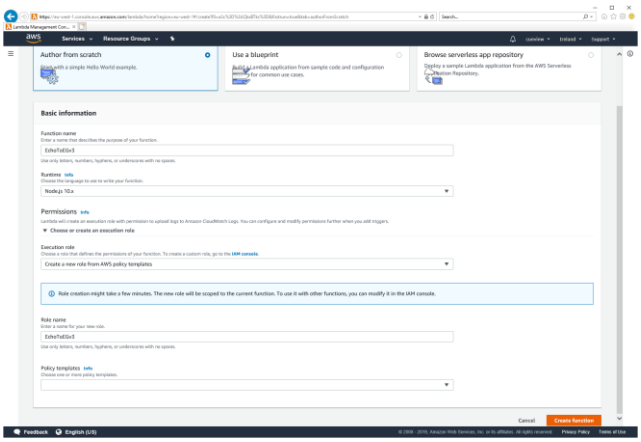
9. For the function name enter **EchoToEGv3**, the Description can be left blank, and for the runtime select **Node.js 6.10**

Note: If Node.js 6.10 is not available select a later version. You will be presented with more options in a later step.

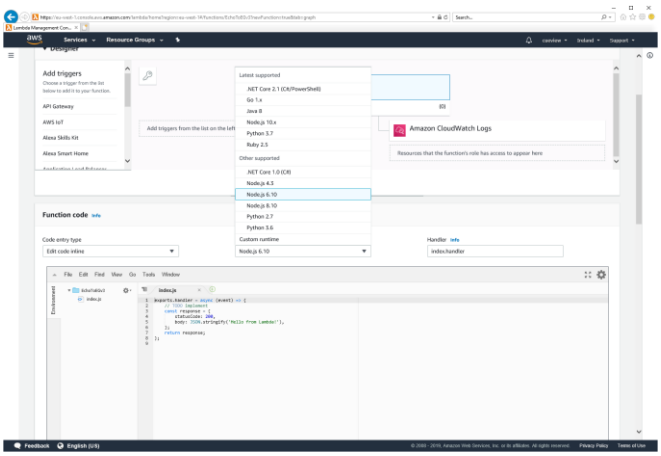
10. Select **Create a new role with basic Lambda permissions** from the drop down.



11. Enter **EchoToEGv3** in the Role name text field
12. Click **Create function**.

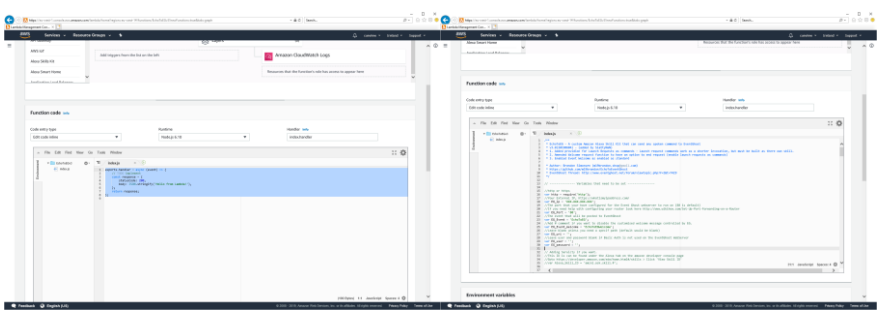


13. If not already set, change the Runtime to **Node.js 6.10**.

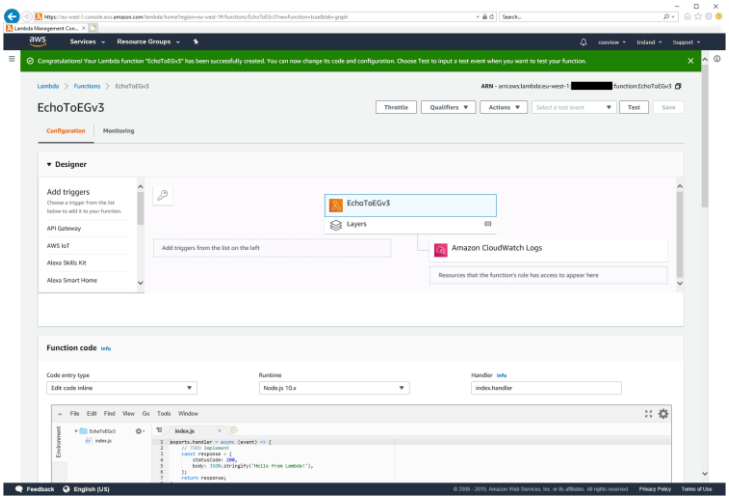


14. Under the Lambda **function code** section leave as **Edit code inline.**
15. Delete the existing code and then **copy in the code** from \EchoToEventGhost\AlexaSkillKit_Code\EchoToEGv2.js
16. Edit the variables in the code:-
    - Line 17: Is your EventGhost Webserver using **http** and https
    - Line 19: **Enter your External IP** or domain name
    - Line 22: Enter the **port number** you configured in the EventGhost Webserver
    - Lines 30 & 31 are your EventGhost Webserver user name and password

Note: There are other variables that can be set for added security or personalisation. It is recommended you get the function working before exploring these.

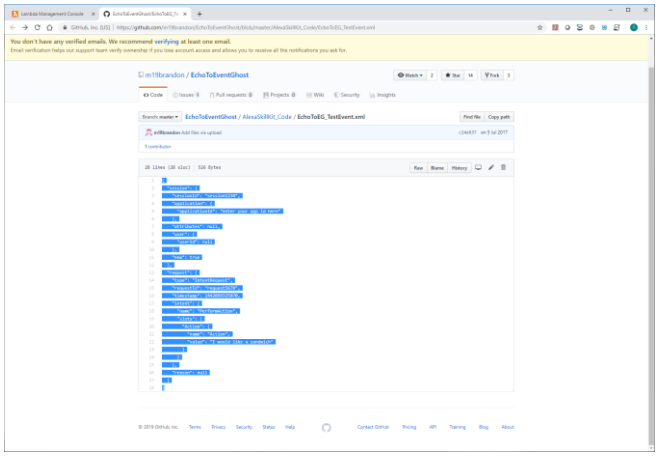


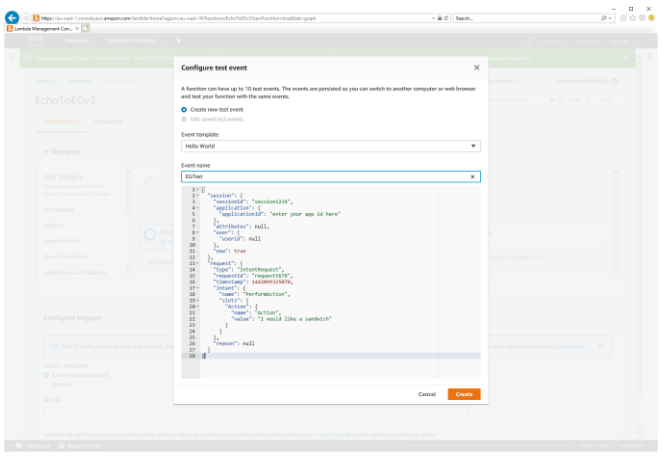
17. Click **Save**.



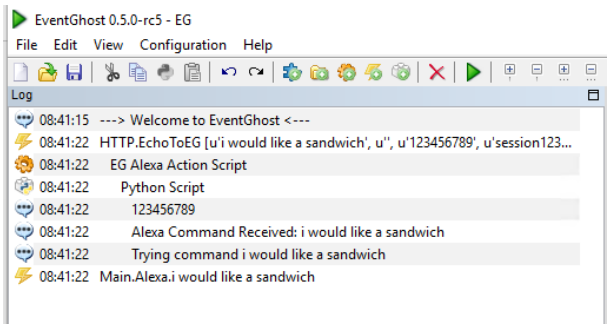
18. You can test your function by using the test function.
    - Click the **Select a test event** dropdown and then select **configure test events**.
    - Change the Event name to **EGTest** or your preference
    - Paste the contents of \EchoToEventGhost\AlexaSkillKit_Code\EchoToEG_TestEvent.xml in the code section.

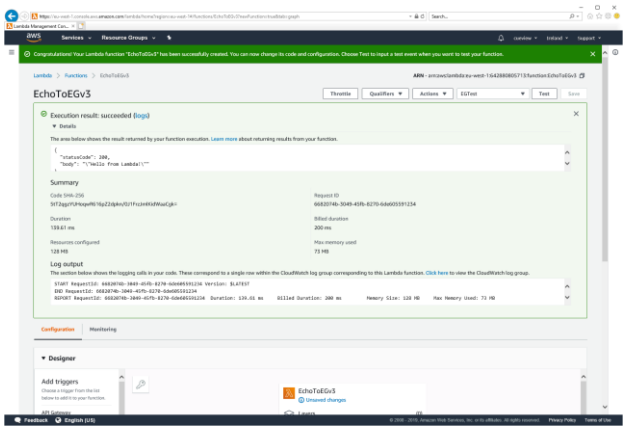
    Note: Line 22 is the command that is passed, change it as you desire.



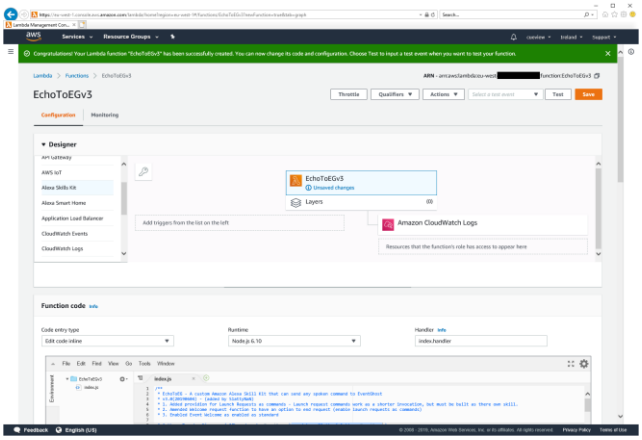


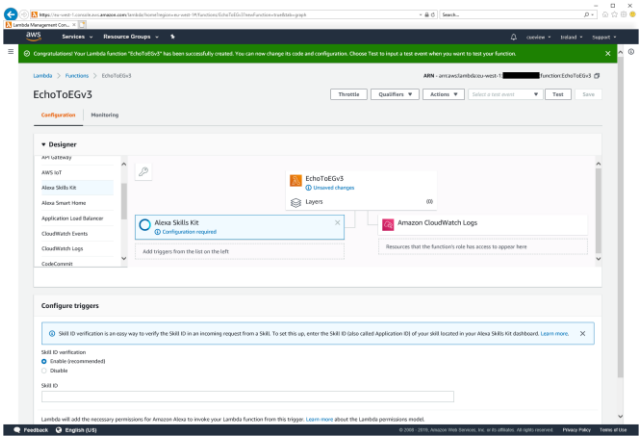
19. Click **Create**
20. Click **Save** and then **Test,** if all works you will see an event in your EventGhost log and an **Execution result: succeeded** message. Click the Details drop down if you wish to see the detail.



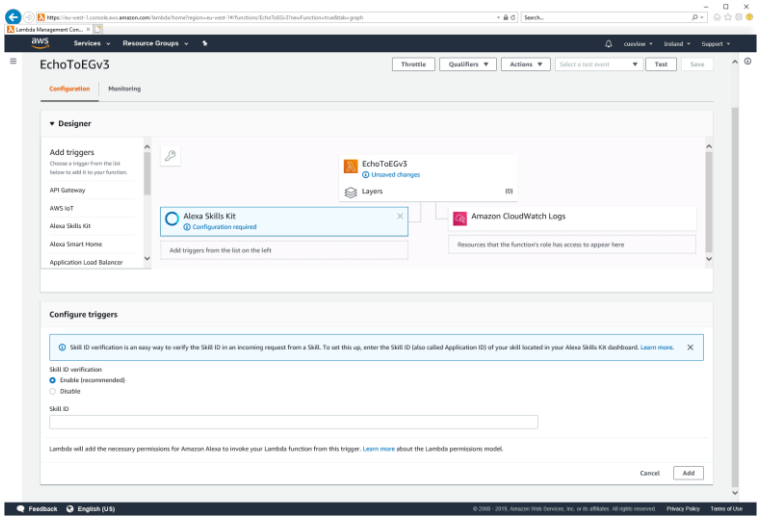


21. Click on the **Designer** drop down and select **Alexa Skills Kit.**





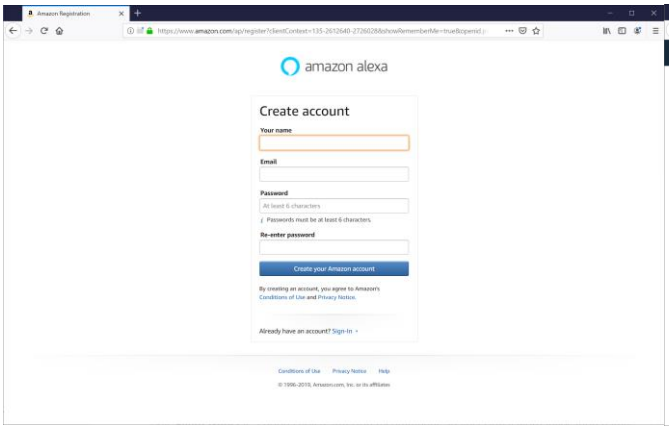
22. The configure triggers section will appear below. You will need to link this to the Alexa skills you make in later steps. For now leave this webpage open and continue the tutorial in another browser tab.

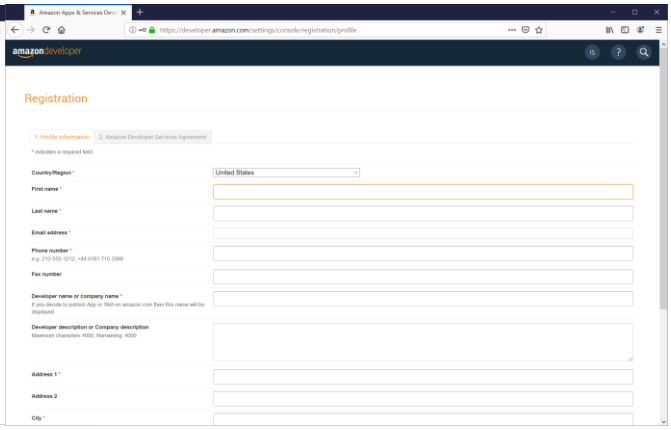


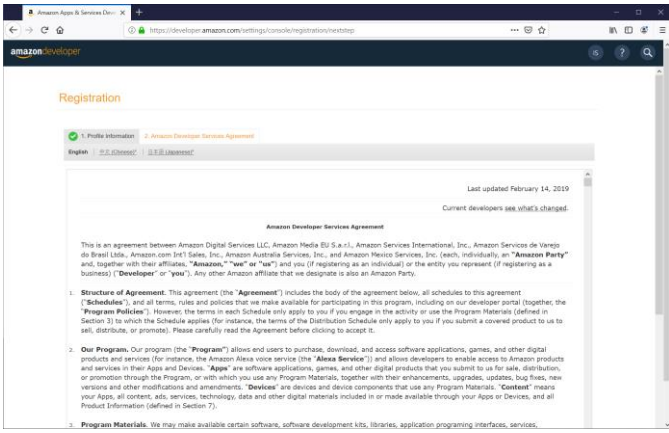
Section 2 (Create your Alexa Skill and link your Lambda function)

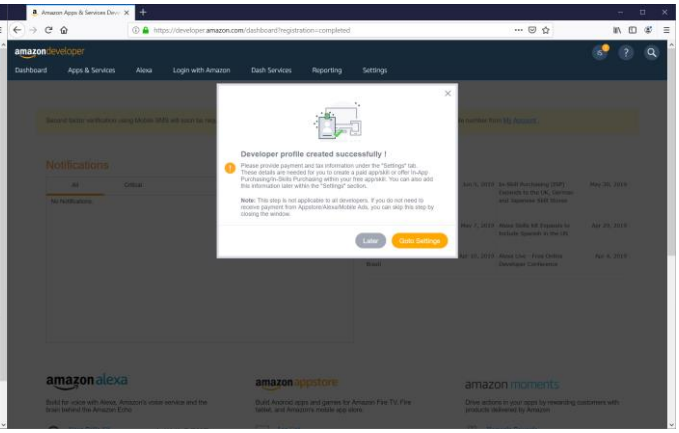
23. Sign in to the **Amazon developer portal** (https://developer.amazon.com/alexa/console/ask). If you haven't done so already, you'll need to create a free account. https://developer.amazon.com/edw/home.html#/

NOTE: You must use the same account linked to your Alexa, otherwise the skill you create will not linked to your Alexa device.

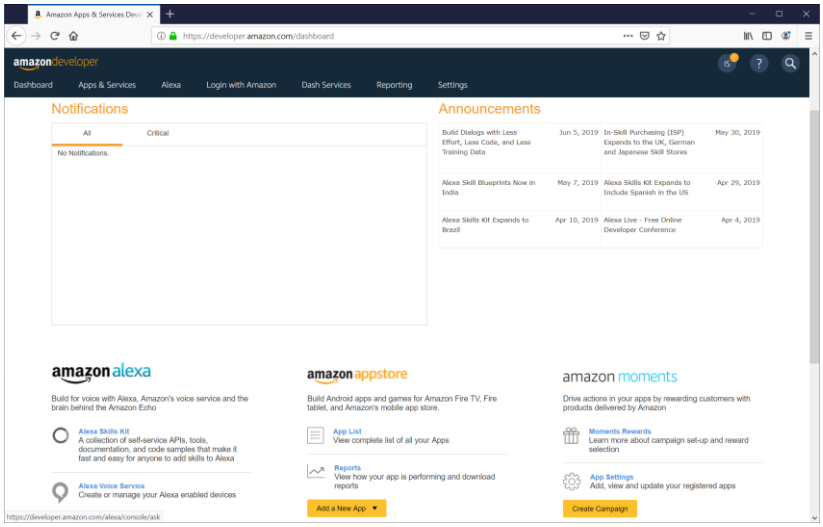




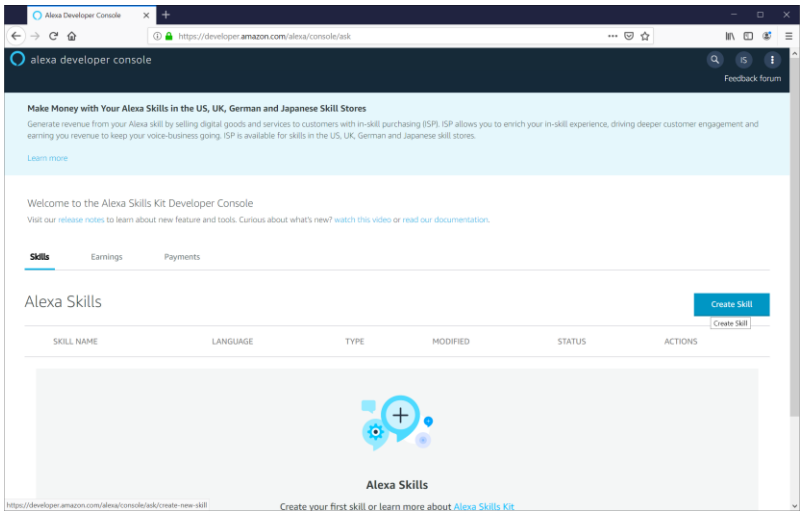




Note: You do not need to enter payment details.



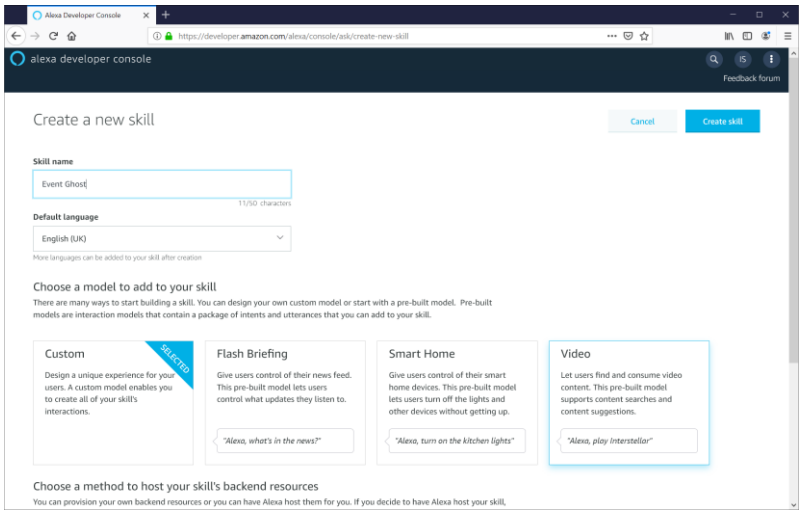
24. Click on **Alexa Skills Kit** under the amazon alexa section.



25. Click **Create Skill**.

26. Name your skill. This is the name displayed to users in the Alexa app. **Event Ghost** is a good choice.

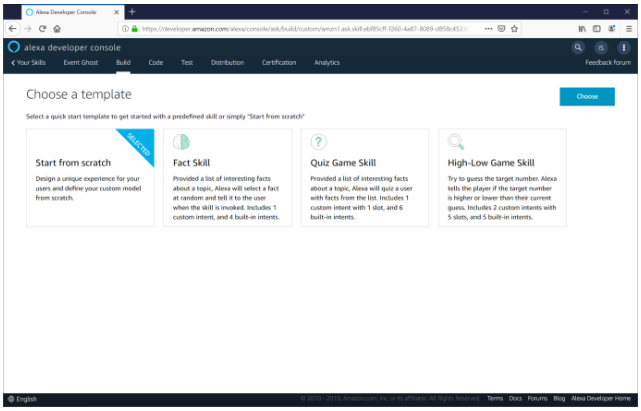
Note: This skill can be configured to do multiple commands for both option 1 and option 2 configurations.

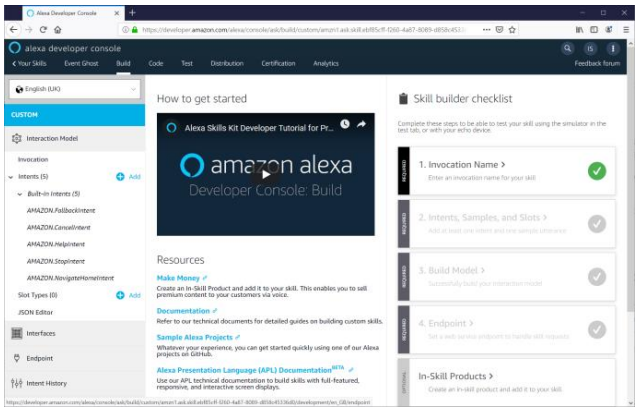


27. Leave **Custom** and **provision your own** options selected.

28. Click **create skill**.

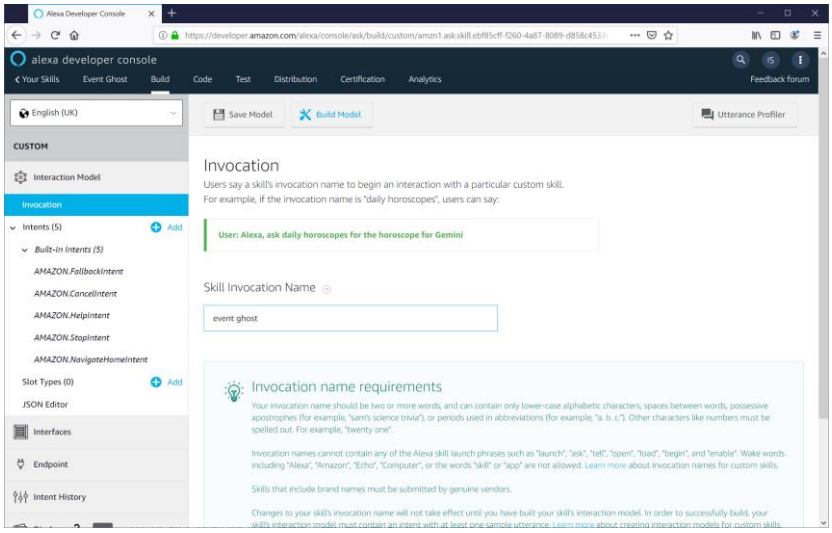
29. Leave from scratch selected, click **choose**



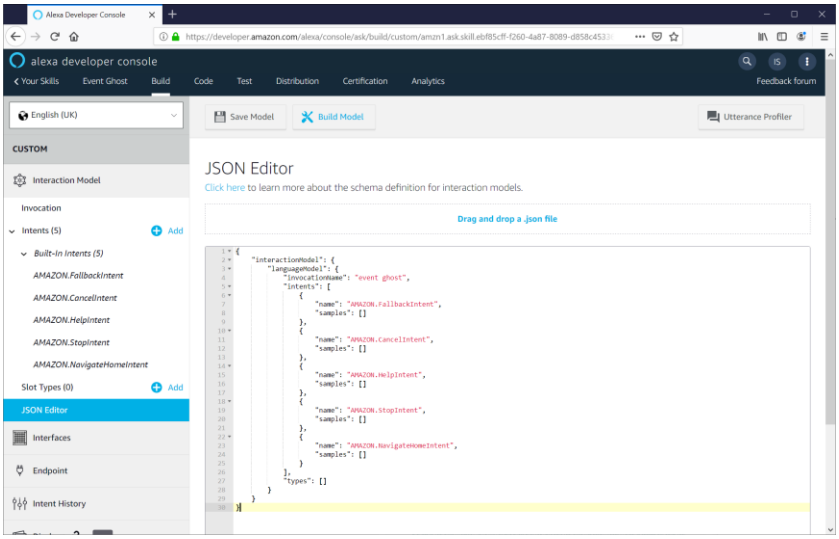


30. Select **Invocation** in the left side column.

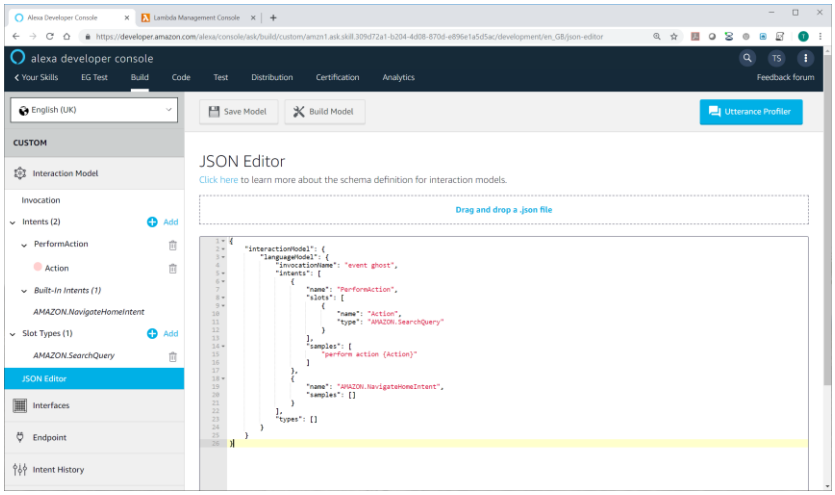
31. Create an invocation name. This is the word or phrase that users will speak to activate the skill. **Event Ghost** is a good choice. Amazon recommends against signal word invocation name.



32. Click **JSON Editor** in the left side column.



33. In the code box, paste the JSON code from \EchoToEventGhost\AlexaSkillKit_Code\IntentSchema.txt



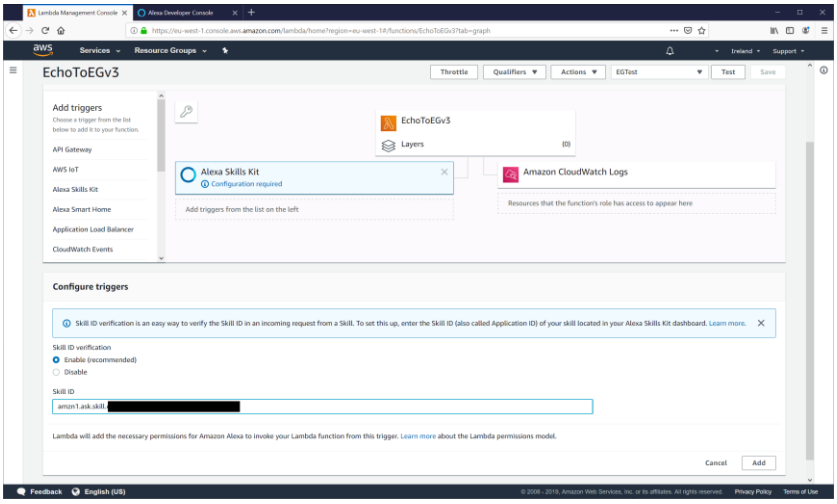
34. Click **Endpoint** in the left side column.

35. Select **AWS Lambda ARN**.

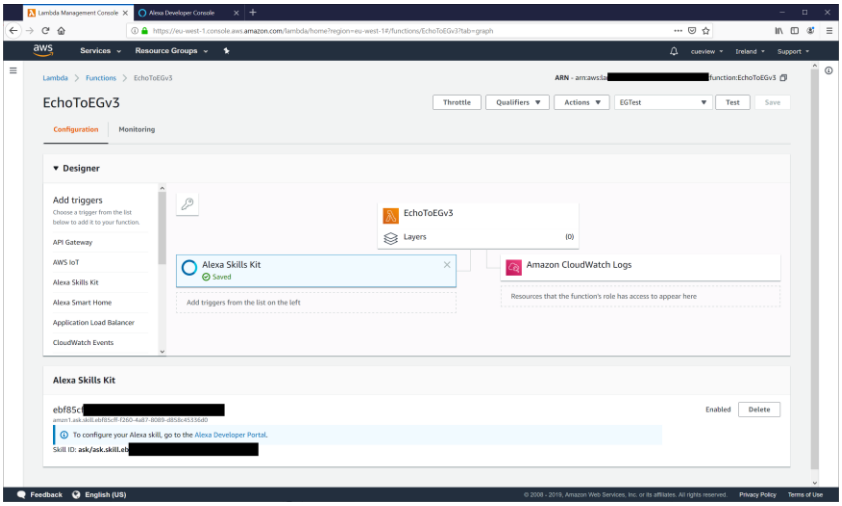
36. Click Copy to Clipboard to **copy your skill id**.

37. Go **back the AWS Console tab** as left in step 22.

38. Paste the skill ID into the **Skill ID box** and click **Add**.

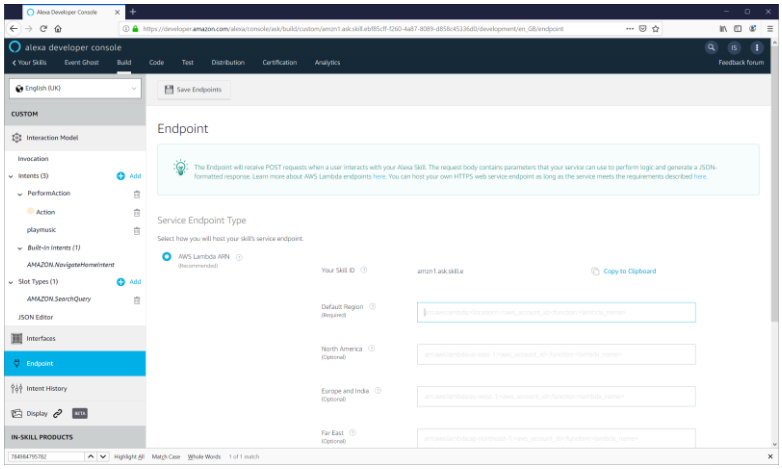


39. Click **Save**.



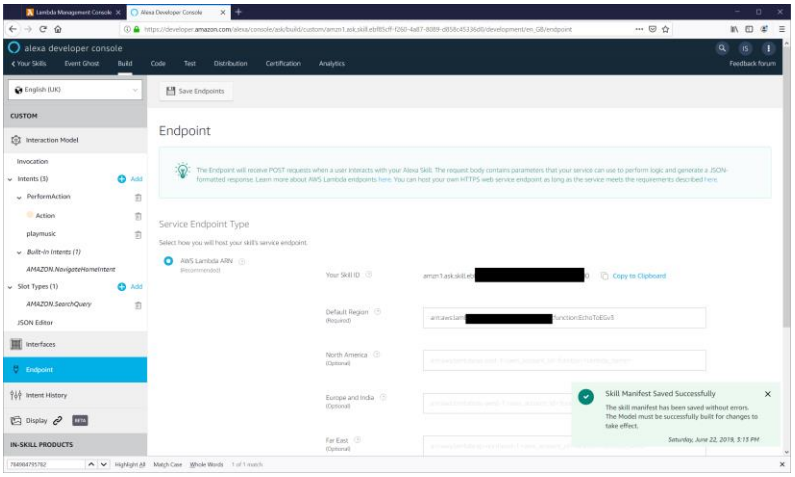
40. Click on the **copy** button to copy the **ARN address** to clipboard.

41. Return to the **Alexa skill tab**.



42. Paste your **ARN number** into the **Default Region**.

43. Click Save **Endpoints**.



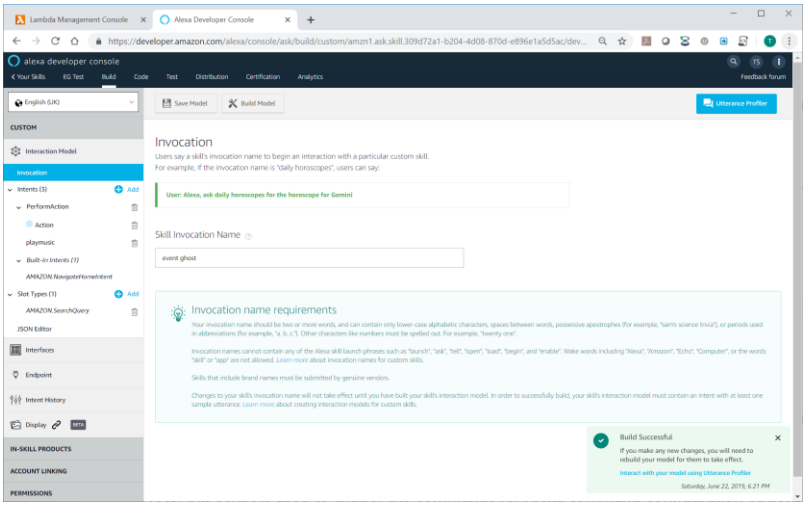
44. Check that the **Skill Manifest Saved Successfully** message appears.

45. Click **Innovation** in the left side column.

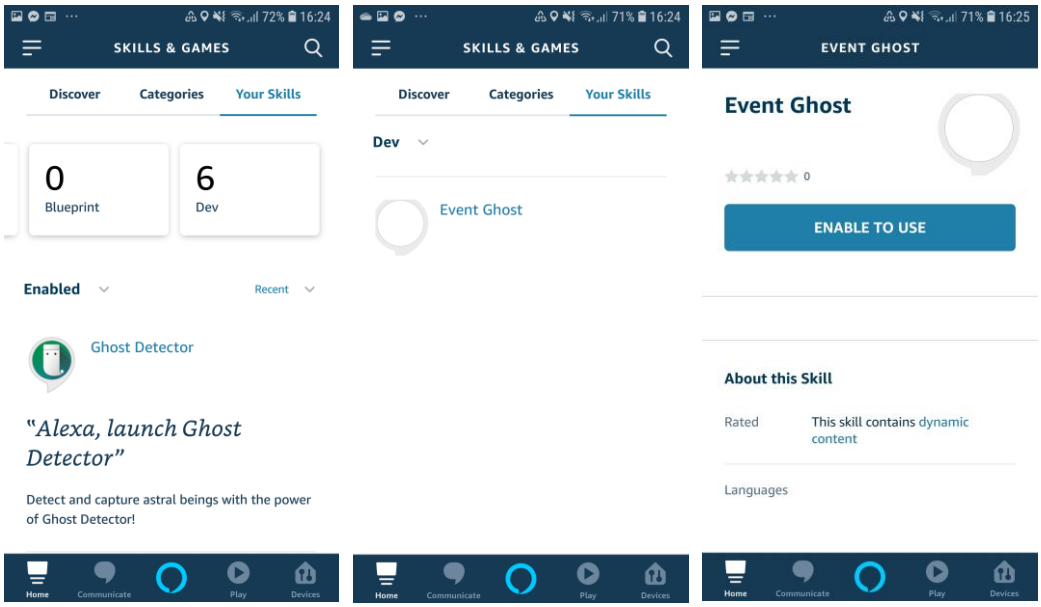
46. Click on **Save Model** and then **Build Model**.

47. A message box confirming the build was successful should appear.

Note: There is no need to publish the skill.



48. Open the **Alexa app** on your **mobile phone**.
    - Click on the burger menu on the top left and select **SKILLS & GAMES**.
    - Click on **Your Skills**.
    - Slide past the Enabled, Updated and other App groups to the Dev group appears and **select Dev**.
    - **Open the skill** (Event Ghost or whatever your name it)
    - Unless already enabled automatically, select **ENABLE TO USE**.
    - Close the app.



Section 3 (EventGhost, add some code to make stuff happen)

49. Open EventGhost

50. Create a new **Python Script Macro**

51. Copy and paste the contents of \EchoToEventGhost\EventGhost_Code\PythonWelcomeScript.py

52. **Rename** the macro to something suitable, example: **EG Alexa Welcome Script**

Note: this script is used to customise the welcome response from Alexa when one of your Alexa skills is activated and trigger commands setup for Alexa command configuration option 3.

Note: There is more information on setup and configuration in Section 4

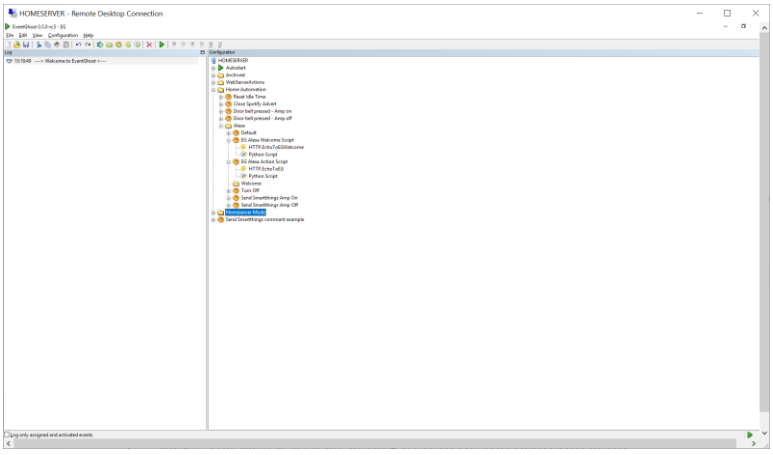
53. Create a new **Python Script Macro**

54. Copy and paste the contents of \EchoToEventGhost\EventGhost_Code\PythonActionScript.py

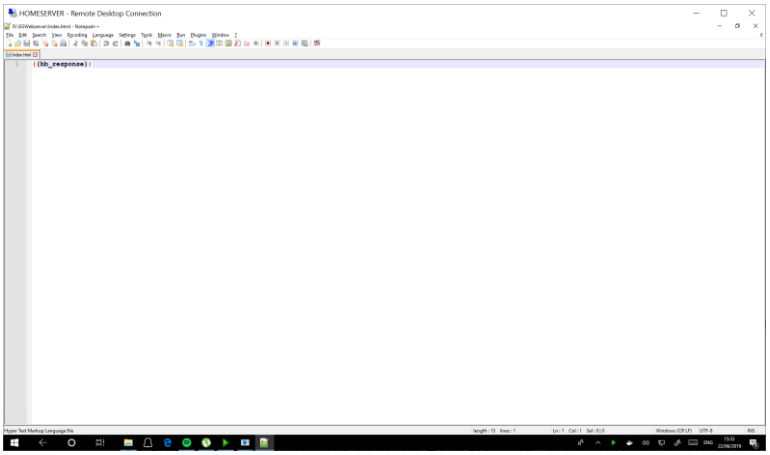
55. **Rename** the macro to something suitable, example: **EG Alexa Action Script**

Note: this script is used to trigger the commands for Alexa command configuration options 1 & 2.

Note: More information on setup and configuration in Section 4



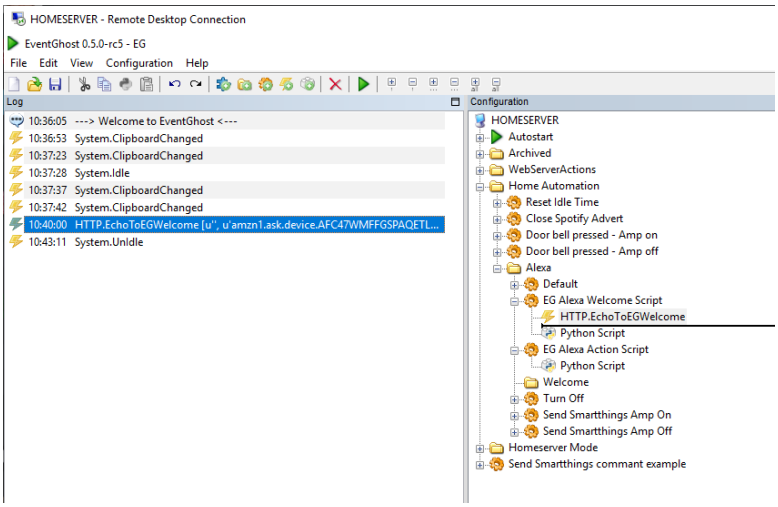
56. Copy \EchoToEventGhost\EventGhost_Code\index.html into your webserver folder



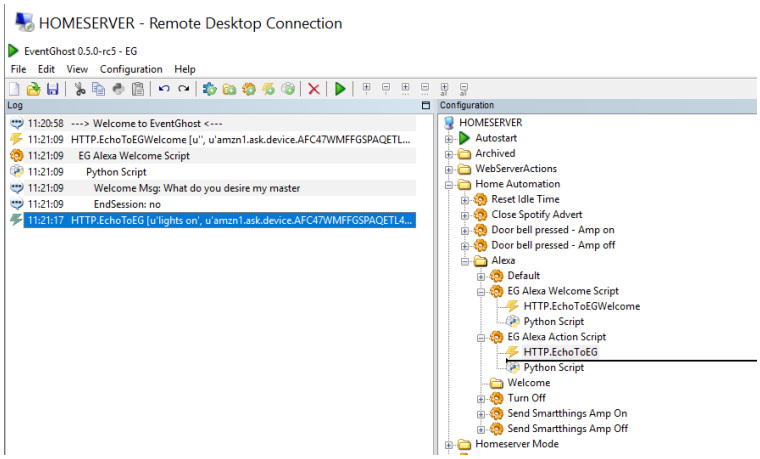
57. Now it's time to talk to Alexa! **Say: "Alexa, Event Ghost".**

Note: This is Alexa, INVOCATION_NAME, if you have already customised ahead of tutorial change Event Ghost to your chosen Invocation name.

58. Alexa should reply with "There is a problem with the requested skills response", don't panic, this is good news ☺.
59. A trigger event should have appeared in your EventGhost log named HTTP.EchoToEGWelcome[etc...]. drag the event to the welcome macro made in step 50.



60. Now retry the Alexa command "Alexa, Event Ghost".
    - This time you will get one of the preconfigured welcome responses asking you to confirm the command.
    - Respond with "perform action, test event".
    - Alexa will then respond "Got it, working on the command lights on", shortly after it will prompt, "I do not understand, please try again", this is normal, let the retry time out.
61. This time a trigger event should have appeared in your EventGhost log named HTTP.EchoToEG [etc...]. drag the event to the Action macro made in step 50.



62. Now you are ready to go! You have successfully configure EG and Alexa to work in Alexa command configuration option 1.

63. The following section talks you through how to setup commands in option modes 2 and 3, as well as some of the advanced options. I recommend you try a few Alexa commands to get an understanding of what's going on before proceeding:-

Examples:

- "Alexa, event ghost, perform action I like toast"
- "Alexa, Tell event ghost to perform action toast"

Note: Avoid "lights on", were are going set that up as an example of configuration 1 in section 4.

Section 4 Setting up your commands

Option 1 Configuration tutorial – Invocation, intent and action:-

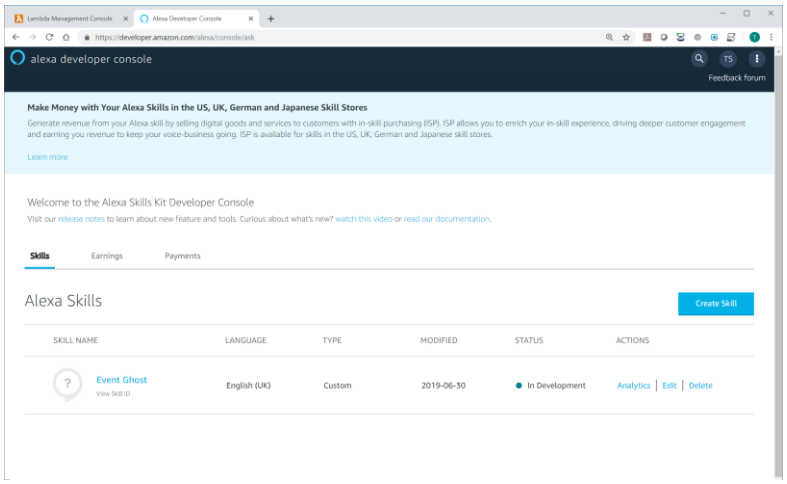
- (Invocation, Intent, Action)
  - Example 1: Alexa, "Event ghost", "perform action", "lights on"
  - Example 2: Alexa, "ask Event ghost to perform action lights on"
  - Example 3: Alexa, "Event ghost", "Dim lights", "in dining room"
- Benefits:
  - The actions can be configured with all sorts of pre-configured structure using the Alexa toolkit. This is useful if you want content like numbers, dates, weekdays etc...
  - Can be configured with lots of commands without spending a lot of time configuring Alexa skills.
  - The example is configured with Action slot defined as a search query, this allows you to pass any command to EG to trigger events (i,e Alexa, "Event ghost", "perform action", whatever you say)
- Drawback:
  - Lots of words to initiate actions

Good news, you have already successfully implemented an example of the configuration option 1 by getting to this stage of the tutorial. However let's step through the configuration again with a few additional configuration options to improve the experience.

The following steps are very similar to the steps completed previously, the main differences are:-

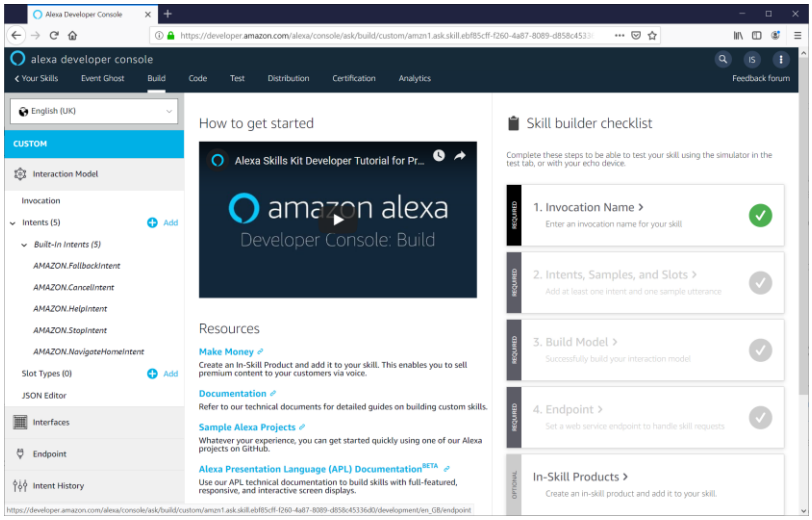
- You are going to set the skill name to something relevant to the command you are making
- Choosing your skill name will require more thought to avoid use of Alexa trigger words.

64. Open the alexa development console webpage https://developer.amazon.com/alexa/console/ask

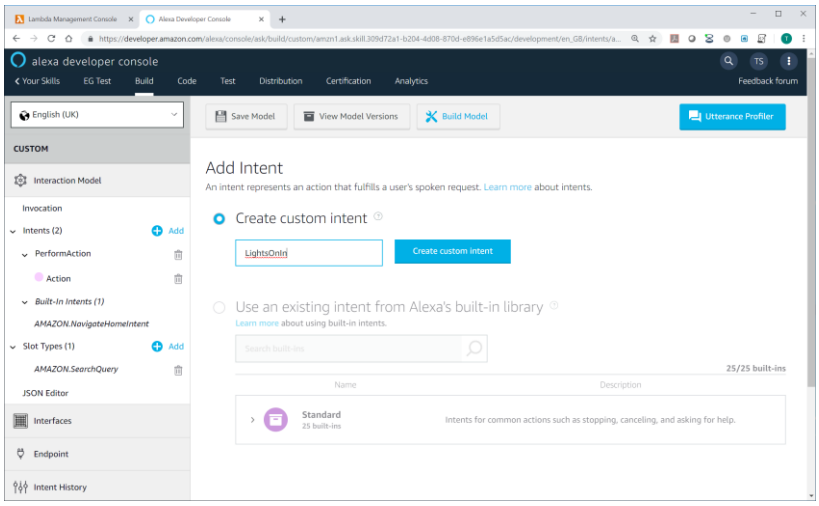


65. Click **Edit** under actions on the right side of your Event Ghost skill.

Note: If you want to use an invocation name other than "Event Ghost" create a new skill following step 25 to 48 before continuing with the next step.



66. Click on the **Add** button next to **Intents** in the left hand column.



67. Enter a name for the intent. **LightsOnIn** is used in this example.

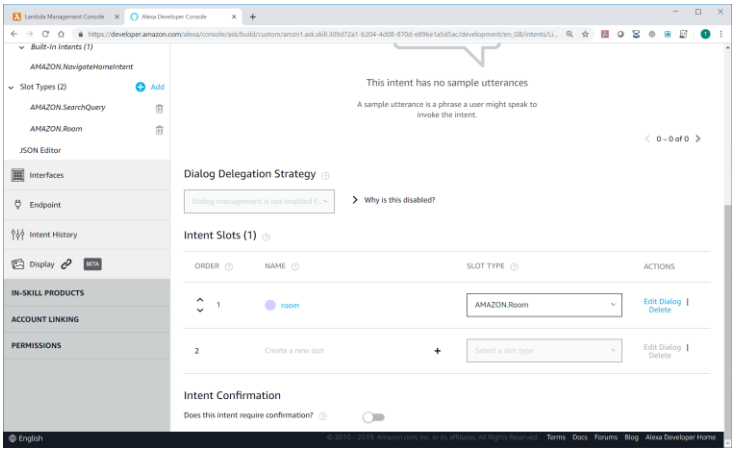
68. Click **Create custom intent**.

69. Click on **Create a new slot** and enter a slot name, in the example I have used “room”.

Note: A slot is variable that can be configured within your utterance. In this example the slot will be the room name.

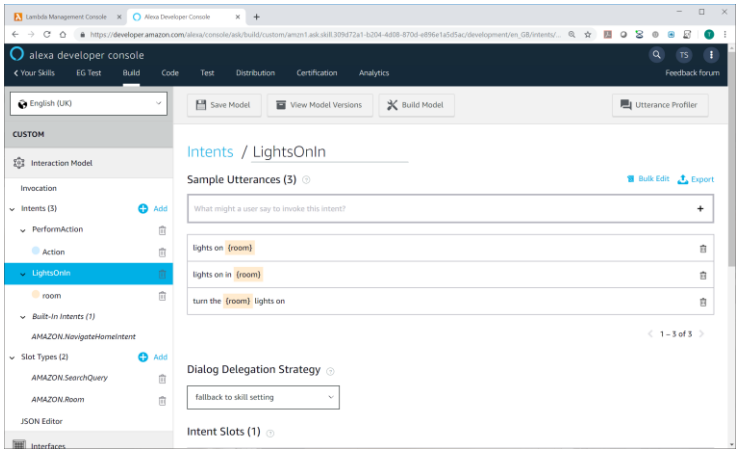
70. Change the Select a slot type to **AMAZON.Room**.

Note: Configuring the slot to AMAZON.Room means you are using a slot preconfigured and optimised by Amazon for recognising words relating to rooms. There are many options that may be applicable to other commands you may wish to program.



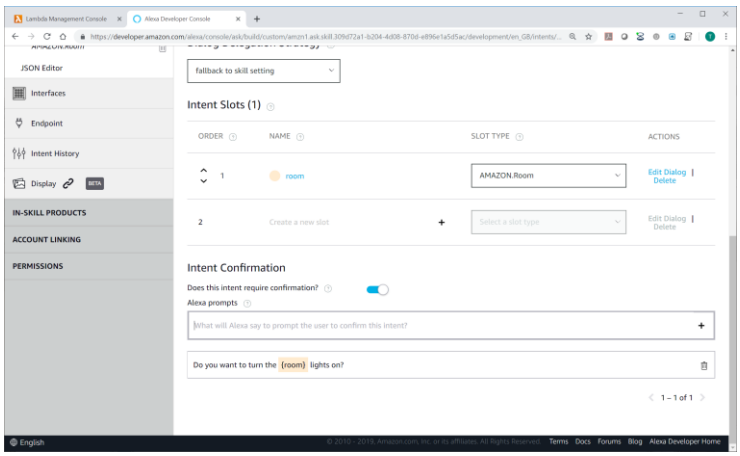
71. Add Sample Utterances. In this example I have used:-

    - turn the {room} lights on
    - lights on in {room}
    - lights on {room}



72. Drag down to the bottom of the window and turn the “Does this intent require confirmation?” slider to the on position.
73. Add the **Alexa prompt,** “Do you want to turn the {room} lights on?”.

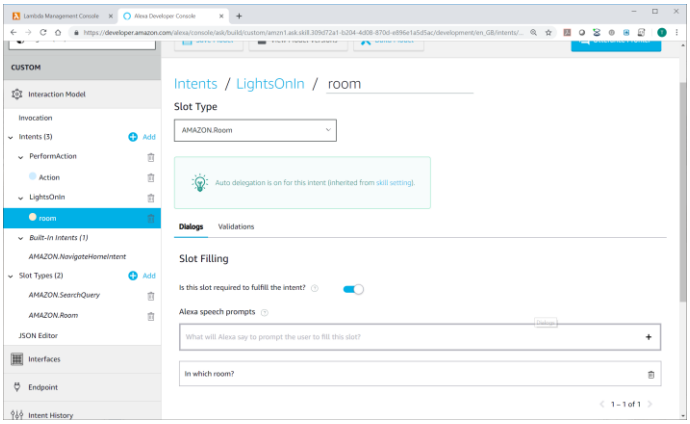
Note: Steps 72 and 73 are optional. You can leave this disabled if you do not want Alexa to confirm the command has been understood correctly.



74. Click on the slot **"room"** in the left hand column.

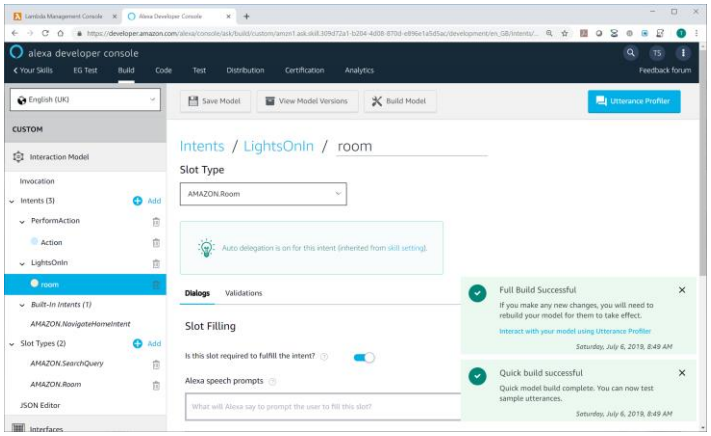
75. Click the "**Is this slot required to fulfil the intent?**" slide to the **on** position.

Note: You could add the rooms to the User utterances if you wanted to restrict the options to only applicable rooms and ensure Alexa understand the room is understood correctly.



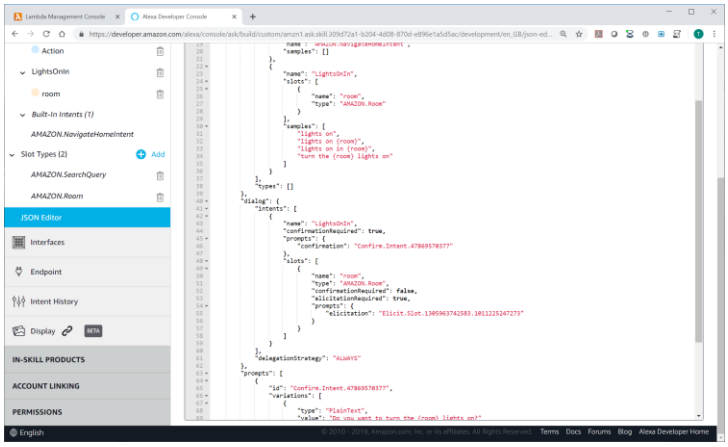
76. Click **Save Model** and then **build model**.

77. Confirm the build is successful.



78. Let's take a look at the generated code, click on **JSON Editor** in the left side column.

Note: You will see that the JSON code has been automatically generated by the configurator. You can modify either directly in the editor or configurator. The configurator is required to create the references between the prompts and the confirmations, so I recommend using the configurator for most use cases.



79. Test your skill works.
    - Say “Alexa, event ghost”, Alexa should respond with one of the welcome responses configure in the event ghost python script.
    - Reply “lights on”, Alexa should respond “in which room”
    - Reply “Dining room”, should prompt “do you want to turn the dining room lights on”
    - Reply “Yes”, Alexa should respond “trying command lights on in dining room”.

Note: You can shorten the Alexa command request in a number of ways when the skill is configured in this way. Try these commands:-

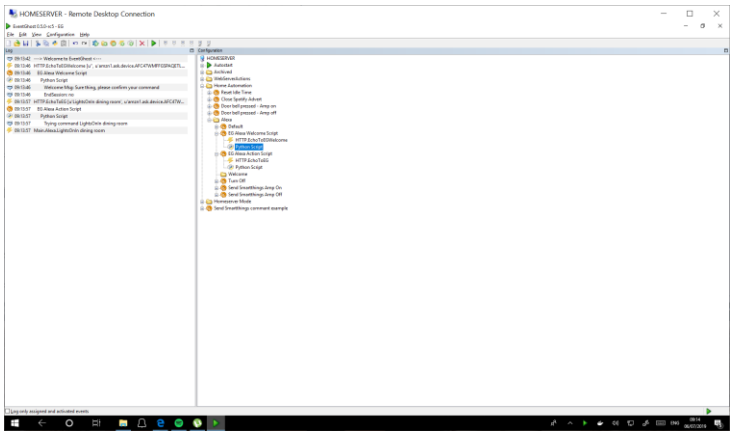
- “Alexa, ask event ghost to turn the dining room lights on”

  Note: this works because we added the utterance “turn the {room} lights on” in step 71.

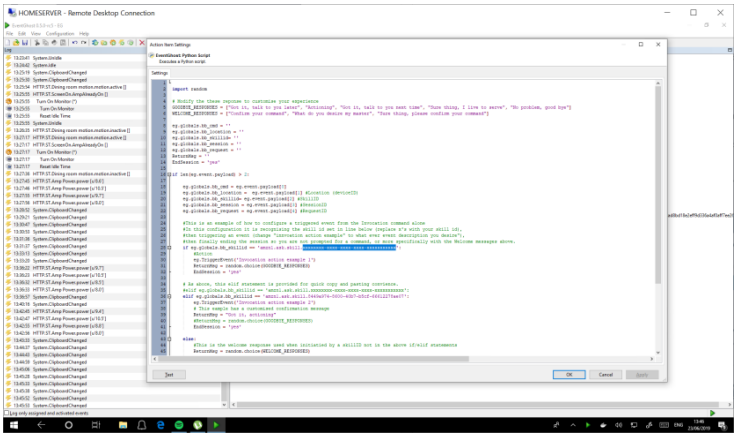
- “Alexa, tell event ghost lights on”

  Note: Alexa will then skip the welcome response and just ask you to confirm which room.
  Note: tell and ask in above commands are interchangeable.

80. Open EventGhost and check that you have a trigger event for “Main.Alexa.LightsOnIn dining room” in the log.



81. Let’s change the acknowledgement to be relevant to the command. Open the **EG Alexa Actions Script** python script.



Note: Code in sample file may vary from image

82. Remove the # symbols from 4 code lines under the #This is an example of the command "lights on" line:-

    elif eg.globals.bb_cmd.startswith('LightsOnIn'):
        eg.TriggerEvent('some_lights_on')
        ReturnMsg = 'Bing! the ' + eg.globals.bb_cmd.partition(' ')[2] + ' lights are on.'
        EndSession = 'yes'

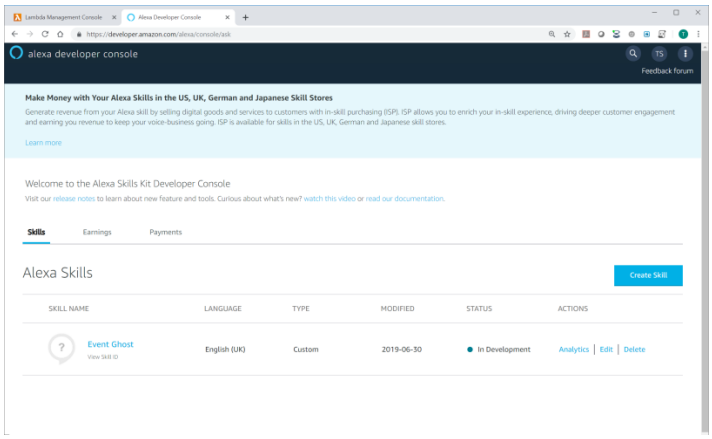
83. Click **Apply**.
84. Test your skill once more. “Alexa, ask event ghost to turn the dining room lights on”, then confirm “yes”. This time Alexa should respond with “Bing! The dining room lights are on.
85. You can change the response by customising the ReturnMsg. The eg.globals.bb_cmd.partition removes the first word from the parsed command, in this case LightsOnIn is removed so that just the room name is used.

Option 2 Configuration tutorial – Invocation and intent:-

- Shorter (Invocation, Intent)
    - Example 1: Alexa, "Event ghost", " play music"
    - Example 2: Alexa, "ask Event ghost to play music"
- Benefits:
    - Requires a little effort to configure
    - Provides restricted options to ensure Alexa understands the command.
    - Can be configured with multiple utterances (word variances)
- Drawbacks:
    - Not as short as it can be

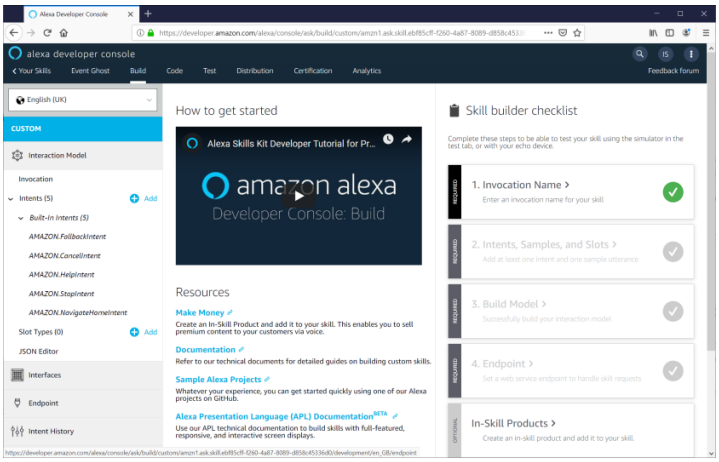
The following steps are mostly a reduced version of the steps in option 1.

86. Open the alexa development console webpage https://developer.amazon.com/alexa/console/ask

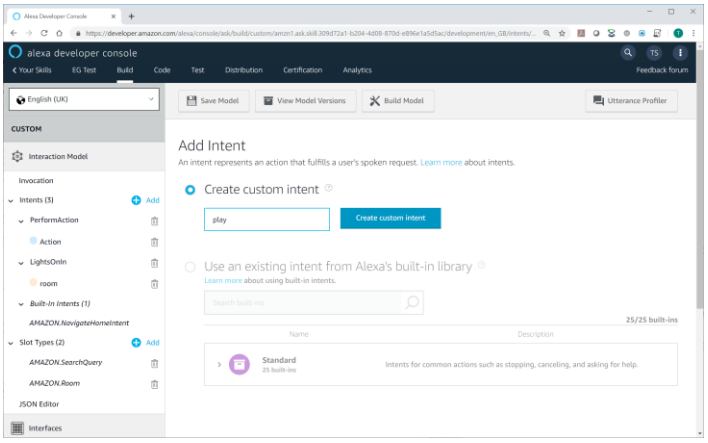


87. Click **Edit** under actions on the right side of your Event Ghost skill.

Note: If you want to use an invocation name other than "Event Ghost" create a new skill following step 25 to 48 before continuing with the next step.



88. Click on the **Add** button next to **Intents** in the left hand column.

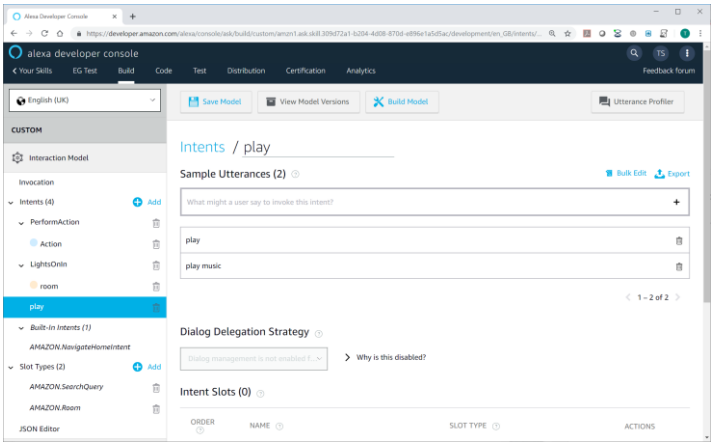


89. Enter a name for the intent. **play** is used in this example.

90. Click **Create custom intent**.

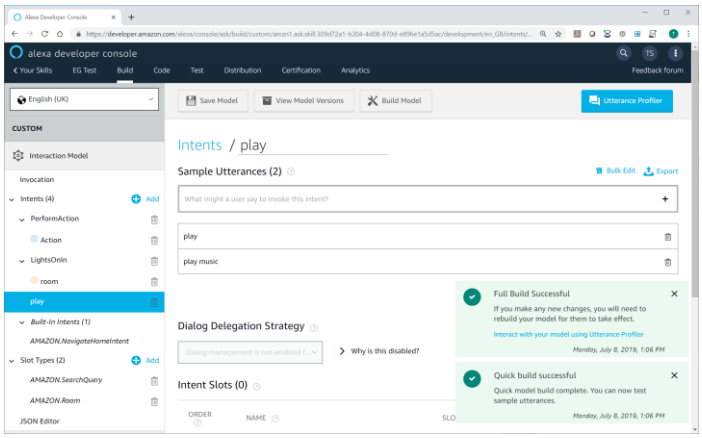
91. Add Sample Utterances. In this example I have used:-

    - play music
    - play



92. Click **Save Model** and then **build model**.

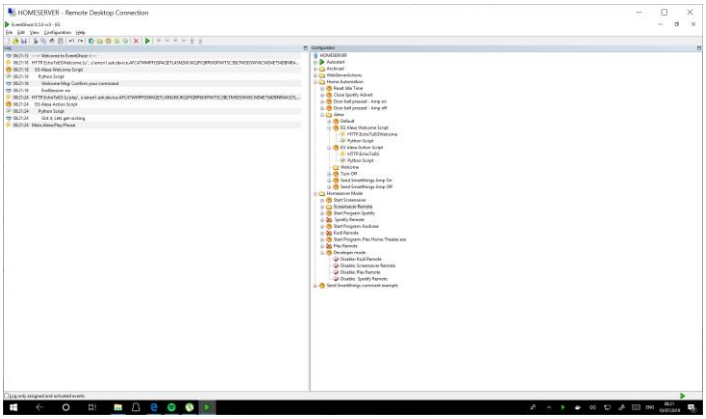
93. Confirm the build is successful.



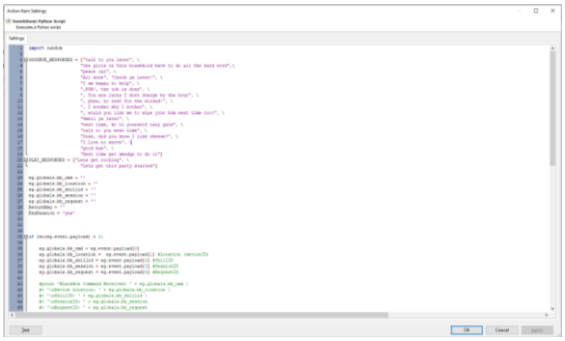
94. Test your skill works.
    - Say "**Alexa, event ghost**", Alexa should respond with one of the welcome responses configure in the event ghost python script.
    - Reply "**play**" Alexa should respond "got it, let's get rocking" or "got it let's get this party started".

Note: The play command has already got a preconfigured response in the EventGhost action python script. Let's take a look at it in the following steps.

95. Open EventGhost and check that you have a trigger event for "Main.Alexa.Play/Pause dining room" in the log.



96. Open the **EG Alexa Actions Script** python script.



Note: Code in sample file may vary from image

97. Firstly notice the **PLAY_RESPONSES** line at the top of the script.

Note: These are the returned confirmations that Alexa repeats when the command is performed. The are selected randomly when the Return message contains random.choice(PLAY_RESPONSES).

Note: You can edit these as you desire, or create new ones for new skills you create.

98. Now take a look at code line 53 to 51. This is the section of the script which is processing command and defining the actions and response to the intent play (defined in step 89).

Note: eg.TriggerEvent('Alexa.Play/Pause') calls the trigger event that occurs in the EventGhost log. You can edit the Alexa.Play/Pause to whatever you desire.

Note: The **EndSession** should always be set to **yes** for any recognised command in the action script. Options should be built in the Alexa skill as per configuration in Option 1.

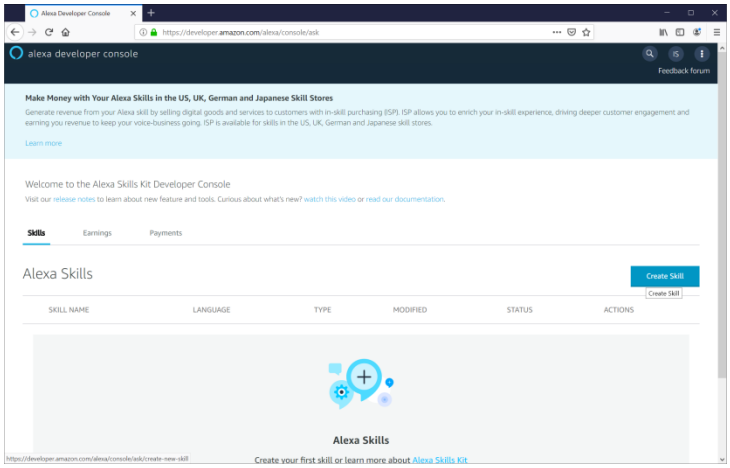
Option 3 Configuration tutorial – Invocation only:-

- Shortest (Invocation)
  - Example 1 Alexa, "initiate music"
  - Example 1 Alexa, "welcome me home"
- Benefits:
  - Few words required
- Drawbacks:
  - Most effort to configure, requires an Alexa skill per command, linked by the skill ID to the EG python code.
  - You have to stay clear of Alexa's trigger commands (and there are loads which you will stumble over lots! For example "Play" or "start" would have been the obvious choice rather than "initiate music", if they weren't restricted!)

99. The following steps are very similar to the steps completed previously, the only differences are:-
    - You are going to set the skill name to something relevant to the command you are making
    - Choosing your skill name will require more thought to avoid use of Alexa trigger words.

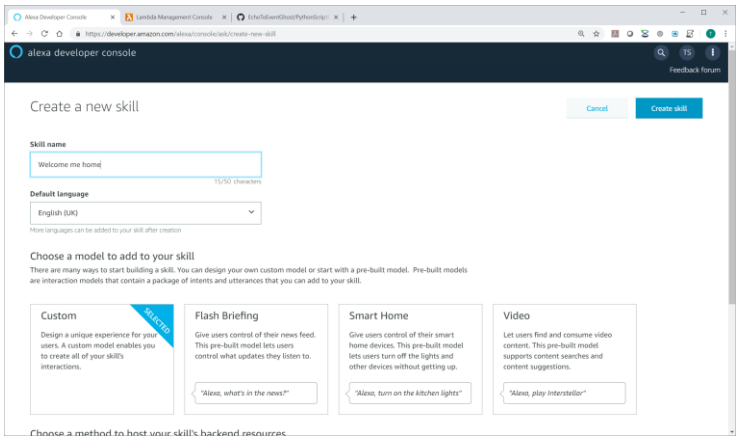
100. Open the **alexa developer console** web page.

101. Click on **Alexa Skills Kit** under the amazon alexa section.



102. Click **Create Skill**.

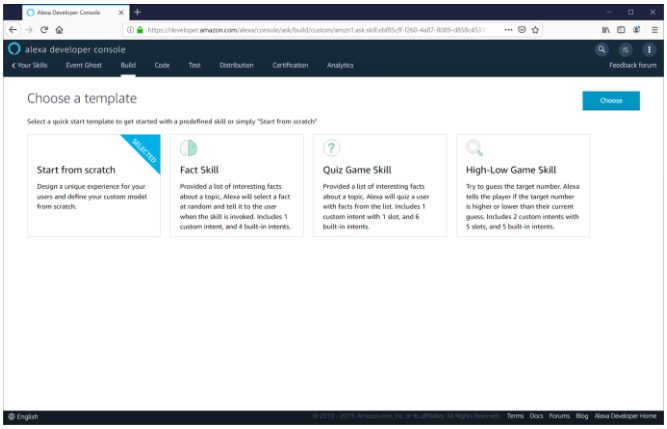
103. Name your skill. This is the name displayed to users in the Alexa app. In this example I have used **welcome me home**.

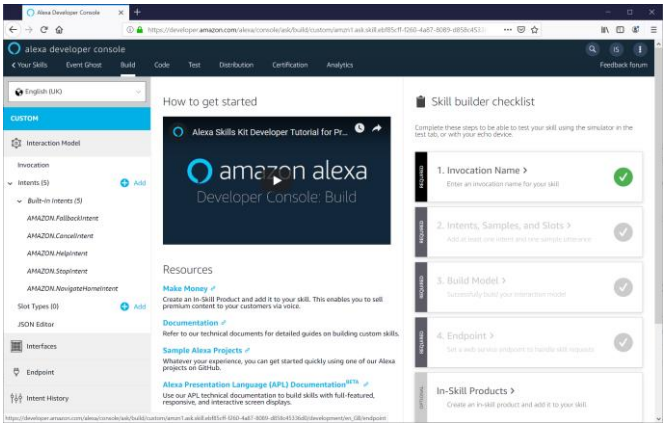


104. Leave **Custom** and **provision your own** options selected.

105. Click **create skill**.

106. Leave from scratch selected, click **choose**.

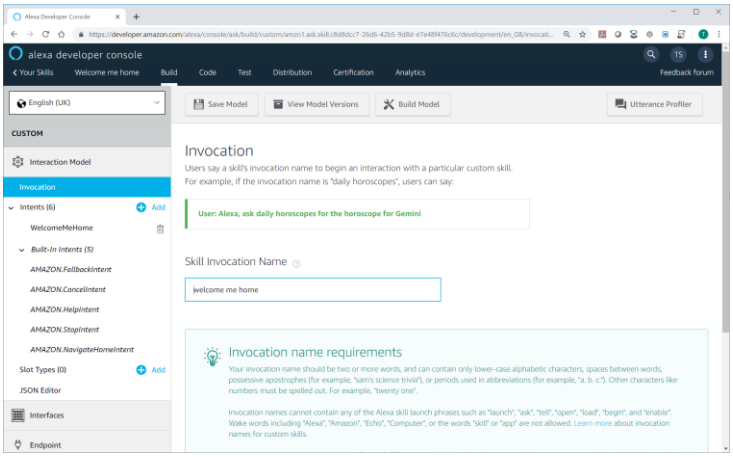




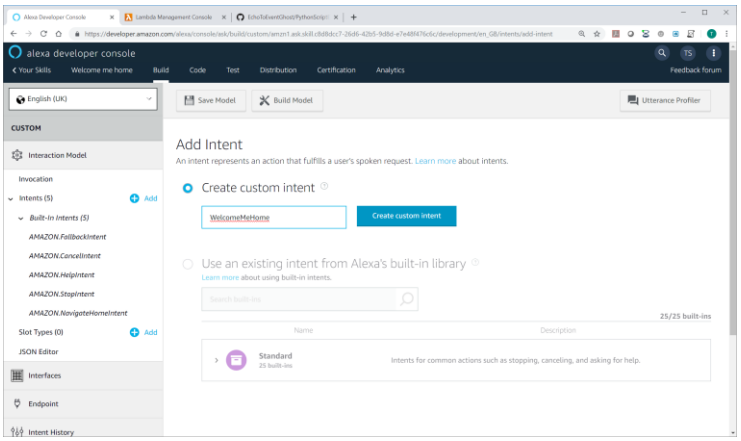
107. Select Invocation in the left side column.

108. Create an invocation name. This is the word or phrase that users will speak to activate the skill. I have used **welcome me home** in the example. Amazon recommends against signal word invocation name.

Note: Invocation name cannot contain the launch words "ask", "begin", "launch", "load", "open", "play", "resume", "run", "start", "talk to", or "tell".



109. Click on the **Add** button next to **Intents** in the left hand column.



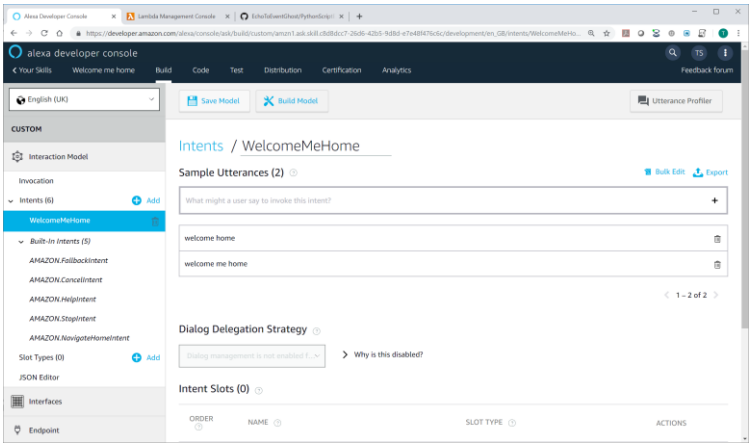
110. Enter a name for the intent. **WelcomeMeHome** is used in the example.

111. Click create custom intent.

112. Add Sample Utterances. In this example I have used “welcome home”.

    Note: No spaces are allowed.

    Note: Technically steps 109 to 112 are irrelevant as we are not going to use the intent as part of the command, however they are required by the Alexa skill building tool before you can build the skill.



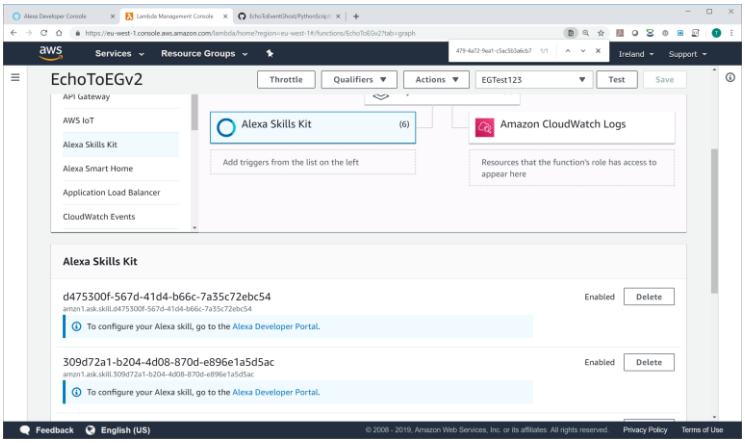
113. Click **Endpoint** in the left side column.

114. Select **AWS Lambda ARN**.

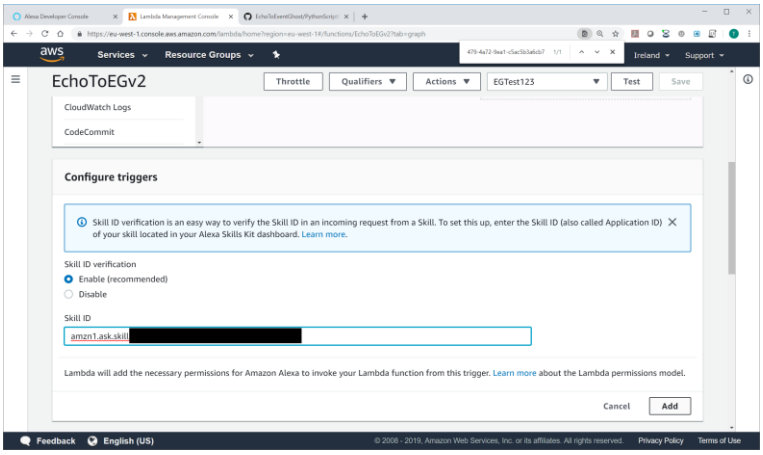
115. Click Copy to Clipboard to copy your skill id.

116. Go back the AWS Console tab as left in step 22.

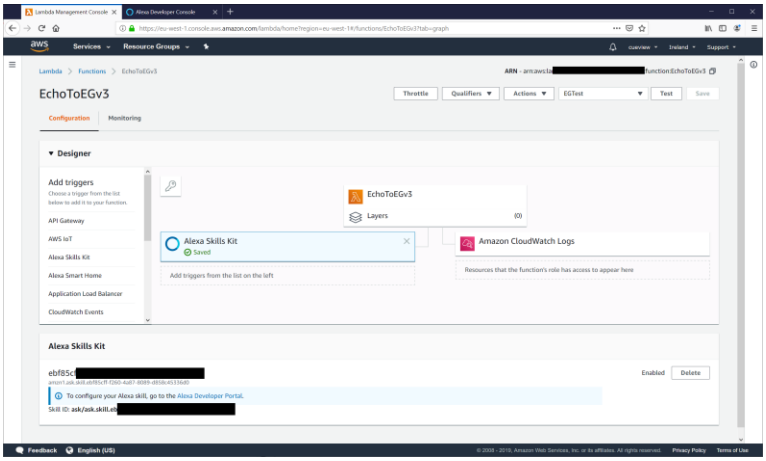
117. Click **Alexa Skills Kit** to add another skill and bring up the configure triggers section.



118. Paste the skill ID into the **Skill ID box** and click **Add**.

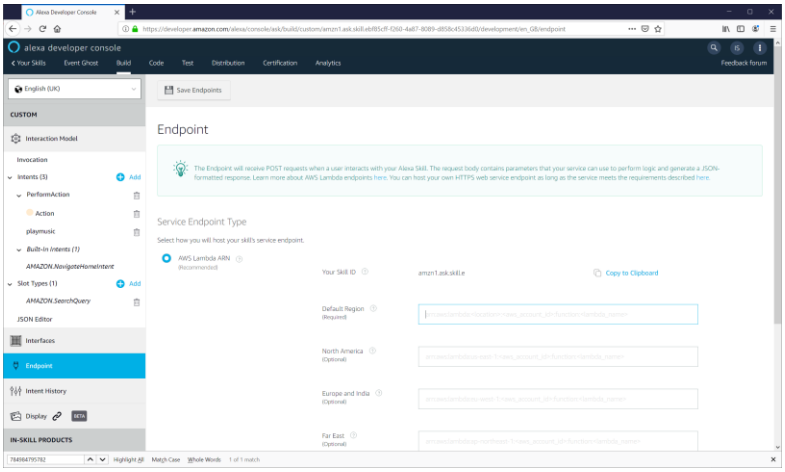


119. Click **Save**.



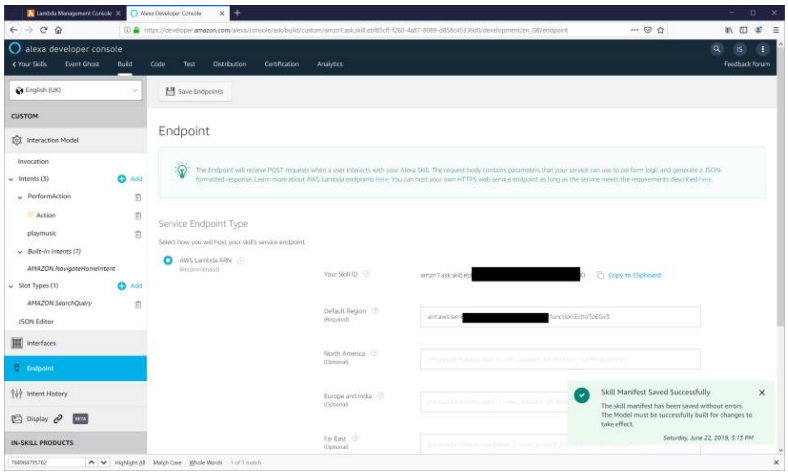
120. Click on the copy button to copy the ARN address to clipboard.

121. Return to the Alexa skill tab.



122. Paste your **ARN number** into the **Default Region**.

123. Click Save **Endpoints**.



124. Check that the **Skill Manifest Saved Successfully** message appears.

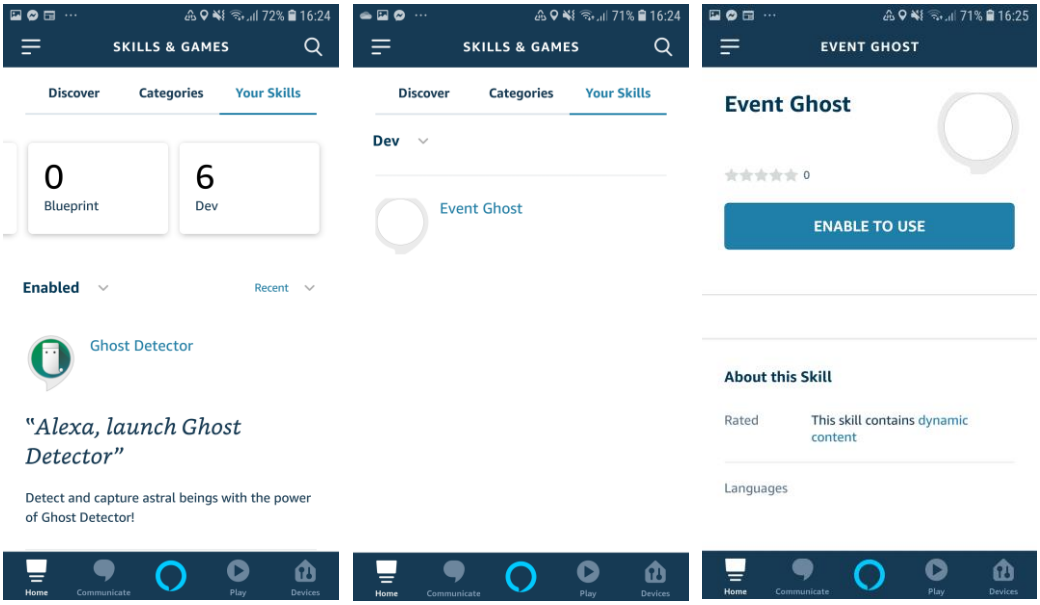
125. Click **Innovation** in the left side column.

126. Click on **Save Model** and then **Build Model**.

127. A message box confirming the build was successful should appear.

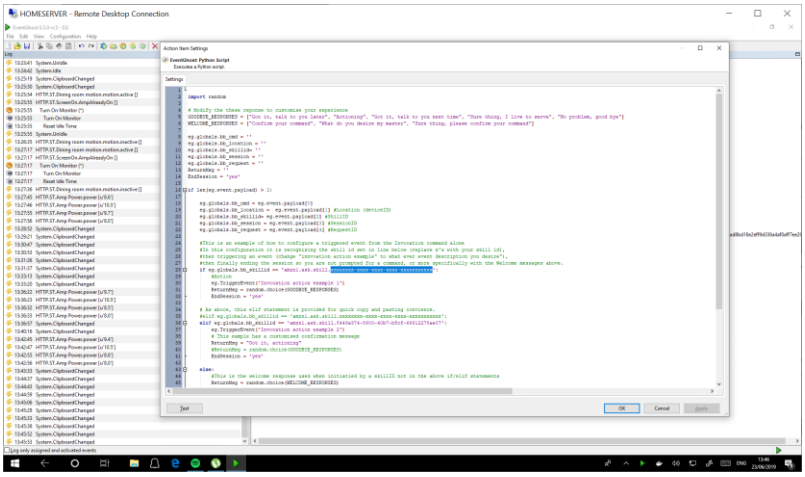
Note: There is no need to Publish the skill.

128. Open the **Alexa app** on your **mobile phone**.
- Click on the burger menu on the top left and select **SKILLS & GAMES**.
- Click on **Your Skills**.
- Slide past the Enabled, Updated and other App groups to the Dev group appears and select Dev.
- Open the skill (Event Ghost or whatever your name it)
- Unless already enabled automatically, select **ENABLE TO USE**.
- Close the app.



129. Open EventGhost

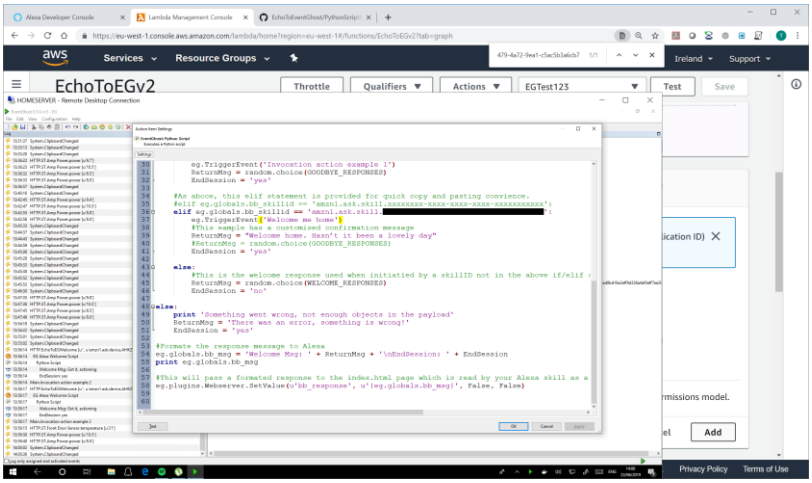
130. Open the Python script associated to the **EG Alexa Welcome Script.**



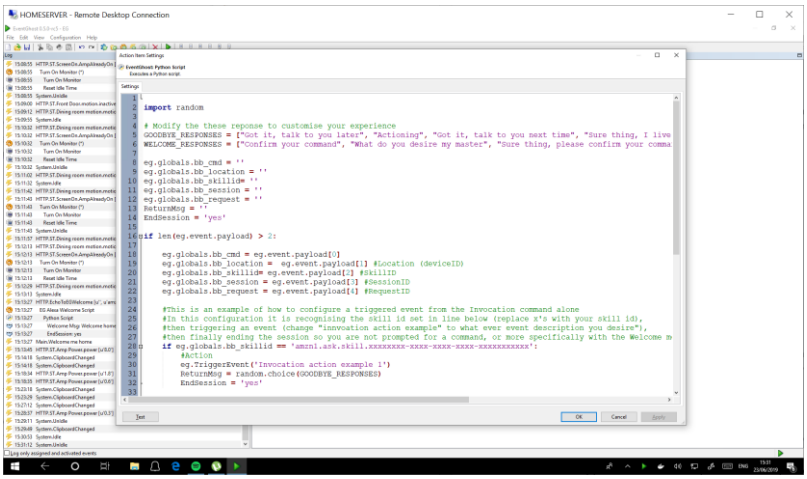
131. Paste the skill ID copied in step 115 over the amzn1.ask.skill.xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx text. Ensure the apostrophes (') remain either side of the ID.

132. Customise the trigger command and return message to suit your needs. Examples:-
    - Existing code: eg.TriggerEvent('Invocation action example 1')
    - Example altered code: eg.TriggerEvent('Welcome me home')

    - Existing code: ReturnMsg = random.choice(WELCOME_RESPONSES)
    - Example altered code: ReturnMsg = "Welcome home. Hasn't it been a lovely day"



133. Test your skill works. "Alexa, welcome me home".



Other useful tips:-

EventGhost welcome python code options:-

1. Open your Alexa Welcome python code created in step 50.
2. There are multiple options here that can be configured. Let's go through the key ones:-
3. You can modify the welcome responses that you are greeted with when you initiate the skill by using the invocation name alone. Simply add, delete or replace the response in the "WELCOME_RESPONSES = "code line.
4. Note: Ensure the responses are surrounded by quotations (") and separated by commas (,).
5. Note: Do not use single commas in the word punctuation, they would be seen as python code and will cause errors.