International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 10 | Oct 2019 www.irjet.net p-ISSN: 2395-0072

# Amazon Echo Controlled Smart Web Application: Integrating Echo with Website using Web Socket

**Shivam Shukla[1]**

[1]*Student (Information Technology), IMS Engineering College- Ghaziabad, U.P, India,*

---

**Abstract –** *In the present era of science and technology, we can see many applications of IoT, Artificial Intelligence, Machine Learning etc. We have heard of Smart Home, but we may not have heard of Smart Website in which, we do not have to control the website physically, we just give commands to our Echo device and it will get the things done. We can make our websites smart by integrating a personal assistant to it. There are several trending personal assistant such as Amazon Echo, Google Home, Cortana, Google Assistant etc. In this paper, we focus on how to configure Echo device to identify our commands and then integrating it with the website so that it can be controlled using Amazon Echo.*

***Key Words*: Smart Website, Amazon Echo, Amazon Alexa, Web Socket, automation.**

## 1. INTRODUCTION

Amazon Echo is used to get the voice commands form user. It internally uses ALEXA, which is a personal assistant developed by Amazon. Amazon's assistant is a female voice that talks to us in a conversational manner, ready to help us with many tasks [1]. We may have used the Echo device for various purposes such as listening songs, making to-do list, setting alarms, etc. We can also use the Echo device for controlling the website or to give commands to website. First we need to create the Alexa Skill for making the Echo device, identify and respond to our commands. After that we need to integrate the device with website using Web Sockets.

Suppose the college ERP is a smart website (Alexa controlled), then the student can give following command after logging in to the ERP -

User - "Alexa, what is my attendance."

Alexa -"Your attendance is 80 percent."

For e-commerce website, users can login to their accounts and they may give following commands -

User – "Alexa, print the last order invoice."

Alexa – "Printing the invoice for your last order."

And the invoice will be printed by the system on which the user is logged in. If the user is not logged in, Alexa will prompt the user to log in first.

### 1.1 TECHNOLOGY STACK

The technologies used are-

1) Amazon Echo – This device identifies the user command and responds accordingly.

2) Alexa Skill Kit – This is the API used to create a custom skill for Echo devices so that it can identify the website related commands.

3) AWS Services – Some AWS Services such as Lambda, SES, DynamoDB are used. Lambda is used for hosting the Alexa Skill code( we do not need to host the code on our server explicitly)[2] , SES is used for sending mails to configure the Echo device with user account by mailing the device ID, DynamoDB is used to map the username with Echo Device ID.

4) Web Socket – This protocol is used for the communication between Echo device and our website. This is the component which is responsible for Echo to respond and make website act as per the given command.

## 2. STEPS INVOLVED

The whole process of developing a smart website (voice controlled) can be divided into certain steps –

### 2.1 SETTING UP ALEXA SKILL

In this step, we create Alexa Skill to configure the Echo device to accept commands as per the requirements of our web application. For this, we need to go to developer console of Amazon and then create a Alexa Skill. There, we have to create the Intent (it is the name used to trigger the code when a particular command is given and used while writing alexa backend code) and define the Utterances (what commands we want to give to Echo, ex: Show invoice number 78)[3].

The utterances will be based on what features we are offering to users. Suppose our website has only one voice command: 'What is the sale for last month?', so for this, we have to create one intent and some utterance. The intent name will be used in our backend code to perform some task when user give the command.

After setting up the intents and utterances, we need to write the Alexa backend code to specify what is to be done when a

command is given. First we created a NodeJS project and install the ask-sdk (alexa skill kit ) package to write the skill handler code.

We make use of Web Socket library. This will allow us to send message to our application (after receiving command from user) and then receive the message from our application (what alexa will prompt to user as an answer) to respond accordingly. After writing this code, we upload this file to AWS Lambda, for hosting.

Now we can test our skill and after that we can publish our skill so that it is openly available for other users. The users who want to use the voice command features has to search for our skill in the Alexa skill store and enable it.

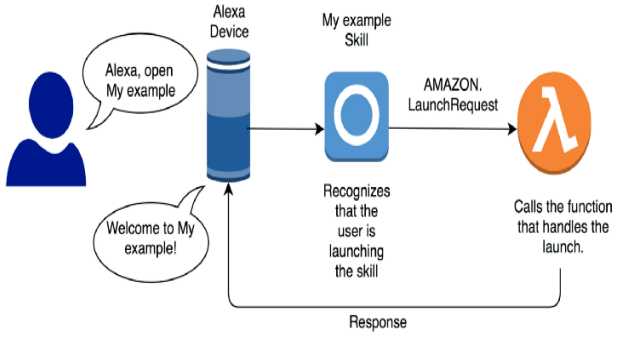



**Fig -1**: Alexa: What happens behind the scene[4]

### 2.2 SETTING UP APPLICATION BACKEND

In our application, we need to first give option to user to Enable / Disable Alexa. By clicking this button, user can update the alexa status in the user mapping table in DynamoDB. Only after enabling alexa, user can be able to give commands to alexa.

We have to decide what features will be offered to users and based on which, we can define the event handlers. In our event handlers, we receive the message (in JSON) from Alexa, and perform the task based on the message received from Alexa.

If the command is a query such as Alexa, what is my attendance? In this case, we have to fetch our application database and then send the attendance as JSON to the Alexa. Now alexa will respond to user by telling the attendance.

If the voice command is to perform some task such as Alexa, show me the invoice number 78. In this case, the validation is performed on the invoice number and the Alexa will respond accordingly and also the invoice will be displayed on the system screen on which the user is logged in.

### 2.3 SETTING UP WEB SOCKET SERVER

The last step is to setup the Web Socket server, which is responsible for the communication between the Amazon Echo and our Website. We used a simple client-server concept [5]. After getting the command from user, Alexa sends that as JSON to this server, and this server decides for which user this command is and then sends to the application on which that user is logged in.

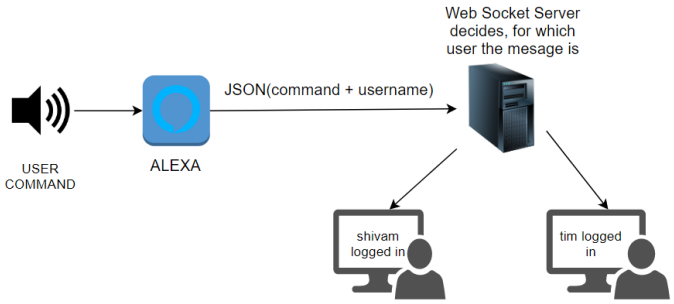



**Fig -2**: General overview of working of web socket server

### 3. WORKING OF WHOLE STUFF TOGETHER

After completing above three steps, the thing works in following manner-

- The user has to first login to the website and has to enable alexa from the application. This will update the alexa status column in User Mapping Table so that Alexa can now respond to the user commands. Application also sends a message to web socket server to establish connection with it and server uses this message to add the username to the session.

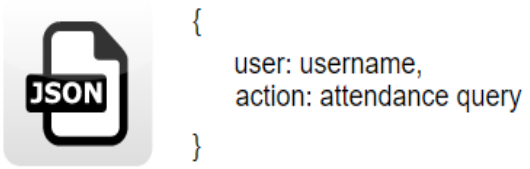
- User can now invoke the skill by giving command to Echo, 'Alexa, open <application name>.' (We have to set this while creating the Alexa Skill). If the user is not logged in, Alexa will prompt the user to login first as the alexa status column in the database is disabled by when the user logs out. If the user is logged in and enabled the Alexa, user can now continue by giving commands.

- Suppose user gives the command – 'Alexa, what is my attendance.' On receiving this command, alexa will first seek the database for the username associated with that Echo device ID to identify which user is linked with that Echo device. It also fetches the alexa status to ensure that whether the user has allowed Echo device to receive voice commands or not.

- After getting username and alexa status, if the status is enabled, alexa sends message to server and server uses this message to add this alexa device to the session using the username.

- At this time, our application and Alexa are now connected to web socket server using the same username.

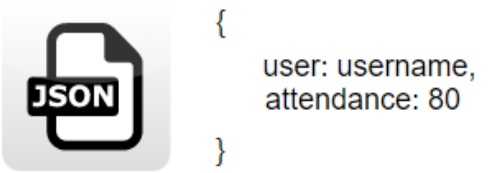
- After that, Alexa sends the command along with the username in JavaScript Object Notation to the Web Socket Server. The JSON will be –




Here the 'user' field refers to the user to which this message will be sent. Then Alexa waits for the response from the client application.

- Now the role of web socket server is to identify the user by extracting the username from the JSON received by Alexa. After this, the server forwards this message to the appropriate user.

- Now in our application, on receiving the message from server, the validation is done (if required), and the action is performed based on the command. Ex: In case of attendance command, the application seeks the database to find the attendance of that particular user and sends to the server. The JSON will be –




In this way, both, Alexa and our web application communicate with each other using web sockets.

Following diagram explains what is happening behind the scene -

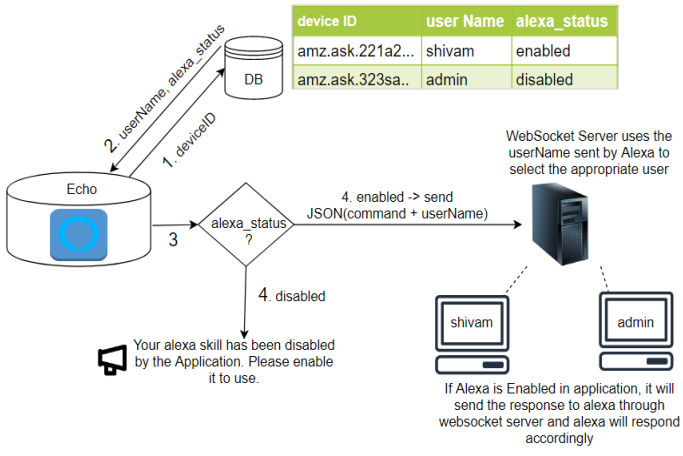



**Fig -3**: How the things are working

## 4. Mapping of Echo device ID and username

For the first time, the Echo device ID and username needs to be mapped in order to ensure that the given command to a particular Echo device affects only the account linked with that device (not all user accounts). To do this, we created a intent in our Alexa Skill in which user gives a command – 'Alexa, configure user <username>.' On receiving this command, Alexa mails the device ID (on which this command has been given) and the username to us so that we can map the Echo Device ID and username in our database.

After successful mapping, user can give all the commands offered by our application using that Echo device.

In case, the Echo device is not registered with any account, and if try to invoke our Skill, Alexa will seek the database for username, as there will be no username, it will simply prompt 'This device has not been yet configured with any account. To link this device with your account, give command, configure user followed by username and your account will be configured within 1 day.' and alexa will end the session so user will not be able to give commands to Alexa. Now we get the mail sent by alexa containing the deviceID and username and we can add an entry to user mapping database to link that device with the username.

## 5. Ensuring Security

As this concept uses a third party Web Socket Server, so in case we want higher security, we can encrypt the data before sending to the server. We can use one of encryption algorithms to ensure privacy and protection of our data.

Before sending data to web server, Alexa will encrypt the message and when it reaches our application, there it can be decrypted (in our application code). After decryption and result formation, again the new message is encrypted and sent to server, which will be decrypted at alexa end (in Alexa backend code)

## 6. APPLICATIONS

Integrating the personal assistants to our application can really make the application smarter and present age application.

If Echo is integrated with a large e-commerce or shopping web application, the users can order the things by just giving voice commands.

'Alexa, repeat my last order' to reorder a item, users can know the delivery status just by asking to Alexa, Where is my order?

If echo device is integrated with the college ERP, the faculty can give voice commands such as 'Alexa, print the debarred student list of IT branch.' Or 'Alexa, mail the test schedule to

students.' Student can give commands like 'Alexa, show me the attendance.' Or 'Alexa, Tell the upcoming test dates.'

Suppose a web application for Jewelry shop is smart application. Now the owner can give voice command to display the invoice or bill for a particular customer by simply saying to Echo, 'Alexa, show the invoice number <invoice number>.' Or give command to print the bill for customer by saying 'Alexa, print the bill number <bill number>.' The owner need not physically search for the bill number and then clicking the print button to print it.

## 7. CONCLUSIONS

Now-a-days the personal assistants are widely used and are the most common application of AI. Just integrating the existing personal assistants, we can make our applications smarter and more productive. By using web sockets, our application and personal assistant can communicate with each other.

We can integrate Echo not only with web application but also with any applications whether it is a desktop application or mobile application.

We can configure the Echo for any type of commands and requirements by creating a custom Alexa Skill. The only limitation is our imagination.

## REFERENCES

[1] What is Alexa and what it can do: https://www.digitaltrends.com/home/what-is-amazons-alexa-and-what-can-it-do/

[2] Hosting custom skill on AWS Lambda: https://developer.amazon.com/docs/custom-skills/host-a-custom-skill-as-an-aws-lambda-function.html

[3] Creating a custom Alexa Skill: https://developer.amazon.com/docs/custom-skills/create-custom-skill-from-quick-start-template.html

[4] Alexa: what happens behind the scene, https://www.freecodecamp.org/news/a-beginners-guide-to-the-new-aws-python-sdk-for-alexa-105c0ed45f4e/

[5] How to use web socket to communicate between client and server: https://blog.revathskumar.com/2015/08/websockets-simple-client-and-server.html