Doctor Prompter Alexa Skill

Procedures for Additional Questions

To add more questions to the Airtable database, you can do this in multiple ways. This tutorial will let you know how to add them one by one or by downloading an Excel spreadsheet.

The easiest way to add questions is directly into the database. This will ensure that the format is correct and that each of the rows are input correctly. To begin, you will want to log in to airtable.com. Once you have been logged on, you will want to click on the “MD2B” base. This is your database. On the top left of the screen, you will see 2 tabs, Data and User. Make sure you choose the “data” table. See figure 1.This is where all of the questions are stored for the quiz. You can scroll down to the bottom of the sheet and add the topic (make sure the topic is lowercase), question, and answers per question. Once you add that information, there is another column labeled “answer” that you will want to make sure that each correct answer is entered separately, i. e. potassium or K. The spreadsheet will save and update automatically.

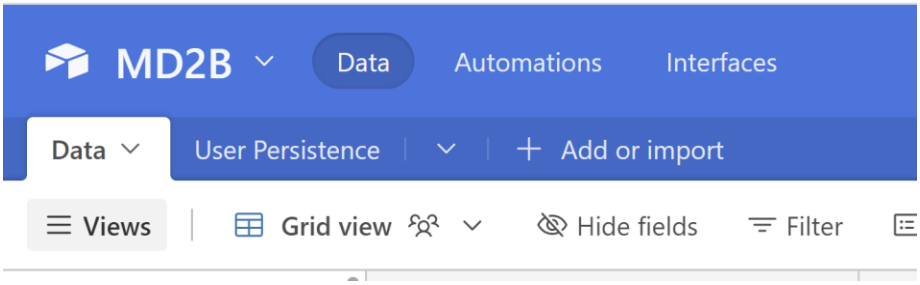

Figure 1. Data table

Another way to add questions is to type them out in an Excel spreadsheet and then upload them to the Airtable base. Make sure that your columns are lined up the same way as the database. Your first column will be the topic, then the question, and then the answer. You will also log into the airtable database and access the MD2B base and then go to the “data” table. There is a drop down icon next to the “data’’ name. When you click this icon, you will then select “Import data” and then choose “Microsoft Excel” see figure 2.

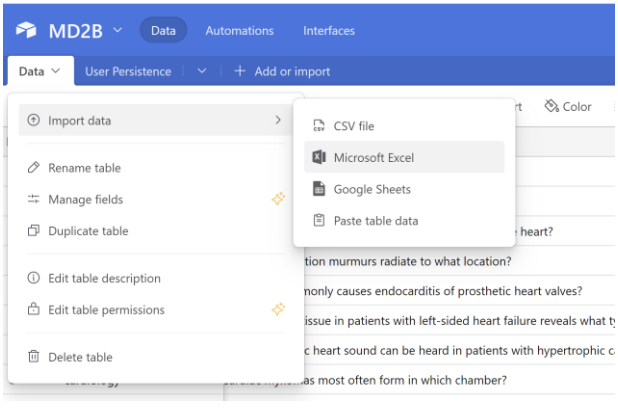

Figure 2. Importing Excel File

A popup will come up where you will select your excel file to upload from your computer. Click upload and then another popup will appear (Figure 3), and you will select the “existing table” and select “data”(Figure 4).

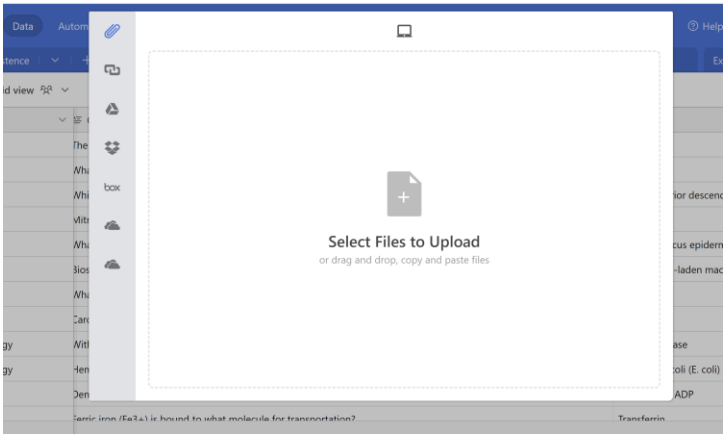

Figure 3. Pop file upload

If this is not done correctly, your new questions could end up being added to a new table or another table. There is more than one table used for this Alexa skill. The other table handles the user’s scores and other information needed for the program. Then choose the “next” button where you will see a preview of the way your data will appear in the Airtable base (Figure 4). Make any adjustments here so that the questions upload to the correct column. You will be able to select which columns to import. This is just a preview screen so you can see how the settings change when you click on different options (Figure 4.1). Once you have previewed the upload, click the “import” button. This will then upload the new questions. Make sure your topics are in lowercase. You will then have to manually add the “multiselect” answers like you would do if you were manually adding the questions (figure 5). This is very important for the Alexa skill. At this point you are done with Airtable and can log out. Changes are saved automatically.

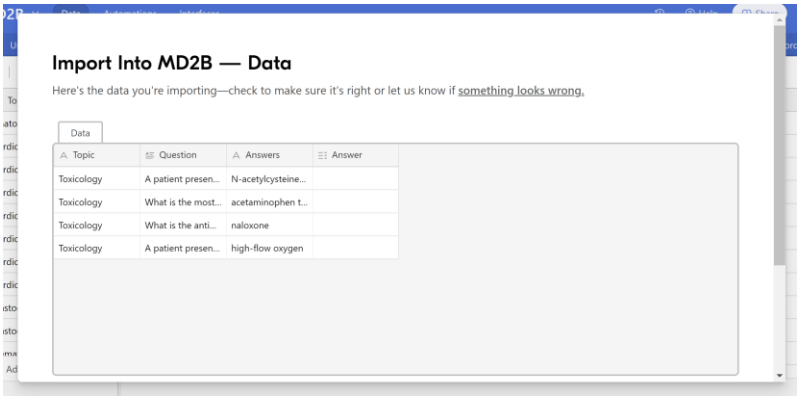

Figure 4. Selecting Correct columns.

Figure 4.1. The bottom of the popup.

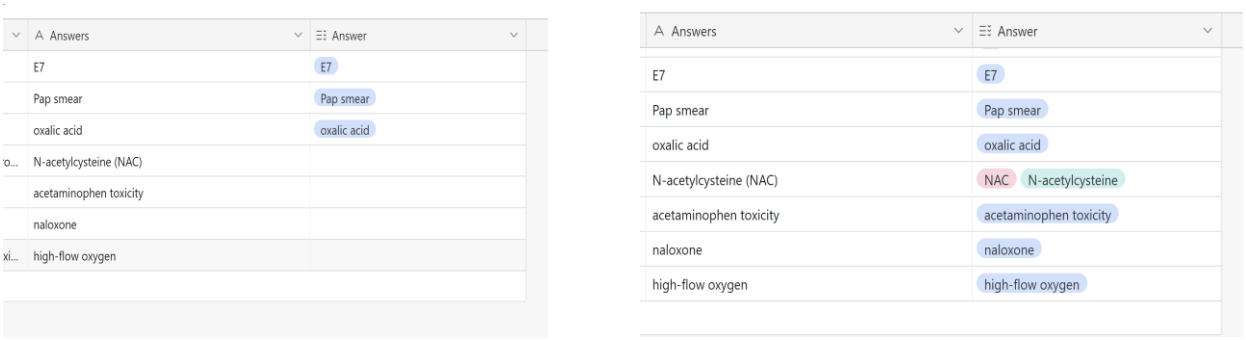

Figure 5. Shows the “Answers” column and where to input the multi-select “Answer” column. The first shows before and the second shows after.

It doesn’t matter which way you add the questions to Airtable, you will then have to add the answers to the Alexa skill. To begin you will log in at developer.amazon.com. Click on the “Doctor Prompter” skill (figure 6).

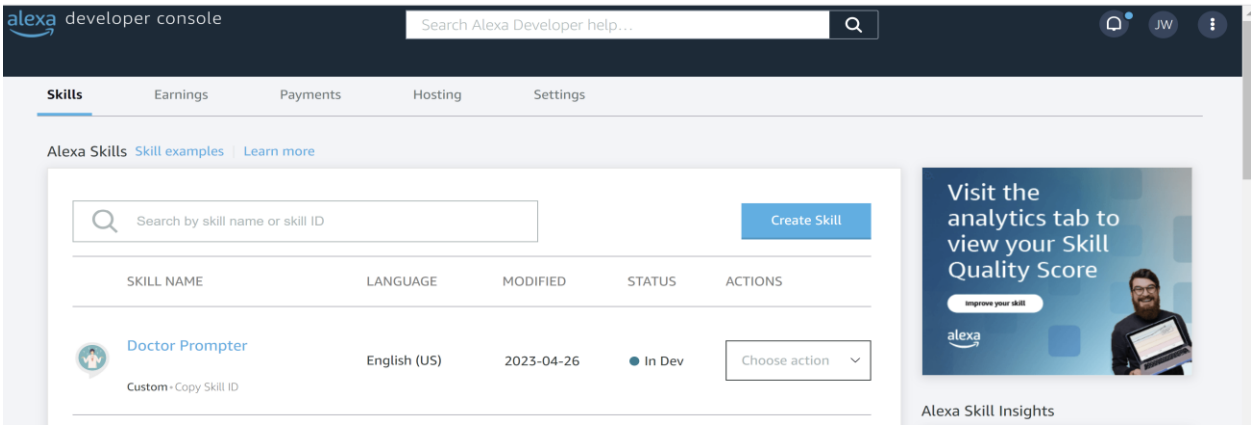

Figure 6. Amazon Developer Console Home

Be very careful only to make the necessary changes or the skill could be modified and not work correctly. On the menu bar, you will click on the “Build” tab. On the left hand side, there will be a menu. You are looking for one called slots. Click there and the slots will appear on the screen. This is what Alexa listens for the user to say. There is one labeled “answer”. Click on it (Figure 7). This holds all the correct answers from the airtable database.

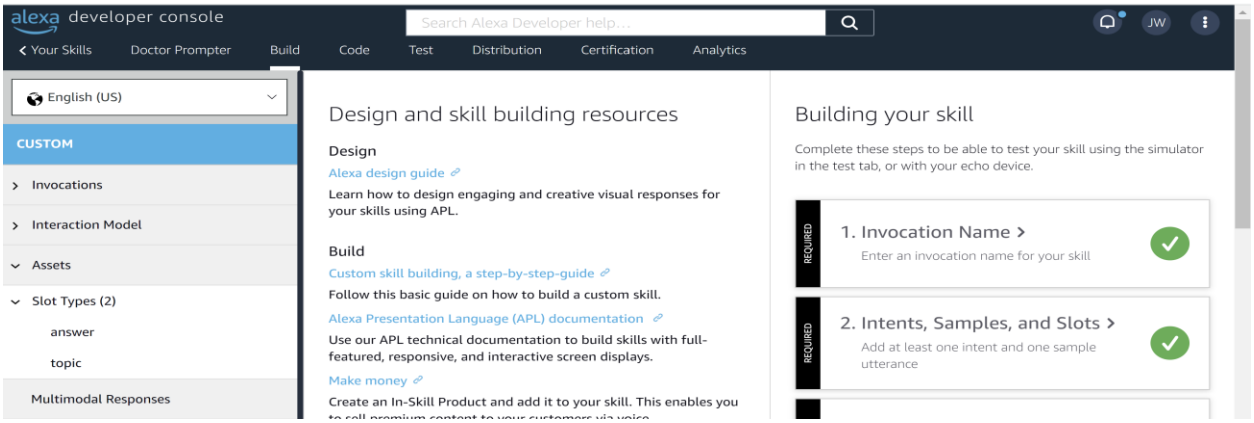

Figure 7. Build Menu

You are going to add the correct answer(s) from the new questions you just entered. They do not have to be in any order. You will begin be typing in the answer at the top and then click the plus (+) symbol (figure 8). If there are any alternate answers or synonyms you will add those where the synonyms line is (Figure 9). Then you will add this answer with the + (plus) button at the end of the line.

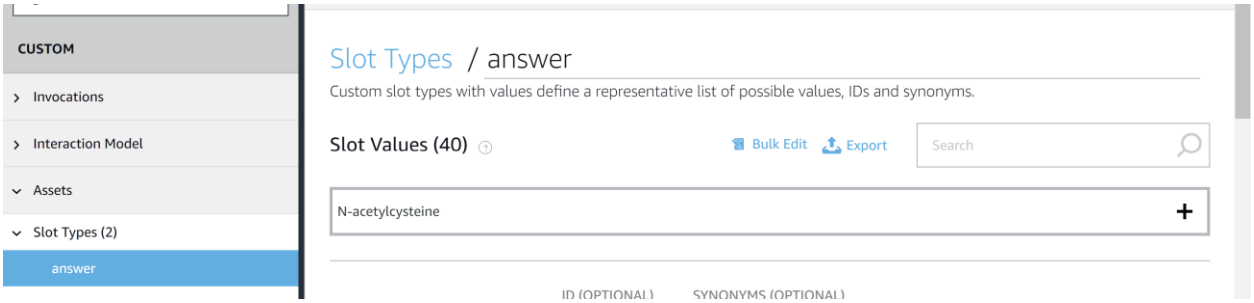

Figure 8. Adding answer slots.

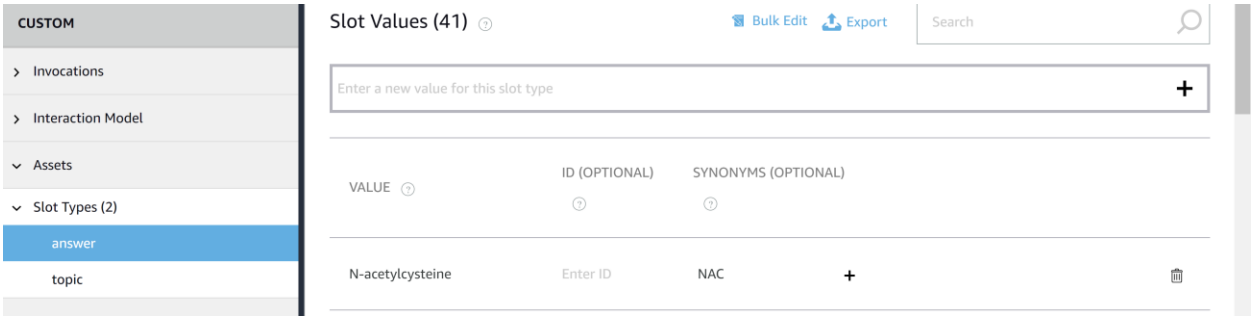

Figure 9. Adding synonyms.

Once you have entered all of the new answers, you will click the “save build” button at the top of the page. Then you will click the “build” button at the top of the page (Figure 10). This will now handle all the new questions and answers that you have.

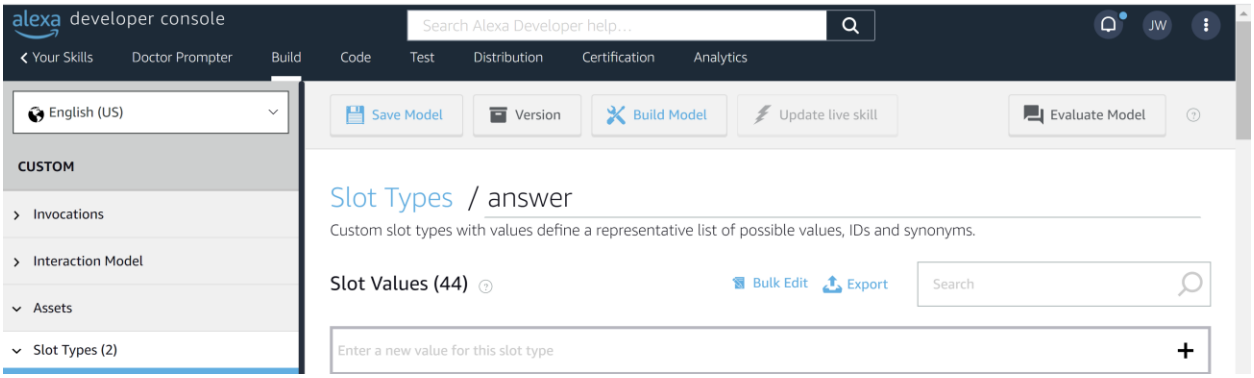

Figure 10. Saving and building model

If any of your questions have a new topic, you will need to follow the exact steps above for the “answer” slot but choose the “topic” slots instead. You will simply add the new topics here.

Procedure for Changing Number of Questions

The quiz has been set up for 5 questions for beta testing. After the presentation, it will be changed to 10 questions. If you want to change the amount of questions asked, follow the instructions below.

Login to the Amazon Development Console and click on Doctor Prompter. Once opened, click on the “code” tab at the top of the screen. Scroll down to the “StudyEverythingIntentHandler” and find the session attributes. You will simply change the number in the line that starts “totalQuestions: 5”. See figure 11. Simply change this number to whatever number you want. You will also need to change the same line under the “StudyOneTopicIntentHandler” as well. Once complete, click “save” and then “deploy”. This will update the skill to your changes.

```
const StudyEverythingIntentHandler = {
    canHandle(handlerInput) {
        return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest'
            && Alexa.getIntentName(handlerInput.requestEnvelope) === 'StudyEverythingIntent';
    },
async handle(handlerInput) {
        const questions = await getQuestions();
        const randomQuestion = questions[Math.floor(Math.random() * questions.length)];
        const speakOutput = `Here's your question: ${randomQuestion.question}`;
        console.log(`Study Everything Stored answer: ${randomQuestion.answer}`);

        const attributesManager = handlerInput.attributesManager;
        attributesManager.setSessionAttributes({
            question:randomQuestion.question, //store question
            answer: randomQuestion.answer,
            isTopicMode: false,  //not in OneTopic mode
            questions: questions, //Store all questions within the topic - may or may not be needed here
            currentQuestionIndex: 0, //initialize current question index to 0
            hasNextQuestion: questions.length > 1, //checks for next question
            nextQuestion: questions.length > 1 ? questions[1] : null, //how many questions
            score: { correct: 0, total: 0 }, // score property
            correctAnswers: 0, // initialize correctAnswers to 0
            //totalQuestions: questions.length, // set totalQuestions to the number of questions
            totalQuestions: 5 // set to 5 for only 5 questions to be asked

        });
```