

This is a process regarding the how to get started with Watson Assistant.

- 1.) Create a Watson Cloud account at cloud.ibm.com
- 2.) If wanting to add more collaborators, go here => user access within the IBM cloud homepage.
 - a. Add in the User's email and the appropriate level of security.
 - b. Add in access groups to easily configure the account services.
 - c. Read further instruction about the access role in roles.
 - d. The user can access the account by changing into the account that is being used as main.
 - e. Read more instructions here =>
<https://cloud.ibm.com/docs/Activity-Tracker-with-LogDNA?topic=Activity-Tracker-with-LogDNA-iam>
- 3.) Create a service called IBM Watson assistant.
 - a. Create a Watson assistant service at the homepage.
 - b. Make sure the service locations are the same throughout the project, if it's different, services are not able to identify the related service/API keys.
 - c. Follow the document here for the official guide =>
<https://cloud.ibm.com/docs/assistant?topic=assistant-getting-started>
 - d. It's best to follow the guide and tweak it a little bit at the end.
 - e. For the creation of the project, it might be best to choose the plus trial for 30 days. This can ensure the service is not used when it's not ready.
 - f. Tips/references for chatbot creation=>
 - i. Do frequent testing and expect what is the outcome of the chatbot.
 - ii. Do play with it first before you create a complex architecture like DyCE.
 - iii. Use the sample app in the beginning, it would cut down a lot of phrase creation and time wasted on setting up.
 - iv. Use this link to integrate the Facebook messenger chatbot. =>
<https://www.youtube.com/watch?v=YyZO7TnA5l0>
 - v. Use this link to integrate the Slack chatbot in Slack channel. =>
<https://www.youtube.com/watch?v=Z2j1HKz2h0A>
 - vi. Use this link for integrate cloud foundry for cloud functions with node.js as a point of connection between chatbot and other service, such as website or other IBM service (same region). => <https://www.youtube.com/watch?v=j8TBqD2rx2o>
 - vii. The code (node.js) that was able to return any question is from here: =>
<https://github.ibm.com/skwatra/DyCE-Watson-Virtual-Classroom/blob/master/DyCE-ServerSideNode.js>
 - viii. There are many kinds of feature snippets that are implemented on the internet, one of those is Google Feature Snippet. So, there is a limitation from just implementing Wikipedia API and only searching their database.
 - ix. With the Watson Search Skill that IBM offers with the Plus plan, it further searches a website or database knowledge document for further detail regarding their organization.
- 4.) Further implementation (possibility with Chatbot and guidance)
 - a. Integrate with Google calendar for appointment.
 - i. This can further book that block of time for the activity. Including teacher appointment with student, what's the schedule like for the next couple of weeks for both teacher and student as well as the school organization.
 - ii. Link for node.js implementation for Google calendar =>
<https://developers.google.com/calendar/quickstart/nodejs?pli=1>
 - b. Integrate the Question Answer session from teacher to student
 - i. This can help the teacher to further understand if the student comprehends the material from time to time during the lecture. If the student does not understand a

certain concept, the chatbot can help with the explanation when reviewing and with the plan to guide the student back on the right track.

- ii. If the student is still struggling to understand the concept, the teacher would be required to explain the concept with a whiteboard system during office hours.
- iii. All of this is to aim at increasing student abilities to understand the education in a better manner, especially for now when students are facing a computer instead of a face-to-face teaching.