

Chatbot Research: Front-end

3 main parts of implementation:

1. HTML/The chat button
2. UI/The chat box
3. Communication with back-end

Basic flow of the front-end:

- HTML code in which you make the chat button and if clicked on it, it opens a chat box.
- The button thus links to an application, i.e. the chat box (e.g. in JavaScript)
 - The chat box can be implemented with JavaScript, for which there are multiple different libraries, but there are also a lot of non-JavaScript options to choose from
 - As there are a lot of different, distinct ways to create the API I could not find ‘the perfect’ solution for this
- Input/output to/from the chat application has to be communicated to the back-end, again this can be done via a lot of different ways (e.g. JSON and Postman)

Possible tools we can use (among others)

- | | |
|----------------------------------|------------|
| • HTML/CSS (duh) | • PHP |
| • Python (Flask, Chatterbot) | • Django |
| • Javascript (React.js, Node.js) | • Postman |
| • AJAX | • Telegram |
| • Fetch | • Twilio |
| • Dialogflow | • RASA |
| • Socket.io | |

Obviously there are a lot of different ways in which the front-end part can be developed, but it honestly it is kind of a jungle with a lot of possible permutations. The problem with the most examples for building a chatbot from scratch is that they have the goal to build a chatbot quickly (e.g. BUILD CHATBOT IN REACT.JS WITH DEEP LEARNING IN 5 MIN), this means that the examples I found just yield some working, but very simple and ugly API result.

Since we are using python for our back-end implementation, I think some implementation of the chatbox in JavaScript will fit well. Here is a nice explanation on how to let your JavaScript code communicate with your python code.