

chatbot_webservice

Intelligent and interactive resume website (responsive) with a chatbot communication using Gradle webservice. The chatbot have the capacity to interact with the résumé information with a personalized style of talk.

2 in 1 chatbot application that can both play the role as a Robot and as a Web-service Chat. Everybody all around the world can contribute into the chatbot application, every message will be located with the name of the users and the corresponding flag. Here a demo between the chatbot and two humans from France :

Chatbot Application | Vincent

B

BOT 🤖 from 🇫🇷 [FR]

12/12/2018 à 11:43:29 : Welcome, I'm Vincedroid, the chatbot of Vincent. Ask me any question about professional stuff and more. If your question isn't recognized, try a keyword.

J

Jessica from 🇫🇷 [FR]

12/12/2018 à 11:44:26 : Well ! So tell me Vincent's passion :)

B

BOT 🤖 from 🇫🇷 [FR]

12/12/2018 à 11:44:58 : In the spare time of Vincent, he loves to listen to music, watch movies, read books and hang out with friends. He also likes to go out on adventures with friends and family. He loves innovation and is ready for the excited times coming when virtual assistants, AI and IoT become bigger !

J

Jason from 🇫🇷 [FR]

12/12/2018 à 11:46:57 : Hey Jessica ! How are you ?

J

Jessica from 🇫🇷 [FR]

12/12/2018 à 11:48:36 : Fine ! Nice to meet you J ! You're from France, cool 😊

So you come from France too, kinda nice :)

Send

Requirements

Web Framework:

- Spring boot

Languages:

- Java,
- HTML5,
- CSS3,
- Javascript,
- Groovy,
- Sqlite
- XML
- JSON

Build system:

- gradle
- maven (to compile Java code)

Package:

- NLP (Natural Language Processing) with Compromise : <https://github.com/spencermountain/compromise>

Run server

- cd server
- ./gradlew bootRun

Server

- ChatController.java
- Message --> User, Date, Message

Web client to open in web browser

- firefox client/index.html

Client

- index.html : Display the page with the correspondant updated div.
- style.css : Give an intuitive look to the page.
- main.js : Every control is done in this area.
- detect_location.js : Information from users IP.
- Compromise package for NLP.

Integration tests

To run Curl-style integration tests --> you need to install curl, jq then run

- cd test

send a message with the terminal :

- curl -X POST -H "Content-Type: text/plain" http://localhost:8080/api/new-message -d 'Hello there'
- ./new-message "Your message"

Ask server to display ALL messages :

- curl http://localhost:8080/api/messages | jq .
- ./read-messages

You can also run java tests from directory 'server/src/tests/java' by :

- ./gradlew test