Challenge

Below is a specification for a (relatively) simple bot-based chat. This assignment is not meant to consume many hours/days of your life. The specification below is intentionally large, and you are expected to complete both levels. If you didn't manage to finish everything, please tell us which parts you completed.

We don't use any specific framework at the project, that's why we won't ask you to use any, but if you feel more comfortable using one, we would prefer it to be Django. Please let us know if you did use it.

Our intention is not to actually run this code. Some members of our team may attempt to run your program, while others are only interested in visually inspecting it, so don't fret too much over architecture specifics, instead focus on writing well structured code, classes and stuff. However, please take into account that the files and folders structure will be evaluated as that shows how easily the env will be set up and some secret things about you, believe us.

If you don't have any frontend coding ability, that's okay. If you *want* to do some front end coding for this assignment, feel free, but you are not at all expected to. This is an examination of your backend talents as the project is more backend oriented.

To Submit: Please tar/zip up the necessary files and return them to us by email. Thank you for your time and effort!

Model a Financial Chat

- Level 1
    - Have persistent users with profiles.
    - Order the messages by their timestamps
    - Show only the last 50 messages
    - Make the user's chat history accessible from their profile
- Level 2
    - Users can post commands to the chatroom with the following format **/stock=APPL**, and it will trigger a bot that will reply the stock quote.
        - This command won't be saved on the database as a post but it will trigger a RabbitMQ message.
        - A decoupled bot will get the XML information (do not use the json version) from the external API at http://finance.yahoo.com/webservice/v1/symbols/AAPL/quote.
        - The response from the external API must be posted back into the chatroom with the following format: "APPL (apple INC) quote is $93.42 per share". The post owner will be the bot.
        - But there is a little trick here, the API work only for mobile devices, so the User-Agent has to be the one of a mobile device; something like "Mozilla/5.0 (Linux; Android 6.0.1; MotoG3 Build/MPI24.107-55) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.81 Mobile Safari/537.36"
    - Users can post commands to the chatroom with the following format **/day_range=APPL**, and it will trigger a bot that will reply the days-low and days-high stock quote.
        - This command won't be saved on the database as a post but it will trigger a RabbitMQ message.
        - A decoupled bot will get the XML information (do not use the json version) from the external API at http://query.yahooapis.com/v1/public/yql?q=select%20*%20from%20yahoo.finance.quotes%20where%20symbol%20in%20(%22AAPL%22)&env=store://datatables.org/alltableswithkeys
        - The response from the external API must be posted back into the chatroom with the following format: "APPL (apple INC) Days Low quote is $92.11 and Days High quote is $94.15 ". The post owner will be the bot.
- Bonus
    - Add Unit Test for what you feel should be tested.
    - Handle troublemaker RabbitMQ messages (i.e. messages that are not understood or any exceptions raised within the bots)
- Final Notes:

- We do care a lot about our planet and energy efficient stuff and that's why we'll have to ask you to write really lightweight (ro)bots; which means that heavy resource-consumers such as celery won't be allowed to be used.