Rate API

The next step is a coding challenge.

For this test, submitted to the Greenhouse link **AND** GitHub/Bitbucket

The problem we would like you to solve involves writing a **rate-limited API**.

- Given the attached hoteldb.csv file, provide an HTTP service with 2 endpoints:
- `/city`, that returns all the hotels belonging to a specific city
- `/room`, that returns all the hotels that have the requested
- Both the endpoints can have an optional request to sort the hotels by price (ASC or DESC)
- We need to **limit** the usage of the API, by limiting the rate at which the endpoints are called

1.

a. The `/city` endpoint can receive a maximum of 10 requests every 5 second
b. The `/room` endpoint can receive maximum 100 requests every 10 seconds
c. Both these values have to be easily configurable per endpoint (requests and/or periods), and should fallback to a default 50 requests every 10 seconds if no configuration is provided.
d. If the rate gets higher than the threshold on an endpoint, the API should stop responding for 5 seconds on that endpoint ONLY, before allowing other requests.
e. Your rate limiter should be able to support multiple endpoints in future (not just for 2 endpoints above).
f. Please *do not use any external library or key-value store* to implement this rate limit functionality (you are welcomed to use standard one).
g. Provide at least one integration test that calls your API using HTTP.

- Provide a way to demonstrate rate limiting (up to you).

Here some extra guidelines:

- Treat this project as nearly production system (you may think of POC). Point of this exercise is to see how you can contribute to Agoda projects.
- The key component of this exercise is a **rate limiter**, please be focused on this part.
- Provide instructions on how to compile/run your tests and API(s), and provide basic instructions on how to use the API's
- With the submission, please add a small readme file explaining your choices and the decisions you made
- If there are any requirements that are unclear, use your best judgement and provide documentation for your choice
- I would prefer the solution written in either Scala, Java or Python, but also solutions in C#, C++, Haskell (hm, try :) or F# are accepted.

Don't hesitate to send us any questions you might have, and please confirm that you have received this email.