Report for the FRI:DAY code challenge

Tihomir Vachkov

April 18, 2018

1 Instroduction

This report describes my deliveries for the code challenge given to me by the FRI:DAY company. The delivery comprises of this PDF report, a report about the code coverage when the test have been executed (<link>) and a report about the coverage of the tests themselves (<link>).

My system information:

- OS - Windows 10 Pro
- JDK - 1.8.0_144
- IDE - IntelliJ Idea 2017.2.7 (Comunity edition)

2 Getting started

The start of the application requires input parameters for the address and for the execution mode. The address must be specified by `-address`, followed by the address to be parsed, enclosed in double quotes. For example, `-address "Auf der Vogelwiese 23 b"`. The execution mode determines whether we should separate street name and number by pattern matching or by requesting Google. This must be specified by `-mode`, followed by either `regex` or `google`. For example, `-mode google`. No need for double quotes since the option does not contain spaces. The order of the parameter pairs is not relevant.

3 Assumptions

I assumed that I should not write a universal method for all the possible formats of addresses in the world or in Germany, but only for the given cases. Even though, I have add some more cases, which you can see in the tests.

4 Task implementation

During the development, I needed to clear some doubts and in order to do that independently, I would like to explain some details and describe the some considerations. The task itself was clear to me. While planning, I came up with two ideas. One is by pattern matching on the input strings and the other is to find some service or library which does this. While looking for libraries I found a very interesting way to accomplish this - making a request to Google Maps API and parse the returned JSON response for street name and number. I kept both versions for the delivery since one will show my skills for implementing tasks from scratch and pattern matching and the other would verify the existence of the input address.

4.1 Pattern Matching

The basic idea behind the pattern matching is to try to find patterns for the possible street number formats. The rest of the string would be the name of the street. Hence, we have separated correctly the input string into two substrings. I wrote four rules (patterns to match) in a specific order, i.e. the the most restrictive rule goes first and the most general rule goes last. After the pattern

matching finishes, I create an object with the resulting variables of custom type `Address` in order to facilitate testing with an elegant return types and make them easily comprehensible with fewer LOC.

4.2 Requesting Google Maps API

The motivation for this version is that Google Maps finds the correct address even if it is provided in a wrong format and accepts many international formats for address. Another good reason to consider this version is that I can know if the "Winteralle 3" does not exist in reality. I thought this would be very useful in real situations.

There are drawbacks of this method. Let's consider the existing address "Am Berlin Museum 2". If I request it with "strasse" at the end, it will return me a completely different street and number(i.e. the address of the Judische Museum which is "Lindenstrasse"). This means that I have to make some checks on the result. Here the complexity started to grow significantly. For example, if I request for "Warschauerstrasse 23", Google will return correctly number 23 but the street name would be "Warschauer Strasse". The system cannot know if this is correct or we have the previous situation, if we do not introduce some rules for the result check. I found also other limitations and the problem this way goes beyond the requirements of this code challenge because I also aim to check the existence of the input address by having only address and number.

5 Tests

For the testing aspect I used TestNG framework. As mentioned earlier, I have generated code coverage reports for the source code and for the tests. I have completely covered the pattern matching execution and extracting the input parameters, while requesting Google is not tested at all. The implementation of that version was intended to be done if some time remains but then I got into the drawbacks mentioned earlier.

6 Conclusion

Thank you very much for your time! I hope I described sufficiently my delivery. If you have any questions, do not hesitate to contact me - tivachkov@gmail.com.
I am looking forward to hear from you soon.