

## Gopher translator service

Gophers are friendly creatures but it's not that easy to communicate with them. They have their own language and they don't understand English.

Create a program that starts http server. This server should be able to translate English words into words in the gophers' language. Don't worry, the gophers' language is pretty easy.

The language that the gophers speak is a modified version of English and has a few simple rules.

1. If a word starts with a vowel letter, add prefix "g" to the word (ex. apple => gapple)
2. If a word starts with the consonant letters "xr", add the prefix "ge" to the beginning of the word. Such words as "xray" actually sound in the beginning with vowel sound as you pronounce them so a true gopher would say "gexray".
3. If a word starts with a consonant sound, move it to the end of the word and then add "ogo" suffix to the word. Consonant sounds can be made up of multiple consonants, a.k.a. a consonant cluster (e.g. "chair" -> "airchogo").
4. If a word starts with a consonant sound followed by "qu", move it to the end of the word, and then add "ogo" suffix to the word (e.g. "square" -> "aresquogo").

Your program should accept one command line argument "--port" which is the port that the server is running.

Your http server should have the following endpoints:

5. POST "/word" - by given English word, the server should return the word's translation in gopher language. It should accept json data in the format {"english-word":"<a single English word>"} and should return json data in the format {"gopher-word":"<translated version of the given word>"}.
6. (OPTIONAL) POST "/sentence" - by given English sentence (in which each whitespace separated sequence counts as single word) the server should return the sentence translation in gopher language. It should accept json data in the format {"english-sentence":"<sentence of English words>"} and return {"gopher-sentence":"<translated version of the given sentence>"} Assume that every sentence ends with dot, question or exclamation mark.
7. (OPTIONAL) GET "/history" - should return each English word or sentence that was given to the server from the time the server was started along with its translation in gopher language. The output should look like {"history": [{"apple":"gapple"}, {"my":"ymogo"}, ...]} The returned array should be ordered alphabetically ascending by the English word/sentence.

Please don't confuse the gophers as they don't understand shortened versions of words or apostrophes. So don't use words like - "don't", "shouldn't", etc. Even translated they still won't understand you so skip them in your solution.

It is necessary your code to compile.