

Technical Exercise

In this exercise, you will be asked to create a SpringBoot application that provides an auto-complete feature based on a list of words provided, in the fastest way possible.

For this exercise, you can assume that in the runtime environment, there are no physical limitations of RAM, CPU to run the application. The purpose of the application is to return the list of possible names in the fastest method possible.

# Instructions

1. Create a SpringBoot application
2. Load the provided list of names to an H2 database
3. Create datastructure that contains all the names that was previously loaded to DB, the DS will be filled when app is loaded, the DS should support the ability to find the name in O(K) where k is the length of the name char after char, for example: b, ba, ban, bana, banan, banana(**O(6)**).
4. Implement a REST controller with a method receiving text (name prefix) as an argument (you can choose where the text in the request will be) and returns a list of auto completed names matching the given prefix in **O(K)** using the DS .

# Notes

* The file should only be used for loading the list to the H2 database. The application should not be aware of the file's existence at all.
* The output of the exercise should be a runnable mavenized eclipse project uploaded to github or sent as a zip
* The project should be packed as docker image with port 9090 using maven
* You should be able to run the app from eclipse and from command(docker container) as well
* When reviewing the exercise, we will focus on project structure, design, API standards, input validations, error handling, maven build part, application efficiency and code styling.
* Bonus: make the application case insensitive, but still returning the values in their original case.