

Kry/Livi Code Assignment

Summary

As a part of scaling the number of services running within a modern health tech company we need a way to make sure our systems are running smoothly. None of the monitoring tools that we have looked at satisfy our requirements so we have decided that we will build one ourselves. What we want you to do is to build a simple service poller that keeps a list of services (defined by a URL), and periodically performs a HTTP GET request to each and stores a record of the response ("OK" or "FAIL"). Apart from the polling logic we want to have all the services visualised and easily managed in a basic UI presenting all the services together with their status.

Requirements

Basic requirements (If these aren't met the assignment will not pass):

- A user needs to be able to add a new service with URL and a name
- Added services have to be kept when the server is restarted
- Display the name, url, creation time and status for each service
- Provide a README in english with instructions on how to run the application

Extra requirements (No prioritisation on these, pick the ones that you find interesting):

- We want full create/update/delete functionality for services
- The results from the poller are automatically shown to the user (no need to reload the page to see results)
- We want to have informative and nice looking animations on add/remove services
- The service properly handles concurrent writes
- Protect the poller from misbehaving services (for example answering really slowly)
- URL Validation ("sdgf" is probably not a valid service)
- Multi user support. Users should not see the services added by another user



Constraints

This task is intentionally open-ended with no “boilerplate” code provided. It’s up to you to implement the solution using frameworks that you deem suitable for this task. The solution should have a backend which the user accesses over some sort of network, and you should use Java as the base language for the backend solution. We also ask you to use an SQL-based database. Apart from that you’re free to use whatever suits you best in performing the task. Be prepared to be able to explain the reasoning behind why you choose a particular solution.

If you want to keep with our tech stack, it’s based on [vertx](#), [rxjava](#), [react](#) and [mysql](#) and builds with [gradle](#).

If you decide to use MySQL as a database, we ask you to:

- Configure it to run on port 3309.
- With credentials
 - User: dev
 - Password: secret

We have provided a provided a GitHub repository which can help you set this up using docker-compose at <https://github.com/webbhalsa/code-assignment-tools>

If you however decide to use something else which needs a running server, we ask you to provide instructions on how to set it up using Docker with the expected database name and credentials.

Evaluation

The assignment will be evaluated on these following areas:

- Code structure
- Testability
- Scalability
- Clean code
- Usage of data structures
- Api design

We don't want you to spend too much of your time on this assignment. We expect you to spend approximately **four hours** working on it and make sure to finish the issues you start.

Submission

Put the code together with instructions on how to run it in a git repo on GitHub and send us the link to techcase@kry.se when you are done.

Good luck!

