**Recruitment Assignment**

for

**Java Developer**

presented by



**Date:** 2018-06-19

**Revision:** 3.0

Welcome recruit!

Firstly, congratulations! The fact that you have received this assignment, means that you have impressed the (quite demanding) iOCO interview panel sufficiently to justify entry into stage 2 of the recruitment process.

This assignment is designed to test your competence as a senior developer. Assuming you have the relevant skills, you should be able to complete it in less than 8 hours (perhaps 4 hours, but it really depends on your experience level).

The assignment assumes you have working experience with a number of technologies that iOCO uses on a daily basis. Perhaps you’ll need to sharpen your skills in one or more, but ideally you should be able complete the task without a large amount of Googling.

The specific technologies involved are:

* Git
* Apache Maven
* Spring Boot

For this assignment you will build a RESTful web service, deployed as a Spring Boot application. The web service will store information into a super-basic database. You must use the H2 in-memory database provided by Spring Boot.

Although the functionality will be very basic, the fact that your solution touches multiple enterprise software layers will make your recruiters confident that you can hit the battleground all guns blazing. Please note, while this assignment focusses on back-end capabilities, the typical developer will often cross between front and back end.

In addition to testing your Java skills, this assignment will also use Apache Maven and Git to confirm that you are familiar with these pervasive tools in enterprise development.

The rest of this document contains the requirements for the assignment. Please complete in your own time and share the final GitHub public repository with your recruiter.

iOCO will review your submission and use the results to make a decision about an invitation to the final in-person interview.

Good luck.

# Business Description

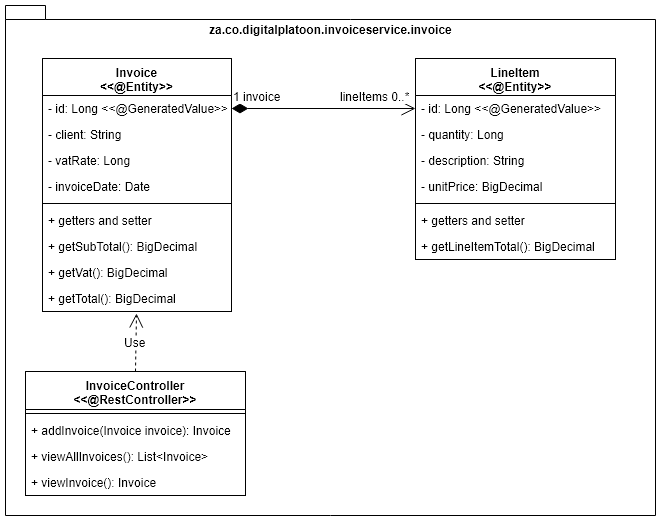
After deployment, your application will expose a single RESTful web service with the following endpoints:

* Add Invoice 🡪 POST http://localhost:8080/invoices
* View All invoices 🡪 GET http://localhost:8080/invoices
* View Invoice 🡪 GET http://localhost:8080/invoices/{invoiceId}

**Your next question may be: “What is the purpose of this web service”**

This service will be used to keep a centralised register of invoices sent to clients.

The following domain model gives a UML view of the classes (and consequently, tables) that should form part of your solution:



The UML class diagram above contains 2 JPA entity classes and a Spring RestController class. Additional classes will be required by your final solution, but we strongly suggest that you stick to the design of these classes.

All Java BigDecimal calculations should be performed using a scale of 2 and a rounding mode of HALF\_UP.

# Evaluation Process

When you are completed with the development of your solution, please ensure you check your code into your own GitHub repository (all developers should have one). Send us the repo URL.

We will clone the project, review your code and try to build using Maven and run it locally.

Most importantly will be that you code runs without any intervention from the person marking your assignment.

We will evaluate very strictly whether you are able to develop from a spec even if the spec is not perfect. This advice goes for both the provided endpoint URLs and the class design.

When your application starts up, it should create a standard Spring Boot H2 in-memory database with the specified tables. It is not required to pre-load the H2 in- memory database with any data. If you prefer, you may optionally provide an H2 console URL so that we can inspect the database while evaluating your assignment.

If time permits, you may optionally provide some test classes to test the system. You will **NOT** be failed if you do not provide test classes, but this is a good way to differentiate yourself from the competition.

Besides the guidance provided above, please feel free to build the service as you please.

And once again, good luck!