Applicant Test Task: Design of a REST API

Introduction

The goal of this exercise is to give you the opportunity to build a very small scale web application. You should not invest more than 4-6 hours into this exercise.

> We do not expect you to fully complete this project. We have all done this by ourselves, and so far nobody has "finished" the task in the provided 4-6 hours. We are mostly interested in a discussion base, seeing how you set up a project, what technical choices you make and what areas you decide to focus on.
>
> Its perfectly fine to focus on the area you feel most productive on first and to make decisions on what not to complete to get as much done as possible in the provided time.
>
> If you get completely stuck somewhere, feel free to focus on a different part of the application or ask us about clarification (You can find our contact details in the email we sent you the task with).

Technical requirements for the application

- The application should work and be implemented in Java. You are free to choose any fitting Java based framework. Preferably Spring, RESTEasy orsimiliar JavaEE REST framework in combination with Hibernate, EclipseLink, etc. for persisting data.

Functional requirements for the application:

The goal is to develop a very simple REST API for an issue tracker of a small team of developers. The requirements of the team are basic:

- There should be a list of developers. Functionality to add or remove developers. A developer just has a name as the only attribute.
- The application should be able to handle two different kinds of issues: Stories and Bugs
  - Both Types have an issue-ID, as well as a title, a description a creation date and an assigned developer
    - ussue-ID must be unique
  - Stories also have an estimated point value, as well as a status that can switch between "New", "Estimated" and "Completed"
  - Bugs have a priority ("Critical", "Major" or "Minor") as well as a status ("New", "Verified" or "Resolved")
- The team wants some help with planning their stories. The goal is to know what stories will be worked on in which week.
  - The team has learned that on average one developer can complete 10 story points a week
  - The goal is to complete all the stories in the minimum amount of weeks.
  - It is not necessary to find a truly optimal solution, but the solution should not have any glaring flaws.
  - The total amount of story points in a week should not exceed <developer count> * 10. It is not important to consider which developer a task is assigned to.
    - stories are not assigned beforehand
    - assignment is done in the planning
  - No other constraints on the ordering of the stories exists.
  - Bugs are completely ignored in calculation.
  - As long as no new stories are created, the distribution should remain the same.

If you would like to create a user interface for managing developers, stories, bug and showing the plan it would be a plus 🙂

What you should send to us:

- The source code of your project.
- A short explanation on how we can build and run your project.
- How to access the application (the URL)
- A short description on the technical choices you made and why you made them. If you had problems completing the project, a short description of the problems you faced and how you would continue to resolve them if you had more time.