Trading REST API Project – Sprint 1

Introduction

During the program you will be creating an application that will track a portfolio of stocks. The application will allow the users to save and retrieve records to a MySQL database. These records will describe orders that are being placed for specific Stock trades.

The final application will be made up of a front-end, a back-end, and a database, and will be deployed to a server.

Overview

The aim of this first hackathon is:

- Project team members to get to know each other, and identify your strengths and core areas of knowledge
- Set up a project in Jira to document requirements and track their completion
- Set up a workspace in Bitbucket to host the repositories you will create
- Define the data model – what data will your application store? What tables are required? What fields are needed in these tables? Note that this can be revised as you develop your projects.
- Create a MySQL database structure with some initial dummy data populated.
- Create a script to create the MySQL database and store this in a Bitbucket Repository

Notes

1. There will be no authentication, and a single user is assumed, i.e. Do NOT manage users.

Technical Getting Started Checklist

1. To create and populate the database, one participant should share their screen, but all should agree and discuss what should be done together.
2. Once created, use the MySQL tools to export the database, and then you can use git to put this script into Bitbucket.
3. All participants can then clone the database script and create their own copies on their own VMs.

4. Decide on the absolute MINIMUM fields for the data model for the first working system. An example of the data that the first working version might work with is:

- **id:** the id auto-generated by the database for this record
- **stockTicker:** the "ticker" for a stock e.g. AMZN, AAPL, C, NFLX
- **price:** the price in USD per stock that this order is requesting to be bought or sold
- **volume:** the quantity of stocks that this order is requesting to be bought or sold
- **buyOrSell:** a flag indicating whether this is a request to BUY or SELL (a future enhancement might be to use an Enum here)
- **statusCode:** an integer code indicating the status of this order. For the initial system this will be set to 0. We will enhance this in future weeks so that another component may update this status (e.g. a future enhancement may use this to indicate: 0=Pending, 1=Filled, 2=Rejected...)

Project Management Getting Started Checklist

1. As a team, decide how you will approach the work. E.g. one person on data model / MySQL schema, two people on Java REST Interface.
2. Make a task list. Use Jira to keep track of tasks. Consider the tasks outlined above and break them into smaller mini-sprints.
3. Work in an agile manner; choose the tasks required for a MINIMAL WORKING implementation first.
4. Your instructor will come to each team early on to check on your status. Make a note of any questions you have for your instructor, so you're ready with all of your questions when your instructor arrives.

Suggestions for Success

1. START SMALL. Get a system working that stores a very simple object with minimal fields. You can then enhance to store more complex records.
2. Try pair programming; it can be very effective.
3. Take deliberate steps to keep good energy levels in the team. E.g. give your team a name, and systematically plan check-ins with each other.
4. Emphasise code quality over quantity.

Appendix : Notes on Teamwork

Try to work closely as a team during this project. (E.g. Pair Programming is a great way to learn http://www.extremeprogramming.org/rules/pair.html)

Your team should be self-organising but should raise issues with instructors if they are potential blockers to progress.

Your team should use a task management system such as Trello or Jira to keep track of tasks and progress and divide the work.

Your team should keep track of all source code with git and Bitbucket.

Try to use a sensible git branching strategy, and make use of Pull Requests as code reviews. However, don't let this become a blocker to your progress. If git is slowing you down, then go with a simpler strategy.

Your instructor and team members need to access all repositories, so they should be either A) made public OR B) shared with your instructor and all team members.

Throughout your work, you should ensure good communication and organise regular check-ins with the rest of your team.

Most important - remember this is training, so HAVE FUN!!!