Developer Project

Back-end application in Spring Boot,
using a MySQL database
and exposing a REST front-end

Abstract

The goal of this project is to reinforce the technical skills taught during the training programme, and in particular to explore the application of computer technology in the development of a business application.

During this project, you will be required to perform the design, implementation and testing necessary to build a representative application for use within Allstate using a typical technology stack.

Objectives

Successful completion of the project will require:

- The development of a complete back-end application, using Spring Boot, which will:
  - Provide a REST API for clients to connect to
  - Store and retrieve data from a database
- You will also prepare a presentation of your project, explaining how it works, together with a demonstration. This will be presented to peers, instructors and existing senior staff

Project

The **minimum** requirement for this project is to construct a back-end system to allow the organisation to maintain information relating to an business domain of your choosing. In particular:

- There must be a minimum of 2 entities within the system, and there should be at least 1 relationship. For example, you could choose Customer and Invoice as your entities, with each customer having many invoices.
- Having chosen your entities, you will determine an appropriate set of attributes for each entity. Do not try to model every aspect of your entity in full – between 6 and 10 attributes at most will be appropriate.
- The system must be able to store its data in a MySQL database. It is your choice whether you create the database tables manually or use an ORM tool such as JPA to do it for you.
- The system must contain at least 1 piece of custom business logic, and you should implement appropriate tests to demonstrate the logic is working as expected.

- The system must provide a range of functions via a REST api for clients to access. The exact functionality you choose to offer will depend on your business domain but the following will be an example for Customers with Invoices:
    - Retrieve a list of all Customers (but do not include their invoices in the returned data)
    - Retrieve a single instance of a customer given their unique ID (but do not include their invoices in the returned data)
    - Retrieve a list of all invoices for a customer given the customer's unique ID.
    - Retrieve a list of all customers who are in a particular state.
    - Retrieve a list of all customers with invoices totalling $1000 or more (this could be the business logic requirement).
    - Add a new customer to the system
    - Add a new invoice to a customer
    - Update a customer
    - Update an invoice
    - Delete an invoice
    - Delete a customer
- The design of the API must follow the Rest standards
- You should be able to demonstrate that the REST API is working by using either the CURL command line tool, or via the provided HTML Rest Testing web page.
- When you demonstrate the application running, this must be run from the command line, not from within the IDE. You should expect to also demonstrate that your tests are passing.

Suggested Approach

There will be a number of steps you will need to undertake to complete this project. The following is a suggested order.

You will need to undertake the work using your Virtual Machine – don't forget that you can control when your virtual machine is switched on or off by visiting https://student.conygre.com/ . Please remember to switch off your virtual machine when you have finished using it, if you start it outside of class hours!

- Create a Spring Boot project using the Initializr – choosing the dependencies needed to be able implement all the requirements.
- Create a database on the MySQL database server for the project to use.
- Configure the project, so that it will be able to connect to the database

- Create one of the entities and configure it as needed. Don't worry about having all the fields – one or two is fine for this stage.
- Create a DAO or a repository for the entity - do not add any methods at this stage
- Create a service interface and implementation - do not add any methods at this stage.
- Create a controller, and provide a simple endpoint (such as a healthcheck) that returns the word "ok"
- Run the application and check that a database table has been created to match the entity. Also check that if you access the endpoint you created, you can see the word "ok".
- Add the extra required attributes to the first entity, and create any others needed for your project.

From this point on, take an iterative approach to the project, completing it feature by feature. For example you might start by building the REST endpoint that will return a list of one of your entities. You might create the controller method first, then the service method, and then any required repository or DAO methods. Complete this activity so that it works correctly, and then go onto build the next feature.

You may wish to add 1 or 2 records to the database manually so that you can test your code.

Integration notes

We expect that another team will be working to build a front-end application that will connect to your system via REST (actually you will do that next week!). For now, you will not be able to see this application, and so are working on this project in isolation. As a result, you must ensure that the design of your API meets the generally accepted REST standards.

You can test your application by using the provided HTML Rest Testing web page which can be accessed here: https://github.com/vppmatt/curl-generator

Project Review

Please ensure that your project code has been pushed to a publicly accessible repository on Github or an equivalent, so that a review of your code can be undertaken – this will be done after your presentation.

Project Review Criteria

By way of a checklist, when instructors and others review your project, they will be looking for evidence of the following:

- A demonstration of a fully working system, showing that all the minimum requirements have been met (and the additional optional requirements if attempted)
- Good use of the Rest standards for API design
- The ability to explain code choices you have made.

This is an team project

You will be working with a small team of colleagues on this project. You should use pair-programming techniques during the build process, and we encourage you to discuss options and compare notes throughout the development process. Please ensure that you take it in turns to be the "driver" and that everyone plays a part in the presentation / demonstration.

Support Resources

To help you develop this project, you may wish to make use of some or all of the following resources:

- The provided course slides and notes
- The recordings of the training sessions
- The sample code created during the course and available via the GitHub repository

Good luck!