Documentation

Rock, Paper, Scissors Challenge

Project developed by: Juan Pablo Mora Jiménez

January, 2017

1. Code sharing

The code for this application is available in GitHub. The URL is given below:

https://github.com/jpmoraj/rockPaperScissorsChallenge

Please enter and download the code for its revision.

2. Technologies employed

To develop the solution for this challenge, the next technologies were used:

- Eclipse Mars as Integrated Development Kit
- Maven
- Apache Tomcat, version 7.0.73
- HSQLDB (Embedded database, http://hsqldb.org/)
- Spring Framework for dependencies Injection and RESTful implementation.
- Spring MVC to manage the UI section
- JSP
- Javascript Libraries: JQuery
- CSS, Javascript

I chose Spring Framework to implement RESTful Services, because Java is the language that I like more, and the one I have more experience as well. This was done in this way because the time available to finish the solution was a bit short. In other case, I would use something like MVC .net to learn a different “flavor”.

Spring was chosen to implement UI. Because it was necessary to implement REST methods, I considered Spring MVC was a perfect complement to develop the UI logic.

I also used HSQLDB to implement database scripts and to manage the information entered by user. It is an embedded data base, very useful to spend less time on installations or configurations. And this technology is ideal for small applications like this.

Apache Tomcat was used because it offers excellent conditions to manage Spring features and capabilities, it supports dependencies injection, furthermore, this is not a very heavy web application server, very known for Java Developers.

3. Rest API documentation

We have here three different methods, in the class "ChampionshipRestController". The explanation is coming here:

a) http://jpmoraj.j.facilcloud.com/api/championship/result

Description: Stores the first and second places of a tournament.

HTTP Method: POST

Response Format: JSON

Parameters: first (name of the first place), second (name of the second place), both are text.

Output: A JSON text that indicates the operation succeeded: Example: {status: success}

b) http://jpmoraj.j.facilcloud.com/api/championship/new

Description: Takes a text, containing a tournament, included in a bracketed array, and finds out and returns the winner.

HTTP Method: GET

Response Format: JSON

Parameters: data (A text following a bracketed array standard. Example: {data= [Juan", "P"], ["Pedro", "S"]})

Output: A list of strings, with the winner's data. Example: {winner: ["Marta", "Angela"]}

c) http://jpmoraj.j.facilcloud.com/api/championship/top

Description: Retrieves the first 10 players which have get the highest amount of points in all the championship.

HTTP Method: GET

Response Format: JSON

Parameters: count (an integer, optional). Its default value is 10.

Output: A list of strings, something like {players: ["Juan", "Luis"]}

4. About the author

Juan Pablo Mora is a Costa Rican software engineer, who has been working in different projects for private Banks in Costa Rica, since 2006. His experience is based mainly on working with J2EE Technologies.