Télécom Saint-Étienne
Web Services
Class Project

Remarks

- You can form a group to at most three students.
- DEADLINE: 01/12/2016 À 23:59h

Objectives

- Understanding the Representational State Transfer (REST).
- Design of a resource-oriented architecture.

1 Functional Requirements

In this project, you will design a *social meetings* web service, which provides a small subsets of the funtionalities of Meetup. That is, a social networking portal that facilitates the group meetings between interested parties.

You designed system will support the following entities:

- Users, that have the following attributes:
  - email address (also serve as a unique ID)
  - first and last name
  - brief biography
  - groups, the user is a member of
- Groups, which have the following attributes:
  - unique name
  - description
  - user who is the admin of the group
  - group members
  - discussion board

The system will support the following *scenarios*:

- user sign up
- user changes his/her full name and/or biography
- user deletes his/her account
- user creates new group
- user changes the description of the group he/she owns
- user deletes groups he/she owns
- user views a list of groups with descriptions and membership count
- user joins group
- user views members of the group he/she owns or has joined
- user comments on the dashboard of the group

– user leaves the group
– user checks the profile of other users

These entities will be exposed as addressable, interconnected resources with suitable representations, and the scenarios will be supported through the uniform HTTP request interface (as discussed in the class CM3).

In general, you should follow the architectural and design principles discussed in the class (CM3, see lecture notes for more details) and the RESTful web service book.

You system should support at least following two representattions of the user and group resoruces:
– HTML/XHTML (for human use)
– JSON (for programmatic clients)
The key conceptual component of this project is to map the uniform REST verbs to the functionality described in the requirements. Some of these concepts will be discussed in the coming TD labs.

2 Non-functional Requirements

In addition, the following nonfunctional requirements apply to this project:

– The project should be managed using Maven.
– Restlet is used to expose the given functionality in a resource-oriented form, where appropriate, suitable representations should be included that make the (read-only) functionality accessible through a web browser (XHTML). Otherwise, the functionality should be accessible through cURL and through a suitable client also implemented in Restlet (JSON).
– There should be a comprehensive test suite using JUnit, and HTTP-client-based system-level testing of the service using the JSON representation.
– Persistence (i.e. storing user profiles, groups) can be provided using mongoDB (I'll prefer this and its easier as well) or any other form of SQL/noSQL.
– The deployment and the dependencies among the components can be managed used Spring framework or simply NodeJs and express (see tutorial on this link).

3 Deliverables

For each project, I will personally chair a small session, where you will present your project in detail. I will ask question from each group and the grading will depend on the quality of work and the lesson learned during your project implementation. In general, the deliverables are as follow:

– Working code in a repository (GitHub or Bitbucket)
– Also in the repository, detailed documentation of your resource-oriented architecture, including
  – a list of resources with a brief description of each for each resource, the supported HTTP methods and their meaning/effect on the resource state (in terms of the scenarios listed above).

Please use the repository within your group for collaboration and before deadline you will share your repository with me. No explicit submission will be required and submission by email will not be accepted.