PostBeyond
111 Richmond St
Toronto, ON. M5H3K6
(888) 948-4969

User management API

OVERVIEW

Purpose of this project is to showcase abilities to design and build REST API service using Flask web application framework and SQLAlchemy ORM.

GOALS

1. Project must be portable i.e there should be an automated way to build and run project on local machine after cloning the source. (hint: virtualenv)
2. Use current standards and best practices as much possible

SPECIFICATIONS

User management service is responsible for managing users, groups and relationships between them. It has to resemble modern REST API techniques and recommended ways of building APIs. Views are completely optional (i.e Front end of the project is not required).

REQUIREMENTS

Resources

API service must expose endpoints to work with **two** resources:

- User
- Group

User resource

This resource represents a particular user of the system. User can be part of one or more groups. Each instance of this resource has to have at least these properties:

- Name

- Email

Group resource

Each instance of this resource has to have at least these properties:

- Name
- Date created

Use cases

API must support following use cases:

- Adding new user
- Editing existing user
- Deleting existing user
- Adding new group
- Editing existing group
- Deleting group (! please address the best way of handling deletion of the group that has users that are part of it)
- Adding existing user to an existing group
- Removing user from the group
- Listing all users - sorted in an alphabetical order by name
- Listing all groups - sorted in an alphabetical order by name
- Listing all groups of a particular user
- Listing all users of a particular group
- List of users and a number of groups that users belong to - sorted by the number of groups in ascending order
- List of groups and a number of users belonging to each group - sorted by the number of users in descending order

Optional use case:

- List all users and the names of the groups that user belongs to - users are sorted by name in descending order

Data format

API should work exclusively with JSON and explicitly communication this header

- Content-type: application/json

API should support two formats of the same data:

- Pretty print
- Compact mode