Implementing API to fetch all ratings by a user on different Skills

SUSI Skill CMS allows the users to rate any Skill on a scale of 1 to 5. The user can also provide a feedback to any Skill. This paves the path to implementing a Dashboard, which has all the analytic data of the user. Hence, an API needed to be implemented which could return the ratings done by a particular user on all different Skills.

How Servlets are implemented in SUSI.AI ?

All servlets in SUSI extend AbstractAPIHandler class and implement APIHandler. All servlets have 4 methods, which we overwrite depending on what we want the servlet to do. They are as follows :

```
@Override
    public String getAPIPath() {
        return null;
    }

@Override
    public BaseUserRole getMinimalBaseUserRole() {
        return null;
    }

@Override
    public JSONObject getDefaultPermissions(BaseUserRole baseUserRole) {
        return null;
    }

@Override
    public ServiceResponse serviceImpl(Query post, HttpServletResponse response,
```

```
Authorization rights, JsonObjectWithDefault permissions) throws APIException {
        return null;
    }
```

How these 4 methods work together ?

- ➢ First method is getAPIPath(). It returns the endpoint of the servlet.
- ➢ The second method is getMinimalBaseUserRole(). It returns the minimum privilege level required to access the endpoint.
- ➢ The third method is getDefaultPermissions(). It gets the Default Permissions of a UserRole in SUSI Server. Different UserRoles have different permissions defined in SUSI Server.
- ➢ Whenever the endpoint defined in the getAPIPath() method is called properly, it responds with whatever is defined in the fourth method, which is serviceImpl().

Implementing a servlet to fetch all ratings by a user on different Skills

The task of this servlet is to fetch all the ratings done by a user on all the different Skills, so that this fetched data could be used later on for implementation of various user specific features like Dashboard page.

This is the implementation of the 4 methods of this servlet:

```
    @Override
    public UserRole getMinimalUserRole() {
        return UserRole.USER;
    }

    @Override
    public JSONObject getDefaultPermissions(UserRole baseUserRole) {
        return null;
    }

    @Override
    public String getAPIPath() {
        return "/cms/getProfileDetails.json";
    }

    @Override
    public ServiceResp onse serviceImpl(Query query, HttpServletResponse
response, Authorization authorization, final JsonObjectWithDefault permissions)
throws APIException {

    JsonTray fiveStarSkillRating = DAO.fiveStarSkillRating;
    // JSONObject and JsonArray Declarations
```

```
    // Checking if access_token has been given and is valid
    if (authorization.getIdentity() == null) {
        throw new APIException(400, "Specified user data not found, ensure you
are logged in");
    }

    // Fetching email of the user from access token and storing it in a string
    String email = authorization.getIdentity().getName();

    // Iterating over the fiveStarSkillRating JsonTray by extracting keys at
every level and accessing their child objects through the extracted keys
    for(String model_name : fiveStarSkillRating.keys())
    {
        // Storing the list of group names and iterating over them

        for(String group_name : groupnameKeysList)
        {
            // Storing the list of language names and iterating over them

            for(String language_name : languagenameKeysList)
            {
                // Storing the list of skill names and iterating over them

                for(String skill_name : skillnamesKeysList)
                {
                    skillnameArray = languageObject.getJSONArray(skill_name);

                    for(int i=0; i<skillnameArray.length(); i++) {
                        String jsonEmail =
skillnameArray.getJSONObject(i).get("email").toString();
                        if(jsonEmail.equals(email)) {
                            JSONObject userSkillData = new JSONObject();
                            JSONObject userSkillRatings = new JSONObject();
                            int stars =
Integer.parseInt(skillnameArray.getJSONObject(i).get("stars").toString());
                            String timestamp =
skillnameArray.getJSONObject(i).get("timestamp").toString();
                            userSkillData.put("stars",stars);
                            userSkillData.put("timestamp",timestamp);
                            userSkillRatings.put(skill_name,userSkillData);
                            if(result.has(jsonEmail)) {
                                skillRating = result.getJSONArray(jsonEmail);
                            }
                            skillRating.put(userSkillRatings);
                            result.put("rated_skills", skillRating);
                        }
                    }
                }
            }
```

```
        }
    }

    // Handling case when user hasn't rated any Skill yet
    if(result.length()==0) {
        result.put("accepted", false);
        result.put("message", "User has not rated any Skills yet.");
        return new ServiceResponse(result);
    }

    result.put("accepted", true);
    result.put("message", "User ratings fetched.");
    return new ServiceResponse(result);

    }
```

As it can be seen from the above code, the endpoint for this servlet is /cms/getProfileDetails.json and it requires 1 parameter - the access token of the user.

As the main task of this servlet is user specific, and should only be accessible to the particular user, hence we returned UserRole as USER in the getMinimalUserRole() method.

In the serviceImpl() method, we iterate over the fiveStarSkillRating.json JsonTray and keep on extracting keys and accessing the corresponding JSONObjects until we reach the lowermost layer, where skill names are listed. Iterating over the JSONObjects corresponding to each skill name, we check the email to identify the user. If the 2 emails match, we extract rating and the timestamp for the skill and put it in the result JSONObject.

This is how we can fetch all the ratings by a user on all the different Skills.

Resources

- Iterating over JSONObject in Java
- Adding data to JSONObject in Java
- JSONArrays in Java

Tags:

FOSSASIA, SUSI.AI, Tutorial, GSoC, Skills, API, Servlet, Rating