**REALCALLER REST API**

This is the design document and user-guide document for the REST API developed for this assignment. This API provides the capability to record the user views of a profile and also view the latest views for a particular profile.

**Designed by:**

**Priyajot Syan**

**System Requirements:**

* Maven 3.2 installed.
* Java 1.8

1. **Database Schema:**

I have used 3 tables in this schema:

1. **ProfileData**: This table stores the data of the User Profile. It has profileId, firstName, lastName and country as the columns.
2. **ProfileView**: This table stores the views when one user visits the profile of another user. It has obejctId, userProfileId, viewerProfileId, viewDate as the columns. This table constantly moves the unwanted data to the backup table so that the live table is kept with minimal data.
3. **ProfileView\_Backup**: This table also stores the views when one user visits the profile of another user. It has obejctId, userProfileId, viewerProfileId, viewDate as the columns. It is used to backup the data from the live table**.**

An index “**index\_profile\_view**” has been created on the **profileview** table for the columns userProfileId and viewDate to allow faster and improved performance of the application when querying the views data using userProfileId.

A trigger “**TRG\_PROFILEVIEW**” is created on the **profileview** table which backups the data to **profileview\_backup** table helping in the removal of unnecessary data from the live table

1. **Do you delete any data from the database? If yes, When? Why? If no, Why?**

No, I don’t delete any data from the database. But what I do is copy data from   
“**profileview**” the main live table and move it to the “**profileview\_backup**” archive or backup table. That will keep the data in my live table to a minimum and make the queries faster. The reason for not deleting the data here is that if in case we might need that information in future use but still that depends on the marketing and other aspects of the application if the required data is too sensitive or due to certain laws then we would have to delete it.

1. **Do you have any periodic task type of batch jobs to maintain data? If yes, Why needed? If no, Why not needed?**

Yes, I do have periodic jobs to delete data from live table to maintain the data so that live table will keep as relevant data as possible. I run the batch job automatically every night at 11:00 pm which deletes the data which is older than 10days significantly reducing the number of records from the table and making processing faster and more efficient.

1. **What type of compromise (for example; tradeoff between storage on disk vs latency) did you see? What was your decision? Why?**

Trade-off is that creating an extra table “**profileview\_backup**” will increase the storage space required on the disk but it will reduce the data latency considerably when processing the “**profileview**” table. So taking into account the whole picture of the operation as the disk storage is cheap, is that this trade-off will considerably increase the performance with faster processing.

1. **Steps to run the Application**
2. **To run the database server**

java -cp lib\hsqldb.jar org.hsqldb.Server

1. **To build and package the application**

mvn package

During build itself, 5 unit test cases are run for the Project. Build will fail

If there is an issue with the Resource.

1. **To migrate and setup the database on the application**

java -jar target\Realcaller-0.0.1.jar db migrate config.yaml

1. **To run the application**

java –jar target\Realcaller-0.0.1.jar server config.yaml

## Service API Specification

The list of API functions supported is shown below.

### Get All Existing User Profile’s List

Gets the profile Id, the first Name, the last Name and the country of User’s Profile.

|  |  |
| --- | --- |
| Resource | Realcaller Root |
| Function | Get the List of existing User Profiles |
| URI | http://localhost:8080/users |
| Method | GET |
| Arg | None |
| Success Response | 200 OK JSON data: Profile data |
| Failure Response | 500 Internal Server Error |

Sample call using Curl:

|  |
| --- |
| **curl -H "Content-Type: application/json" -X GET http://localhost:8080/users** |

Sample Response Data:

|  |
| --- |
| **[{**  **"profileId":1,**  **"firstName":"Sachin",**  **"lastName":"Tendulkar",**  **"country":"Russia"**  **},**  **{**  **"profileId":2,**  **"firstName":"Rahul",**  **"lastName":"Singh" "country":"India"**  **}]** |

### Create Profile Data

Create/add a profile. A post is done to the Profile Data Table.

|  |  |
| --- | --- |
| Resource | Realcaller Root |
| Function | Create a Profile with data. |
| URI | http://localhost:8080/users |
| Method | POST |
| Arg | Profile data in the post body |
| Success Response | 200 OK  response data: - the created profile data |
| Failure Response | 500 Internal Error 400 Bad Request |

Sample call using Curl:

|  |
| --- |
| **curl -H "Content-Type: application/json" -X POST -d '{"firstName":"Sachin","lastName":"Tendulkar","country":"India"}' http://localhost:8080/users** |

Sample JSON response data:

|  |
| --- |
| **Response status = 200**  **Created resource –**  **{**  **"profileId":8,**  **"firstName":"Sachin",**  **"lastName":"Tendulkar",**  **"country":"India"**  **}** |

### Get requested User Profile

Gets the profile Id, the first Name, the last Name and the country of the requested UserId’s Profile, if it exists.

|  |  |
| --- | --- |
| Resource | users |
| Function | Get the requested userProfile |
| URI | http://localhost:8080/users/{userProfileId} |
| Method | GET |
| Arg | userProfileId |
| Success Response | 200 OK JSON data: Profile data |
| Failure Response | 500 Internal Server Error |

Sample call using Curl:

|  |
| --- |
| **curl -H "Content-Type: application/json" -X GET http://localhost:8080/users/1** |

Sample JSON response data:

|  |
| --- |
| **Response status = 200**  **{**  **"profileId":1,**  **"firstName":"Sachin",**  **"lastName":"Tendulkar",**  **"country":"Russia"**  **}** |

### Record Profile View

When one user views another user’s profile. It records the UserId, ViewerId, Timestamp of the visit of profile.

|  |  |
| --- | --- |
| Resource | users |
| Function | Log profile view of one user by another user |
| URI | http://localhost:8080/users/{userProfileId}/views/{viewerProfileId} |
| Method | GET |
| Arg | userProfileId, viewerProfileId |
| Success Response | 200 OK JSON data: User Profile View |
| Failure Response | 500 Internal Server Error  400 Bad Request |

Sample call using Curl:

|  |
| --- |
| **curl -H "Content-Type: application/json" -X GET http://localhost:8080/users/3/views/2** |

Sample Response Data:

|  |
| --- |
| **{**  **"userProfileId":3,**  **"viewerProfileId":2,**  **"viewDate":"2015-07-27 18:25:47 PM",**  **"objectId":4**  **}** |

### Latest Profile Views for Requested User

Gets the latest 10 visits to a User’s Profile consisting of UserId, ViewerId, Timestamp of the visit of profile.

|  |  |
| --- | --- |
| Resource | users |
| Function | Get the List of existing User Profiles |
| URI | http://localhost:8080/users/views/{userProfileId} |
| Method | GET |
| Arg | userProfileId, viewerProfileId |
| Success Response | 200 OK JSON data: User Profile View Data |
| Failure Response | 500 Internal Server Error  400 Bad Request |

Sample call using Curl:

|  |
| --- |
| **curl -H "Content-Type: application/json" -X GET http://localhost:8080/users/views/4** |

Sample Response Data:

|  |
| --- |
| **Response status = 200**  **[**  **{"userProfileId":4,"viewerProfileId":2,"viewDate":"2015-07-27 18:56:25 PM","objectId":17},**  **{"userProfileId":4,"viewerProfileId":1,"viewDate":"2015-07-27 18:56:18 PM","objectId":16},**  **{"userProfileId":4,"viewerProfileId":1,"viewDate":"2015-07-27 18:48:23 PM","objectId":15}**  **]** |

### Latest Profile Views for all existing Users

Gets all visits for all User Profiles consisting of UserId, ViewerId, Timestamp of the visit of profiles.

|  |  |
| --- | --- |
| Resource | users |
| Function | Get the List of views for all profiles |
| URI | http://localhost:8080/users/views |
| Method | GET |
| Arg | No Args |
| Success Response | 200 OK JSON data: User Profile View |
| Failure Response | 500 Internal Server Error  400 Bad Request |

Sample call using Curl:

|  |
| --- |
| **curl -H "Content-Type: application/json" -X GET http://localhost:8080/users/views** |

Sample Response Data:

|  |
| --- |
| **Response status = 200**  **[**  **{"userProfileId":4,"viewerProfileId":2,"viewDate":"2015-07-27 18:56:25 PM","objectId":17},**  **{"userProfileId":2,"viewerProfileId":5,"viewDate":"2015-07-27 18:56:18 PM","objectId":16},**  **{"userProfileId":3,"viewerProfileId":4,"viewDate":"2015-07-27 18:48:23 PM","objectId":15}**  **]** |

1. **Additional Tasks**

Also you can run the job manually apart from daily scheduled time using:

**To start a job manually, do a POST at**

http://localhost:8081/tasks/startjob?JOB\_NAME=CleanProfileViewJob

**To start a job manually, do a POST at**

http://localhost:8081/tasks/stopjob?JOB\_NAME=CleanProfileViewJob

**To do a healthcheck, do a GET at**

http://localhost:8081/healthcheck